Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тульский государственный университет»
Технический колледж имени С.И. Мосина

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

по дисциплине «Информатика»

для специальности

10.02.05 Обеспечение информационной безопасности автоматизированных систем

Тула 2024

УТВЕРЖДЕНЫ
на заседании цикловой комиссии информационных технологий

Протокол от «12» сентября 20 24 г. № 2

Председатель цикловой комиссии И.В.Миляева

ПРАВИЛА ТЕХНИКИ БЕЗОПАСНОСТИ ПРИ ПРОВЕДЕНИИ ПРАКТИЧЕСКИХ РАБОТ

1 Для удобства работы:

1.1 Поставьте стул так, чтобы бедра были горизонтальны, ступни — плоско на полу, а колени располагались немного выше сидения.

1.2 Лучше сидеть на стуле, у которого есть упор для нижней части спины.

1.3 При работе на клавиатуре плечи должны быть расслаблены, плечо и предплечье располагаться под прямым углом, запястье и кисть — по прямой линии.

1.4 Верхний край монитора должен быть немного ниже уровня глаз.

1.5 Монитор должен быть установлен так, чтобы экран не отсвечивал.

1.6 Необходимо регулярно протирать экран.

1.7 Следует использовать подставки для документов и монитора, обеспечивающие удобства при работе на компьютере.

2 Чтобы избежать усталости:

2.1 Не сидите все время в одном положении, вставайте или потягивайтесь при появлении чувства усталости. Небольшие перерывы в работе помогают бороться с усталостью и ощущением неудобства.

2.2 Касайтесь клавиатуры мягко. Кисти рук и пальцы должны быть расслаблены. При интенсивной работе без перерывов могут возникнуть неприятные ощущения в кистях рук, запястьях или предплечьях.

2.3 Нажимайте клавишу мышки двумя пальцами (указательным и средним).

2.4 Иногда давайте отдых своим глазам. Время от времени фиксируйте взгляд на каком-нибудь удаленном предмете. Почаще моргайте во время работы.

Практическая работа №1

Тема: Кодирование текстовой, графической, звуковой информации. Расчет объема информации, передаваемой по каналам связи

Цель работы: *знакомство* с кодированием различных видов информации, научиться рассчитывать объем информации, передаваемой по каналам связи

I. Теоретическая часть

Компьютер может обрабатывать числовую, текстовую, графическую, звуковую и видео информацию. Возникает вопрос; «Как и каким образом компьютер обрабатывает столь различающиеся по восприятию человеком виды информации?».

Все эти виды информации кодируются в последовательности электрических импульсов: есть импульс (1), нет импульса (0), т.е. в последовательности нулей и единиц. Такое кодирование информации в компьютере называется двоичным кодированием, а логические последовательности из нулей и единиц машинным языком.

Принципы обработки числовой информации

Числовую информацию компьютер обрабатывает в двоичной системе счисления. Таким образом, числа в компьютере представлены в виде последовательностей 0 и 1 или битов. В начале 80-х годов процессоры для персональных компьютеров были 8-разрядными и за один такт работы процессора, компьютер мог обработать 8 бит, т.е. максимальное обрабатываемое целое десятичное число не могло превышать 11111111. Затем разрядность процессоров росла, появились 16-, 32- и, наконец,. 64-разрядные процессоры для персональных компьютеров, соответственно возросла и величина максимального числа, обрабатываемого за один такт.

Принципы обработки текстовой информации

Начиная с конца 60-х годов, компьютеры все больше стали использоваться для обработки текстовой информации и в настоящее время большая часть персональных компьютеров в мире (и наибольшее время) занято обработкой именно текстовой информации.

Традиционно для кодирования одного символа используется 1 байт (8 двоичных разрядов). Это позволяет закодировать $N = 2^8 = 256$ различных символов, которых обычно бывает достаточно для представления текстовой информации (прописные и заглавные буквы русского и латинского алфавита, цифры, знаки, графические символы и т.д.).

При двоичном кодировании текстовой информации каждому символу ставится в соответствие своя уникальная последовательность из восьми нулей и

единиц, свой уникальный двоичный код от 00000000 до 11111111 (десятичный код от 0 до 255).

Присвоение символу конкретного двоичного кода - это вопрос соглашения, которое фиксируется в кодовой таблице. Первые 33 кода (с 0 по 32) соответствуют не символам, а операциям (перевод строки, ввод пробела и т.д.). Коды с 33 по 127 являются интернациональными и соответствуют символам латинского алфавита, цифрам, знакам арифметических операций и знакам препинания.

Коды с 128 по 255 являются национальными, т.е. в национальных кодировках одному и тому же коду соответствуют различные символы. К сожалению, в настоящее время существуют несколько различных кодовых таблиц для русских букв, поэтому тексты, созданные в одной кодировке, не будут правильно отображаться в другой.

Хронологически одним из первых стандартов кодирования русских букв на компьютерах был код КОИ-8 («Код обмена информацией 8-битный»). Эта кодировка применяется на компьютерах с операционной системой UNIX.

Наиболее распространенная кодировка - это стандартная кириллическая кодировка Microsoft Windows, обозначаемая сокращением CP1251 («CP» означает «Code Page», «кодовая страница»). Все Windows-приложения. работающие с русским языком, поддерживают эту кодировку.

Для работы в среде операционной системы MS-DOS используется «альтернативная» кодировка, в терминологии фирмы Microsoft - кодировка CP866.

Фирма Apple разработала для компьютеров Macintosh свою собственную кодировку русских букв (Mae).

Международная организация по стандартизации (International Standards Organization. ISO) утвердила в качестве стандарта для русского языка еще одну кодировку под названием ISO 8859-5.

Как видно из таблицы, одному и тому же двоичному коду в различных кодировках поставлены в соответствие различные символы. Каждая кодировка задается своей собственной кодовой таблицей.

Например, последовательность числовых кодов 221. 194. 204 в кодировке CP1251 образует слово Э*ВМ,* тогда как в других кодировках это будет бессмысленный набор символов.

К счастью, в большинстве случаев пользователь не должен заботиться о перекодировках текстовых документов. При работе в приложениях Windows предусмотрена возможность автоматической перекодировки документов, созданных в приложениях MS-DOS. При работе в Интернет с использованием броузеров Internet Explorer и Netscape Communicator происходит автоматическая перекодировка Web-страниц.

В последнее время появился новый международный стандарт Unicode. который отводит на каждый символ не один байт, а два, и потому с его помощью можно закодировать не 256 символов, а $N = 2^{16} = 65536$ различных символов. Эту кодировку поддерживает платформа начиная с Microsoft Windows&Ofncc 97.

Как определить числовой код символа

Проще всего, конечно, воспользоваться кодовой таблицей, но если ее нет под рукой, а есть компьютер на платформе Windows, загрузите текстовый редактор **LibreOffice.** Выберете команду меню **[Вставка-Специальные символы...].** На экране появится диалоговое окно <Выбор символа>.

Центральную часть диалогового окна занимает таблица символов для определенного шрифта (например,шрмфт: Times New Roman, подмножество; Кириллица). Символы располагаются последовательно слева направо и построчно.

Числовой кода символа в различных кодировках Windows высвечивается в соответствующем поле диалогового окна.

Как определить символ по числовому коду

Запустите любое приложение на платформе Window's&Office. например. **Блокнот.** С помощью дополнительной цифровой клавиатуры при нажатой клавише *{Alt}* введите число 0224. отпустите клавишу *{Alt},* в документе появится символ *а.* Повторите процедуру для числовых кодов от 0225 до 0233, в документе появится последовательность из 10 символов *(абвгдежзий)* в кодировке Windows (CP1251).

С помощью дополнительной цифровой клавиатуры при нажатой клавише *{Alt}* введите число 224, отпустите клавишу *{Alt},* в документе появится символ *р.* Повторите процедуру для числовых кодов от 225 до 233. в документе появится последовательность из 10 символов *[рстуфхцчшщ)* в кодировке MS-DOS (CP866).

Принципы обработки графической информации

С 80-х годов XX века интенсивно развивается технология обработки на компьютере графической информации. Компьютерная графика позволяет создавать и редактировать рисунки, схемы, чертежи, преобразовывать изображения (фотографии, слайды и т.д.), представлять статистические данные в форме деловой графики, создавать анимационные модели (научные, игровые и т.д.), обрабатывать «живое видео».

Графическая информация на экране монитора представляется в виде изображения, которое формируется из точек (пикселей). В простейшем случае (черно-белое изображение без градаций серого цвета) каждая точка экрана может иметь лишь два состояния — «черная» или «белая», т.е. для хранения ее состояния необходим 1 бит.

Цветные изображения могут иметь различную глубину цвета (бит на точку: 4, 8, 16, 24). Каждый цвет можно рассматривать как возможное состояние точки! и тогда по формуле $N = 2^I$ может быть вычислено количество цветов, отображаемых на экране монитора.

Глубина цвета(I)	**Количество отображаемых цветов**
4	2^4=16
8	2^8=256
16(High Color)	2^{16}=65536
24(True Color)	2^{24}=16777216

Изображение может иметь различный размер, который определяется количеством точек по горизонтали и по вертикали. В современных персональных компьютерах обычно используются четыре основных размера изображения или разрешающих способностей экрана: 640*480, 800*600, 1024*768 и 1280*1024 точки.

Графический режим вывода изображения на экран определяется **разрешающей способностью экрана и глубиной цвета.** Полная информация о всех точках изображения, хранящаяся в видеопамяти, называется **битовой картой изображения.**

Для того чтобы на экране монитора формировалось изображение, информация о каждой его точке (цвет точки) должна храниться в видеопамяти компьютера. Рассчитаем необходимый объем видеопамяти для наиболее графического режима (800*600 точек, 16 бит на точку).

Всего точек на экране: 800 * 600 = 480000

Необходимый объем видеопамяти:

16 бит * 480000 = 7680000 бит = 960000 байт = 937,5 Кбайт.

Современные компьютеры обладают такими техническими характеристиками, которые позволяют обрабатывать и выводить на экран, так называемое «живое видео», т.е. видеоизображение естественных объектов. Видеоизображение формируется из отдельных кадров, которые сменяют друг друга с высокой частотой (не воспринимаемой глазом). Обычно частота кадров составляет 25 Гц, т.е. за 1 секунду сменяется 25 кадров.

Принципы обработки звуковой информации

С начала 90-х годов персональные компьютеры получили возможность работать со звуковой информацией. Каждый компьютер, имеющий звуковую плату, микрофон и колонки, может записывать, сохранять и воспроизводить звуковую информацию. С помощью специальных программных средств (редакторов аудиофайлов) открываются широкие возможности по созданию, редактированию и прослушиванию звуковых файлов. Создаются программы распознавания речи и появляется возможность управления компьютером при помощи голоса.

Звуковой сигнал - это непрерывная волна с изменяющейся амплитудой и частотой. Чем больше амплитуда сигнала, тем он громче для человека, чем больше частота сигнала, тем выше тон. Для того чтобы компьютер мог обрабатывать непрерывный звуковой сигнал, он должен быть дискретизирован, т.е. превращен в последовательность электрических импульсов (двоичных нулей и единиц).

При **двоичном кодировании непрерывного звукового сигнала** он заменяется серией его отдельных выборок- отсчетов.

Каждая выборка фиксирует реальную амплитуду сигнала и присваивает ей определенное, наиболее близкое, дискретное значение. Чем большее количество дискретных значений может обеспечить звуковая карта и чем большее количество

выборок производится за 1 секунду, тем точнее процедура двоичного кодирования.

Современные звуковые карты могут обеспечить кодирование 65536 различных уровней сигнала или состояний. Для определения количества бит, необходимых для кодирования, решим показательное уравнение:

$65536 = 2^I$. т.к. $65536 = 2^{16}$. то I = 16 бит

Таким образом, современные звуковые карты обеспечивают 16-битное кодирование звука. При каждой выборке значению амплитуды звукового сигнала присваивается 16-битный код.

Количество выборок в секунду может быть в диапазоне от 8000 до 48000. т.е. частота дискретизации аналогового звукового сигнала может принимать значения от 8 до 48 Кгц. При частоте 8 Кгц качество дискретизированного звукового сигнала соответствует качеству радиотрансляции, а при частоте 48 Кгц - качеству звучания аудио-CD. Следует также учитывать, что возможны как моно-. так и стерео-режимы.

Можно оценить информационный объем моноаудиофайла длительностью звучания 1 секунду при среднем качестве звука (16 бит, 24 Кгц). Для этого количество бит на одну выборку необходимо умножить на количество выборок в 1 секунду:

16 бит * 24000 = 384000 бит = 48000 байт == 47 Кбайт

II. Практическая часть

Задание1

Рассчитайте необходимый объем видеопамяти для различных графических режимов (таблица 2) и заполните таблицу.

Таблица 2. Объем видеопамяти для различных графических режимов

Режим экрана	**Глубина цвета (бит на точку)**			
	4	**8**	**16**	**24**
640 на 480				
800 на 600				
1024 на 768				
1280 на 1024				

Задание 2

Рассчитайте объем моноаудиофайла длительностью 10 секунд при 16-битном кодировании и частоте дискретизации 44 Кгц.

Задание 3.

Решите задачу.

Максимальная скорость передачи данных в локальной сети 100 Мбит/с. Сколько страниц текста можно передать за 1 сек, если 1 страница текста содержит 50 строк и на каждой строке - 70 символов.

Отчет должен содержать:

1. Название работы.

2. Цель работы.
3. Задание.
4. Результаты выполнения задания.
5. Вывод по работе.

Практическая работа №2

Тема: Перевод чисел из одной системы счисления в другую и наоборот

Цель работы: *научиться осуществлять перевод* чисел из одной системы счисления в другую и наоборот

Теоретическая часть

При работе с компьютерами приходится параллельно использовать несколько позиционных систем счисления (чаще всего двоичную, десятичную и шестнадцатеричную), поэтому большое практическое значение имеют процедуры перевода чисел из одной системы счисления в другую.

Преобразование чисел **из двоичной, восьмеричной и шестнадцатеричной систем счисления в десятичную** выполнить довольно легко. Для этого необходимо записать число в полной форме и вычислить его значение.

Алгоритм: При переводе чисел из системы счисления с основанием *P* в десятичную систему счисления необходимо пронумеровать разряды целой части справа налево, начиная с нулевого, и дробной части, начиная с разряда сразу после запятой, слева направо (начальный номер — 1). Затем вычислить сумму произведений соответствующих значений разрядов на основание системы счисления в степени, равной номеру разряда. Это и есть представление исходного числа в десятичной системе счисления.

При переводе чисел из десятичной системы счисления в систему с основанием P>1 обычно используют следующий алгоритм:

1) если переводится целая часть числа, то она делится на *P,* после чего запоминается остаток от деления. Полученное частное вновь делится на *P,* остаток запоминается. Процедура продолжается до тех пор, пока частное не станет равным нулю. Остатки отделения на P выписываются в порядке, обратном их получению;

2) если переводится дробная часть числа, то она умножается на *P,* после чего делая часть запоминается и отбрасывается. Вновь полученная дробная часть умножкается на P и т.д. Процедура продолжается до тех пор, пока дробная часть не станет равной нулю. Целые части выписываются после двоичной запятой в порядке их получения. Результатом может быть либо конечная, либо периодическая двоичная дробь. Поэтому, когда дробь является периодической, приходится обрывать умножение на каком-либо шаге и довольствоваться приближенной записью исходного числа в системе с основанием *P*.

Если необходимо перевести **число из двоичной системы счисления в систему счисления, основанием которой является степень двойки**, достаточно объединить цифры двоичного числа в группы по столько цифр, каков показатель степени, и использовать приведенный ниже алгоритм. Например, если перевод осуществляется в восьмеричную систему, то группы будут содержать три цифры ($8 = 2^3$). В целой части числа группировка производится справа налево, в дробной части — слева направо. Если в последней группе недостает цифр, дописываются

нули: в целой части — слева, в дробной — справа. Затем каждая группа заменяется соответствующей цифрой новой системы. Соответствия приведены в таблице.

P	Соответствия															
2	00	01	10	11												
4	0	1	2	3												
2	000	001	010	011	100	101	110	111								
8	0	1	2	3	4	5	6	7								
2	00	00	00	00	01	01	01	01	10	10	10	10	11	11	11	11
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Практическая часть

Задание 1. Перевести данное число в десятичную систему счисления:

а) 1000001_2.

$1000001_2=1\cdot2^6+0\cdot2^5+0\cdot2^4+0\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0= 64 + 1 =65_{10}$.

Замечание. Если в каком-либо разряде стоит нуль, то соответствующее слагаемое можно опускать;

б) $1000011111{,}0101_2$.

$1000011111{,}0101_2= 1\cdot2^9+1\cdot2^4+1\cdot2^3+1\cdot2^2+1\cdot2^1+1\cdot2^0+1\cdot2^{-2}+1\cdot2^{-4}=$
$=512+16+8+4+2+1+ 0{,}25 + 0{,}0625 = 543{,}3125_{10}$;

в) $1216{,}04_8$.

$1216{,}04_8= 1\cdot8^3+2\cdot8^2+ 1\cdot8^1 + 6\cdot8^0+ 4\cdot8^{-2}= 512 + 128 + 8 +6 + 0{,}0625 = 654{,}0625_{10}$

г) $29A{,}5_{16}$

$29A{,}5_{16}= 2\cdot16^2+ 9\cdot16^1 + 10\cdot16^0 + 5\cdot16^{-1} = 512 + 144 + 10 + 0{,}3125 = 65б{,}3125_{10}$.

Задание 2. Перевести данное число из десятичной системы счисления в двоичную (получить пять знаков после запятой в двоичном представлении).

а) 464_{10}; б) $380{,}1875_{10}$; в) $115{,}94_{10}$.

Решение:

а)		б)				в)			
464	0	380	0		1875	115	1		94
232	0	190	0	0	375	57	1	1	88
116	0	95	1	0	75	28	0	1	76
58	0	47	1	1	5	14	0	1	52
29	1	23	1	1	0	7	1	1	04
14	0	11	1			3	1	0	08
7	1	5	1			1	1	0	16
3	1	2	0						
1	1	1	1						

а) $464_{10}= 111010000_2$; б) $380{,}1875_{10} = 101111100{,}0011_2$;

в) $115{,}94_{10} = 1110011{,}11110_2$

(в данном случае было получено шесть знаков после запятой, после чего результат был округлен).

Задание 3. Переведем из двоичной системы число $1111010101,11_2$. в восьмеричную и в шестнадцатеричную

$1111010101,11_2$

$001111010101,110_2$. $=1724,6_8$

$001111010101,1100_2 = 3D5,C_{16}$.

Задание 4

1 Запишите десятичные эквиваленты следующих чисел: 120_8, $25A,2_{16}$..

2.Переведите данные числа из двоичной системы счисления в десятичную:

а) 1101110001110; 1010010001010;

б) 0,0101; 0,1011

3. Какое число следует за 677_8, 777_8?

Ответ запишите в указанной системе счисления.

4. Переведите следующие десятичные числа в двоичную систему счисления: 40,5; 31,75.

5. Переведите следующие числа из десятичной системы счисления в восьмеричную: 12,4; 481,625

6. Переведите следующие числа из десятичной системы счисления в шестнадцатеричную: 28,7; 25,25.

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание.
4. Результаты выполнения задания.
5. Вывод по работе.

Практическая работа №3

Тема: Изучение архитектуры компьютера

Цель работы: знакомство с архитектурой компьютера

I. Теоретическая часть

Персональный компьютер (ПК) – это компьютер, предназначенный для индивидуального использования. В настоящее время это мощный универсальный компьютер, который работает как дома, так и на рабочих местах в офисах, легко подключается к различным вычислительным системам.

Персональный компьютер – это универсальное (многофункциональное) электронное программно-управляемое устройство для хранения, обработки и передачи информации.

Технической основой ПК служит микропроцессор (МП). Развитие технологии МП определило смену поколений персональных компьютеров:

- 8-разрядный МП (1975 – 1980) – I поколение;
- 16-разрядный МП (1981 – 1985) – II поколение;
- 32-разрядный МП (1986 – 1992) – III поколение (поддерживает до 32 ядер процессора);
- 64-разрядный МП (1993 г. – по настоящее время) – IV поколение (поддерживает до 256 ядер процессора).

Важную роль в развитии ПК сыграло появление компьютера IBM PC, произведенного корпорацией IBM (США) на базе МП Intel-8086 в 1981 г. Этот персональный компьютер занял ведущее место на рынке ПК. Его основное преимущество – открытая архитектура, благодаря которой пользователи могут расширять возможности ПК, добавляя различные периферийные устройства и модернизируя компьютер. В наши дни » 85% всех компьютеров базируется на архитектуре IBM PC.

Классификация ПК по назначению

ПК общего назначения – предназначены для массового потребителя для развлечения, обучения и работы.

Профессиональные ПК – применяются в научной сфере, для решения сложных информационных и производственных задач, где требуется высокое быстродействие, эффективная передача больших массивов информации, достаточно большая емкость оперативной памяти.

Классификация ПК по конструктивному исполнению

1) *Стационарные ПК (непереносные)*

а) *Раздельная схема* – ПК состоит из системного блока и разнообразных внешних, то есть конструктивно самостоятельных подключаемых к системному блоку извне через стандартные интерфейсы (например: USB, D-Sub, DVI, FireWire), устройств (в частности: мониторы, клавиатура, мышь, микрофоны, звуковые колонки, веб-камеры, принтеры, сканеры, различные внешние модемы, игровые устройства).

Исторически такая схема ПК было самой первой. Она же до сих пор остается самой распространенной схемой стационарных ПК. Например, профессиональные рабочие станции практически всегда строятся по такой схеме.

Главное достоинство раздельной схемы — сравнительно легкая масштабируемость. То есть в любой момент можно без особых затруднений заменить любой из компонентов ПК (например, монитор). Но обратная сторона медали — наименьшая транспортабельность и сравнительная громоздкость такого ПК. Естественно раздельная схема применяется тогда когда главное требование к ПК — легкость и простота масштабирования.

Функциональным ядром в раздельной схеме стационарного ПК естественно является системный блок.

Известны два вида конструктивной компоновки системного блока:

- *desktop* — горизонтальная конструктивная компоновка системного блока, с возможностью размещения монитора на таком системном блоке; настольный компьютер – компьютер, предназначенный для постоянного размещения на столе.

- *tower* — «башенный» системный блок в вертикальной конструктивной компоновке.

б) *Моноблок* – конструктивная схема стационарного ПК, в которой системный блок, монитор и, в настоящее время, микрофон, звуковая колонки, веб-камера конструктивно объединены в одно устройство. Такой ПК эргономичнее (занимает минимум пространства) и более привлекателен с эстетической точки зрения. Также, такой ПК более транспортабелен, чем стационарные ПК, построенные по раздельной схеме. С другой стороны, такой ПК сложнее масштабировать и, в том числе, затруднена самостоятельная техническая модернизация и обслуживание. Например, если у моноблока сломается микрофон, то заменить его на исправный нередко возможно только в сервис-центре.

2) *Мобильные (переносные) ПК*

а) *Ноутбуки* – компактные компьютеры, содержащие все необходимые компоненты (в том числе монитор) в одном небольшом корпусе, как правило, складывающемся в виде книжки (отсюда и название данного вида ПК). Приспособлены для работы в дороге, на небольшом свободном пространстве. Для достижения малых размеров в них применяются специальные технологии: специально разработанные специализированные микросхемы (ASIC), ОЗУ и жёсткие диски уменьшенных габаритов, компактная клавиатура, не содержащая цифрового поля, внешние блоки питания, минимум интерфейсных гнезд для подключения внешних устройств.

Как правило, содержат развитые средства подключения к проводным и беспроводным сетям, встроенное мультимедийное оборудование (динамики, часто, также, микрофон и веб-камеру). В последнее время вычислительная мощность и функциональность ноутбуков не сильно уступают стационарным ПК, а иногда и превосходит их. Очень компактные модели не оснащаются встроенным CD/DVD-дисководом.

Подключая к ноутбуку внешние клавиатуру, мышь, монитор, звуковые колонки, модемы, игровые устройства и иные внешние устройства ноутбук можно превратить в настольный ПК. Это можно делать вставляя ноутбук в специальный док, как это делалось раньше или напрямую (современные ноутбуки, особенно предназначенные для замены стационарных ПК в качестве рабочих станций, дают такую возможность).

б) *Планшетные ПК*

Аналогичны ноутбукам, но содержат сенсорный, то есть чувствительный к нажатию, экран и не содержат механической клавиатуры. Ввод текста и управление осуществляются через экранный интерфейс, часто доработанный специально для удобного управления пальцами. Некоторые модели могут распознавать рукописный текст, написанный на экране.

Чаще всего корпус не раскрывается, как у ноутбуков, а экран расположен на внешней стороне верхней поверхности. Бывают и комбинированные модели, у которых корпус может тем или иным образом раскрываться (например, как слайдер), предоставляя доступ к расположенной внутри клавиатуре.

По вычислительной мощи планшетные ПК уступают стационарным и ноутбукам, так как для длительной работы без внешнего источника питания приходится использовать энергосберегающие комплектующие, жертвуя их быстродействием.

в) *Карманные ПК (PDA)* – сверхпортативные ПК, умещающиеся в кармане. Управление ими, как правило, происходит с помощью небольшого по размерам и разрешению экрана, чувствительного к нажатию пальца или специальной палочки-указки — стилуса, а клавиатура и мышь отсутствуют. Однако некоторые модели содержат миниатюрную фиксированную или выдвигающуюся из корпуса клавиатуру.

В таких устройствах используются сверхэкономичные процессоры и флеш-накопители небольшого объёма, поэтому их вычислительная мощь несопоставима с другими ПК (особенно стационарными). Тем не менее, они содержат все признаки персонального компьютера: процессор, накопитель, оперативную память, монитор, операционную систему, прикладное ПО и даже игры и ориентированность на индивидуальное использование.

Всё более популярными становятся КПК с функциями мобильного телефона (коммуникаторы). Встроенный коммуникационный модуль позволяет не только совершать звонки, но и подключаться к Интернету в любой точке, где есть сотовая связь совместимого стандарта (GSM/GPRS/3G, CDMA).

3) *Нестандартные конструкции ПК*

а) *Barebone* – компьютеры, строящиеся пользователем для выполнения определенных задач (обычно в качестве мультимедийной станции). В продажу поступают в виде так называемых «скелетных» баз в составе корпуса, материнской платы и системы охлаждения. Материнская плата, как правило, оснащена встроенными звуковым и видеоконтроллерами. Выбор конфигурации и соответственно комплектующих в виде дисковых накопителей, памяти и периферии, а также других устройств (ТВ-тюнера, дополнительной видеокарты и

т. п.) остаются на усмотрение пользователя. Как правило, «баребоны» имеют меньшую высоту корпуса и, как следствие, уменьшенный внутренний объём, а также усовершенствованную систему охлаждения, отличающуюся низкой шумностью.

б) *Защищённые ПК*

Ряд компаний производит компьютеры, обладающие устойчивостью к агрессивным средам: сильной вибрации, ударам, большой запыленности, влажности, вандализму – условиям, в которых обычные ПК быстро бы вышли из строя. Как правило, устойчивые ПК выпускаются в формате ноутбуков, более тяжёлых и больших по размерам, чем обычные. Их стоимость также значительно выше. Одна из сфер применения таких ПК — военное дело (например, эксплуатация в полевых штабах).

в) *Промышленные ПК*

Предназначены для решения задач промышленной автоматизации. Отличаются стойкостью к различным внешним воздействиям, увеличенным жизненным циклом изделия, возможностью подключения к промышленным сетям (Profinet, Profibus).

г) *Тихий ПК*

Для использования в жилых комнатах используются конструкции ПК, производящие минимум шума или работающие совершенно бесшумно. Такие модели можно оставлять включенными постоянно, что даёт ряд преимуществ: отсутствует период загрузки, компьютер всегда готов к работе и может постоянно отслеживать новую почту или мгновенные сообщения для пользователя.

Бесшумный компьютер Zonbu

В целом, постоянно включенный ПК может выполнять ряд особенных задач:

- быть мультимедийной станцией (воспроизводить видео-, аудиозаписи, интернет-радио);
- работать как видеомагнитофон: записывать передачи телевидения или радио для последующего просмотра или прослушивания в удобное время;
- служить P2P-клиентом (обмениваться файлами в автоматическом режиме с другими компьютерами)
- служить домашним или даже интернет-сервером;
- следить за температурой или присутствием с помощью соответствующих датчиков или фото-, видеокамеры (веб-камеры).

Чтобы сделать ПК тихим, используется несколько технологий:

- безвентиляторные типы охлаждения:
- жидкостное (с передачей жидкости на большой пассивно-охлаждаемый радиатор)
- применение термотруб (передача всей энергии путем термотруб на поверхность корпуса, также состоящего из меди или алюминия)
- применение очень крупных радиаторов (часто с термотрубами)

- погружение всей электроники в резервуар с некондуктивным маслом
- фреоновое (применяется микрохолодильник с соответствующей электроникой и изоляцией. Не всегда «тихий». К примеру Vapo-chill)
- жидкий азот (только кратковременное, не предназначено для сколь-либо долгой эксплуатации, как правило для разгона — хотя бесшумно)
- малошумные вентиляторы с лопастями специальной формы;
- процессоры, не требующие активного охлаждения (ввиду их маломощности, это не всегда приемлемое решение);
- малошумные жёсткие диски, а также установка их на шумопоглощающие крепления;
- замена жёстких дисков на флэш-память или удалённые дисковые массивы;
- установка бесшумного (noiseless) блока питания.

Большинство современных персональных компьютеров способны снижать потребляемую мощность и уровень шума в моменты низкой нагрузки, но для постоянной тихой работы не обойтись без применения специальных технологий, указанных выше.

д) Компактные ПК

Некоторые компании предлагают ПК значительно меньших размеров, чем стандартные. Такие модели занимают меньше места в рабочей или домашней обстановке, легче вписываются в интерьер, зачастую красивее и тише обычных ПК. Собрать компактную модель по силам большинству пользователей, если подобрать специальные модели корпуса и материнской платы.

Одними из первых компактных компьютеров были модели Macintosh в 1984 году, которые представляли собой моноблок: системные компоненты в одном корпусе с монитором. Значительно позже идея была продолжена в моделях eMac и iMac. Аналогичные по формату компьютеры пытались выпускать и другие компании (например, eMachines), но без особого успеха.

Существует несколько конкурирующих между собой проектов компактных и дешёвых в производстве персональных компьютеров, некоторые из которых предназначены для развивающихся стран: OLPC, VIA pc-1 Initiative, Classmate PC, Asus Eee PC и др. Однако удешевление и миниатюризация достигнуты ценой заметного отставания по вычислительной мощности от полноразмерных ПК.

е) *Хакинтош*

Хакинтош (англ. hackintosh, от слов хакер или хак и макинтош) — ПК, собранный любителем и способный работать под управлением Mac OS X, во взломанном для запуска на «неяблочном» компьютере варианте называемой OSx86, то есть более дешёвый аналог компьютера от Apple. Так как современные макинтоши рассчитаны на процессоры Intel и другие стандартные компоненты, возникает теоретическая возможность запускать Mac OS X на любых ПК на базе этих процессоров. В реальности поддерживается только узкий набор аппаратных конфигураций, которые встречается в настоящих макинтошах, поэтому «хакинтош» должен повторять одну из этих конфигураций. С другой стороны Mac OS X создана для макинтоша и только, и корректно и максимально производительно будет работать только на макинтоше. Кроме того, в легально поставляемой Mac OS X присутствуют ограничения, не позволяющие ей работать

на чужой аппаратуре, так что в «хакинтоше» применяют старую служебную версию без этих ограничений, либо взломанную более свежую версию, либо специальные аппаратные средства, имитирующие сигнатуры макинтоша, проверяемые системой. Установка системы Mac OS X на компьютеры, не произведённые Apple, является также нарушением лицензии на эту ОС.

Аппаратная реализация компьютера

Настольные компьютеры состоят из *системного блока, монитора, клавиатуры и мыши*.

Системный блок является в компьютере главной частью. В нем располагаются все основные компоненты компьютера:

- материнская (системная) плата, к которой подключаются все остальные платы и микросхемы (микропроцессор, оперативная память, контроллеры различных устройств);
- блок питания, преобразующий напряжение сети в постоянный ток низкого напряжения для питания различных компонентов компьютера;
- накопитель на жестком магнитном диске (винчестер);
- дисководы для чтения и записи компакт-дисков.

На заднюю панель системного блока выведены разъемы, через которые к компьютеру подключаются различные внешние устройства: монитор, клавиатура, принтер и т.д.

Каждое внешнее устройство подключаются к центральной части компьютера (микропроцессор и память) с помощью контроллеров (адаптеров). Контроллеры управляют внешними устройствами. Каждому внешнему устройству соответствует свой контроллер.

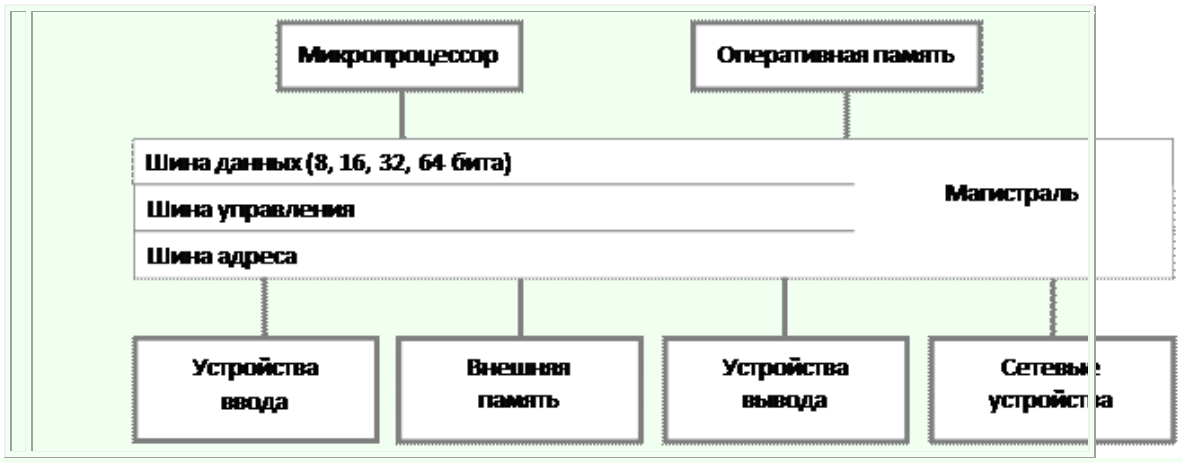
Архитектура– это наиболее общие принципы построения компьютера, отражающие программное управление работой и взаимодействием его основных функциональных узлов.

В основу архитектуры современных персональных компьютеров положен **магистрально-модульный принцип**.

Модульный принцип позволяет потребителю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости ее модернизацию.

Модульная организация компьютера опирается на магистральный (шинный) принцип обмена информацией между устройствами.

К магистрали подключаются **процессор** и **оперативная память**, а также периферийные устройства ввода, вывода и устройства хранения информации, которые обмениваются информацией на машинном языке (последовательностями нулей и единиц в форме электрических импульсов).



Магистрально-модульное устройство компьютера

1) Микропроцессор – выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера.

Назначение процессора:

- обработка всех видов информации;
- управление работой всех узлов компьютера.

В состав процессора входят следующие устройства:

- устройство управления (УУ);
- арифметико-логическое устройство (АЛУ);
- регистры процессорной памяти.

УУ управляет работой всех устройств компьютера по заданной программе.

АЛУ выполняет арифметические и логические операции (обработка информации).

Регистры – это внутренняя память процессора. Каждый из регистров служит своего рода черновиком, используя который процессор выполняет расчёты и сохраняет промежуточные результаты.

Основные характеристики процессора:

- тактовая частота;
- разрядность процессора.

Микропроцессор – это «мозг» и «сердце» компьютера. Основной характеристикой микропроцессора является **тактовая частота**, которая в значительной степени определяет его быстродействие. Чем выше тактовая частота микропроцессора, тем выше его производительность. Другой характеристикой процессора является его **разрядность**. Разрядность процессора определяется числом двоичных разрядов, которые процессор обрабатывает за один такт.

С момента появления первого процессора (1971 г.) тактовая частота процессоров увеличилась в 38 000 раз (с 0,1 МГц до 3 800 МГц), разрядность увеличилась в 16 раз (с 4 битов до 64 битов).

В настоящее время производительность процессоров увеличивается путем совершенствования архитектуры процессора (введение в структуру процессора **кэш-памяти, многоядерность** – вместо одного ядра процессора используется много ядер.

Производительность процессора нельзя вычислить – она определяется в процессе тестирования, по скорости выполнения процессором определенных операций в какой-либо программной среде.

2) Оперативная память – (RAM – англ. Random Access Memory – память с произвольным доступом) — часть системы компьютерной памяти, в которой временно хранятся программы в процессе их выполнения и данные в процессе их обработки процессором.

Характеристики внутренней памяти

Внутренняя память компьютера служит для временного хранения информации и включает в себя:

- оперативное запоминающее устройство (ОЗУ)
- постоянное запоминающее устройство (ПЗУ)

Важное значение для производительности компьютера имеет объем оперативной памяти.

Оперативная память представляет собой множество ячеек, каждая ячейка имеет свой уникальный адрес. Каждая ячейка памяти имеет объем 1 байт. Максимальный объем адресуемой памяти для Pentium 4 равен 64 Гбайт.

В персональных компьютерах объем адресуемой памяти и величина фактически установленной оперативной памяти (модулей оперативной памяти) практически всегда различаются. Величина фактически установленной оперативной памяти может быть 2 – 4 Гбайт.

Существует четыре главные характеристики микросхемы оперативной памяти: тип, структура, объем и время доступа к ячейке.

Постоянная память (ROM – Read Only Memory – память только для чтения) – энергонезависимая память. Содержание памяти «зашивается» при ее изготовлении для постоянного хранения. В постоянной памяти хранятся программы управления работой процессора, внешней памяти, дисплея, клавиатуры, принтера, программы запуска и остановки компьютера, программы тестирования устройств.

Видеопамять VRAM – разновидность оперативного запоминающего устройства, в котором хранятся закодированные изображения. Это запоминающее устройство организовано так, что его содержимое доступно сразу двум устройствам – процессору и монитору, поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.

3) Внешняя память – используется для постоянного (длительного) хранения информации – программ и данных: накопитель на жестких магнитных дисках (HDD – Hard Disk Drive), или винчестер, дисководы для компакт-дисков (CD и DVD).

4) Сетевые устройства – необходимы для подключения компьютера к сети: сетевые адаптеры, каналы связи, устройства, поддерживающие функционирование сети (маршрутизаторы, концентраторы, коммутаторы).

5) Периферийные устройства – это устройства, с помощью которых информация или вводится в компьютер, или выводится из него.

Периферийные устройства рекомендуется подключать к выключенному компьютеру.

<u>а) Устройства ввода</u> – оборудование, с помощью которого можно вводить данные: клавиатура, мышь, джойстик, трекбол, тачпад, световое перо, сенсорные экраны, сканеры, цифровые камеры, ТВ-тюнеры, системы распознавания речи, сенсорные датчики.

II Программа работы

1 Изучите теоретический материал.
2 По предложенному преподавателем варианту выполните задания.

Отчет должен содержать:
1. Название работы.
2. Цель работы.
3. Задание.
4. Результаты выполнения задания.
5. Вывод по работе.

Практическая работа № 4

Тема работы: Набор текста, редактирование и форматирование документа в текстовом процессоре. Создание и форматирование таблиц в текстовом документе. Приемы форматирования таблиц в текстовом процессоре Работа с формулами, ссылками в текстовом документе.

Цель работы: получить навыки форматирования, изучить способы создания сносок, ссылок и оглавлений и форматирования таблиц.

Оборудование, приборы, аппаратура, материалы: персональный компьютер, текстовый процессор LibreOffice Writer.

1 Краткие теоретические сведения

Основные приемы форматирования текста

Форматирование текста выполняется средствами меню **Формат** или инструментами панели **Форматирование**. Основными объектами форматирования являются абзац и шрифт. Для форматирования шрифта необходимо предварительно выделить нужную область текста, а для абзаца – достаточно установить курсор на любую позицию внутри абзаца.

Наиболее популярные способы выделения текста:

• выделение символа – протяжка мышью или *Shift* +стрелка вправо;

• выделение слова – протяжка или двойной щелчок мышью по слову;

• выделение строки – щелчок мышью в зоне выделения против строки;

• выделение абзаца – двойной щелчок мыши в зоне выделения или тройной щелчок внутри абзаца или протяжка мышью в зоне выделения;

• выделение текста от позиции курсора до конца документа – *Shift +Ctrl + End*;

• выделение текста постранично – *Shift+ Page Down*;

• выделение прямоугольной области страницы – *Alt*+ протяжка мышью;

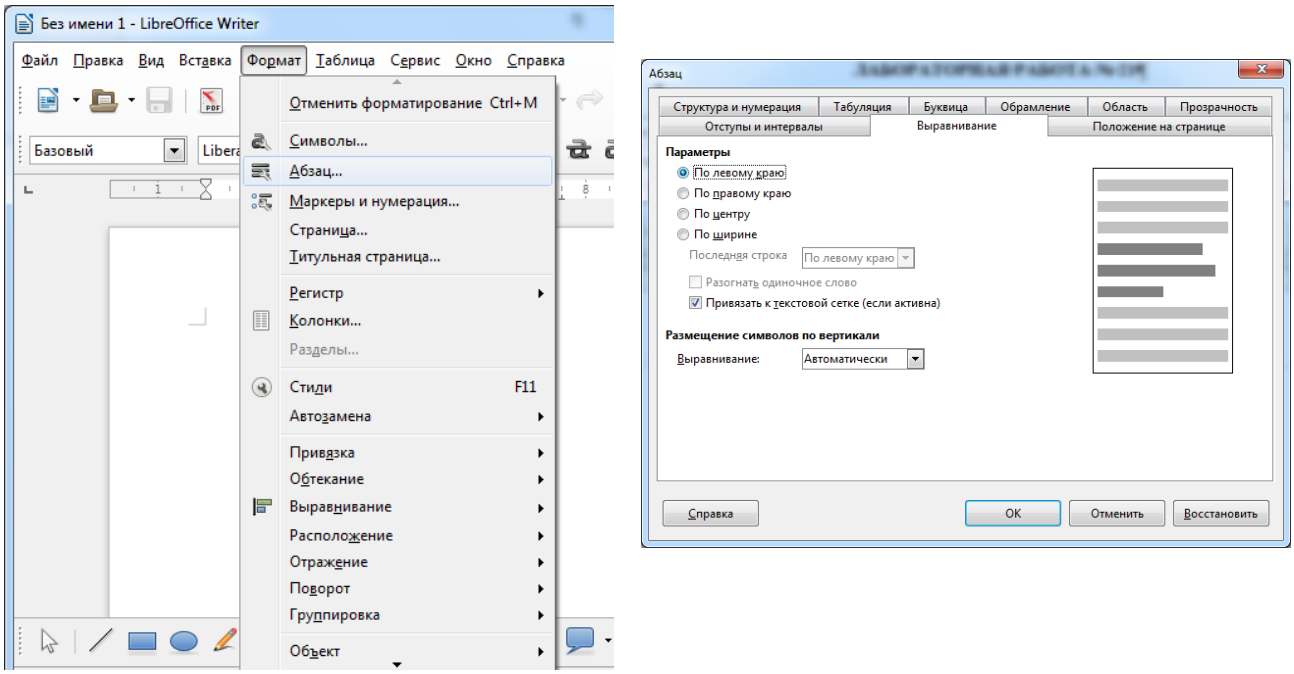
• выделение всего текста – тройной щелчок в зоне выделения или выполнение меню Правка – Выделить все.

Форматирование абзаца

Форматирование шрифта выполняется через меню **Формат – Абзац**. Этот вид форматирования применяется наиболее часто. Окно для настройки параметров абзаца показано на рисунке 1.

Поле со списком **Выравнивание** задает способ выравнивания абзаца: по левому краю, по центру, по правому краю, по ширине. Обычно на панели форматирования для этих настроек выводятся четыре кнопки быстрого доступа. Поля **Отступ слева** и **Отступ справа** задают интервал между левой и правой границами абзаца и соответствующими границами области текста. Эти отступы можно также задать нижними метками на горизонтальной линейке. Поле со списком **Первая строка** задает отступ слева в первой строке абзаца (красная строка). Этот отступ можно отрегулировать также с помощью верхней метки на горизонтальной линейке. Для задания интервала между абзацами служат поля

Интервал перед и **Интервал после**. Эти настройки применяются для улучшения читаемости текста. Интервал между строк абзаца можно задать с помощью поля "междустрочный". Поле "значение" используется для точного задания междустрочного интервала, например, для подгонки текста под размер страницы. Допускается задавать интервал в виде десятичной дроби.



а б

Рис. 1. Окно для настройки параметров абзаца: *а* – доступ к окну **Абзац;** б – содержимое вкладки **Абзац**

Форматирование шрифта

Форматирование шрифта выполняется через меню **Формат – Символ**. Окно **Символ** имеет шесть вкладок: **Шрифт**, **Эффекты шрифта, Положение, Гиперссылка, Фон, Обрамление**. На вкладке **Шрифт** можно задать вид шрифта, его начертание (**полужирный** и другие), размер. Размер задается в **пунктах**, являющихся типографской единицей измерения. На вкладке **Эффекты шрифта** расположены флажки, которыми можно задать зачеркивание, подчеркивание, тень и другие эффекты. На вкладке **Положение** можно задать нижние и верхние индексы, интервал между буквами (уплотненный или разреженный), смещение букв по вертикали и кернинг. **Кернинг** – это изменение расстояния между буквами в зависимости от их начертания.

Форматирование списков

Существует два типа списков: маркированные и нумерованные.

• **Маркированный список** – это последовательность абзацев, в начале каждого из которых расположен маркер.

• **Нумерованный список** отличается от маркированного тем, что вместо маркера используется номер абзаца.

Для организации любого типа списков следует выделить группу абзацев и щелкнуть по значку списка на панели форматирования. Для изменения **формы маркера** или **формата нумерации** необходимо открыть соответствующую вкладку в окне **Список** через меню **Формат – Список**.

Форматирование таблиц

Таблицы являются основным средством форматирования документов со сложной структурой. Доступ к созданию и форматированию таблиц можно получить через меню **Таблица**. Для вставки таблицы следует установить курсор на новую строку и выполнить команду **Таблица – Вставить таблицу** (рис. 2)**.** В окне **Вставка таблицы** следует задать количество строк и столбцов таблицы.

Если включен показ непечатаемых символов, то в каждой ячейке таблицы и в конце каждой строки можно увидеть маленький кружок, означающий конец ячейки (строки). Если длина вводимого в ячейку текста превышает ее ширину, то курсор переводится на новую строку, а высота всех ячеек строки увеличивается. Знак конца ячейки всегда расположен в конце вводимого текста.

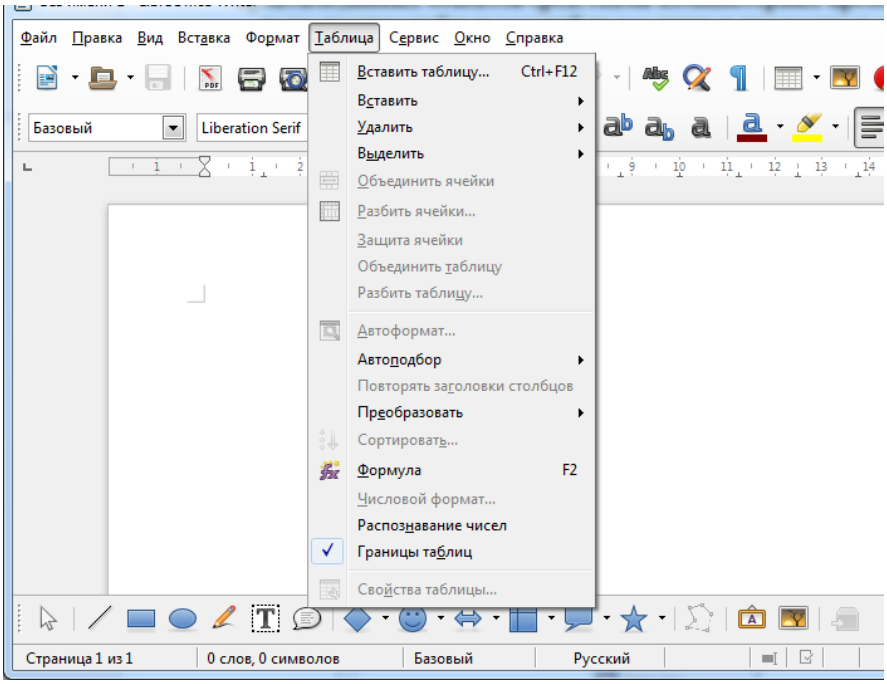


Рис. 2. Организация вставки таблицы

Ячейки таблицы можно объединять как по горизонтали, так и по вертикали. Для этого объединяемые ячейки следует выделить и выполнить **Таблица – Объединить ячейки**. Размеры отдельных ячеек, строк или столбцов можно изменить: выделить соответствующий элемент и переместить границу мышью. При этом указатель должен иметь форму двунаправленной стрелки. Выделить столбец можно, щелкнув указателем мыши по верхней границе столбца. При этом указатель должен принять форму жирной стрелки, направленной вниз. Указанная операция может быть невыполнима, если в таблице имеются объединенные ячейки. Если выделена вся таблица, то операции выравнивания (например,

центрирования) приводят к выравниванию таблицы относительно страницы. В противном случае выравнивается содержимое ячеек таблицы относительно границ ячеек. Обрамление ячеек таблицы форматируется через меню **Формат – Границы и заливка...**. В окне **Границы и заливка** можно задать толщину обрамления или отменить его вообще (рис. 3). Многие из этих операций реализуются кнопкой **Границы** на панели форматирования. Обрамляемые ячейки должны быть предварительно выделены.

Создание рисунков средствами панели Рисования

Панель **Рисование** (рис. 4) вызывается кнопкой **Рисование** на основной панели инструментов. Принципиальная особенность рисунков, создаваемых инструментами панели **Рисования**, в отличие от рисунков в графическом редакторе *Paint* в том, что это **векторная графика**. Ее отличие от **растровой графики** в том, что рисунки хранятся в памяти компьютера не в виде отдельных точек (пикселей), а в виде математических зависимостей. Поэтому векторные рисунки масштабируются без искажения, могут быть разделены на отдельные независимые детали и, обычно, требуют меньше памяти для хранения, чем растровые. Общая методика создания рисунка заключается в подготовке отдельных элементов рисунка и в их последующей сборке. Рисование элементов лучше выполнять на свободном участке страницы, причем мелкие элементы – в увеличенном масштабе.

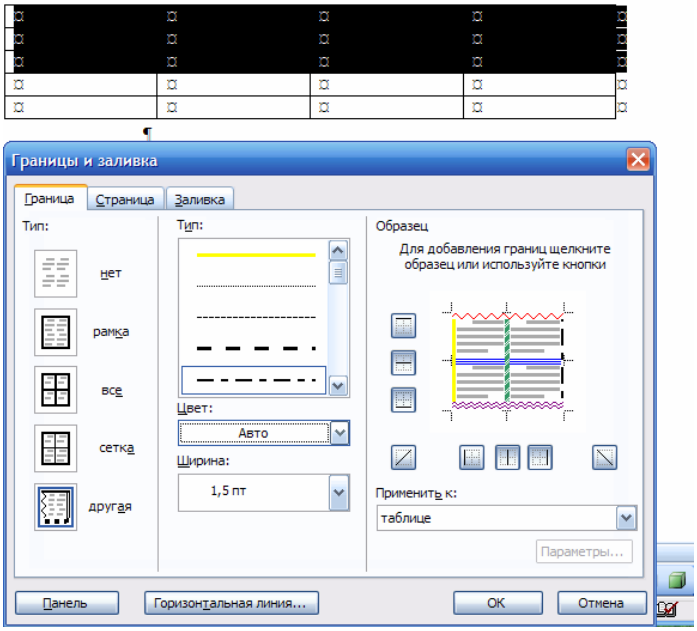


Рис. 3. Создание обрамления ячеек таблицы

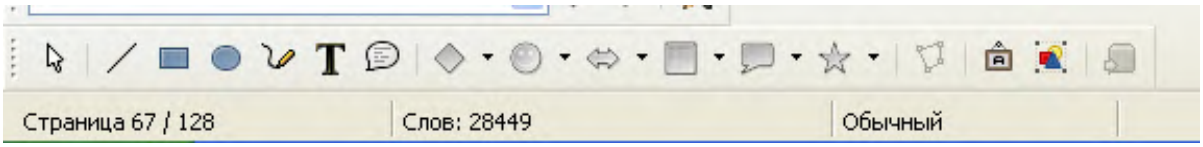


Рис. 4. Инструменты панели рисования

Вставка и редактирование изображений из файлов

Вставка в текст рисунка из файла выполняется командой **Вставка – Объект – Объект OLE…** В окне **Вставка объекта OLE** следует указать файл и установить режим вставки:

- Создать новый – самостоятельно создать рисунок в среде OLE.
- Создать из файла – картинка вставляется из файла.

Вставка номеров страниц

Для вставки номеров страниц следует выполнить: **Вставка – Поля –Номер страницы** (рис. 5). Номера автоматически помещаются в области колонтитулов, расположенные в верхней и нижней частях страниц. Высота области колонтитулов должна быть достаточной для размещения номеров. По умолчанию она равна 1,25 см. Выполнение команды переход на другую страницу осуществляется последовательностью действий **Вставка – Разрыв …** (рис. 6).

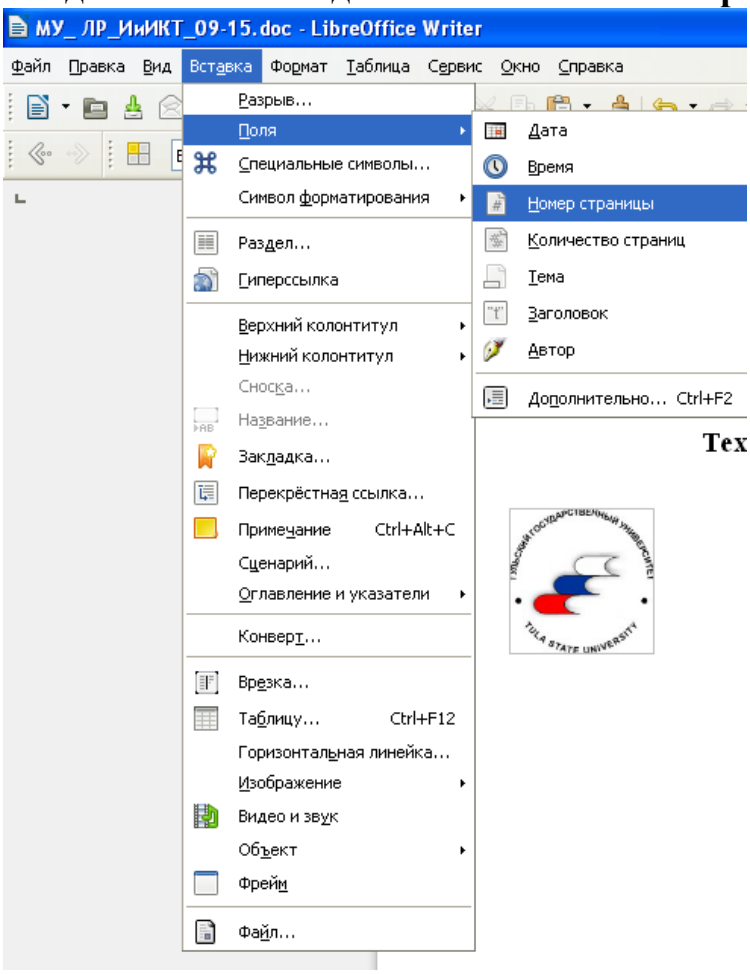


Рис. 5. Выполнение команды **вставка номера страницы**

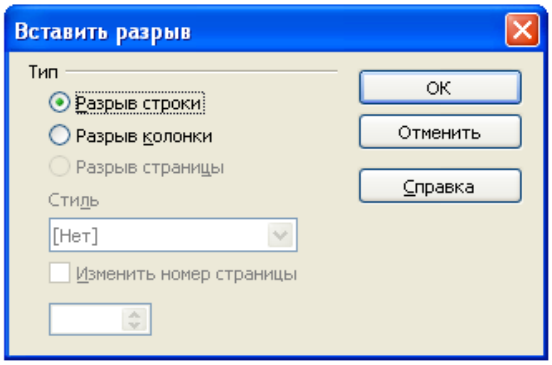


Рис. 6. Выполнение команды **переход на новую страницу**

Колонтитулы

Колонтитулами называют области, расположенные в верхнем и нижнем поле каждой страницы документа (рис. 7). Здесь полями называем пустое пространство на странице за пределами области печати.

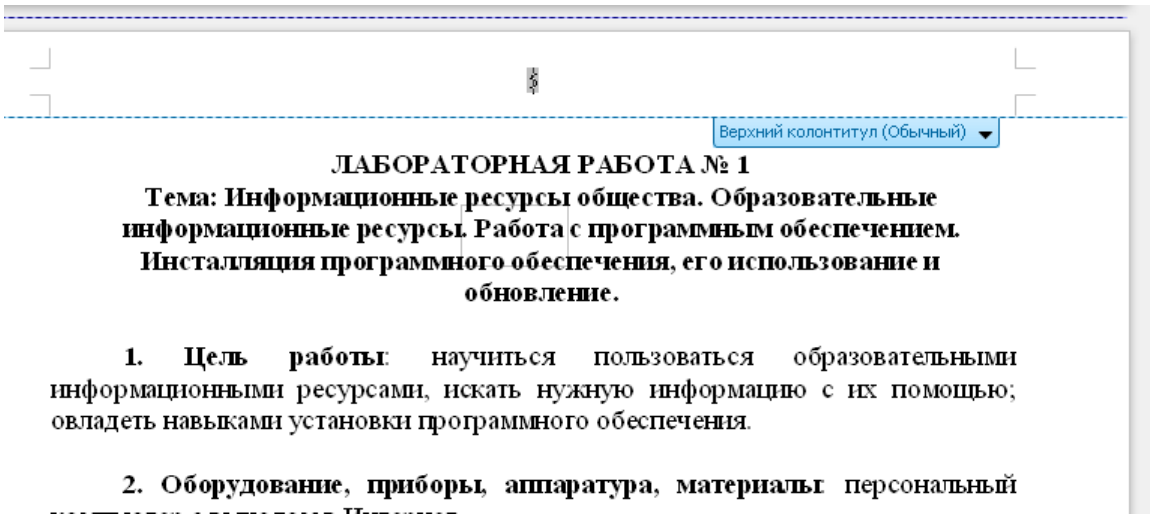


Рис. 7. Схема расположения колонтитулов

В колонтитул обычно вставляется текст и (или) рисунок (номер страницы, дата печати документа, эмблема организации, название документа, имя файла, фамилия автора и т. п.), который должен быть напечатан внизу или вверху каждой страницы документа. Чтобы начать работу с колонтитулом, выберите в меню **Вид** команды **Нижний колонтитул, Верхний колонтитул**.

Вставка объектов

Текстовый процессор *Writer* имеет несколько встроенных программ, реализующих различные объекты в создаваемом документе. Доступ к ним осуществляется через **Вставка – Объект…**Рассмотрим работу с одним из них, редактором формул.

Редактор формул **Формула Math** или более современный его аналог ***MathType***) является программой, которая устанавливается при инсталляции

редактора *Writer* по желанию пользователя. Редактор имеет большой набор математических символов и позволяет изображать достаточно сложные формулы. **Формула Math** позволяет видеть формулу в процессе ее записи. После записи формулы ее размеры могут быть изменены как у обычного рисунка. Для записи формулы необходимо установить курсор в нужное место и обратиться к меню **Вставка – Объект – Формула Math.** При этом должны появиться рамка для ввода формулы и панель редактора формул, содержащая несколько рядов кнопок. Один из них определяет **палитру символов**, другие – **палитру шаблонов**. Завершение ввода формулы можно выполнить щелчком мыши вне поля ввода. **Общий порядок набора формул** заключается в выборе требуемого элемента на панели редактора формул и его уточнении в появляющемся меню. Вписывание обычных символов выполняется с клавиатуры в обозначаемые поля. При переходе от одного поля к другому и при вводе новых полей необходимо следить за положением и величиной курсора. Например, курсор для обыкновенной дроби больше, чем для ее числителя или знаменателя. Таким способом показывается место для ввода очередного символа. **Вставка пробелов** в формулах не может быть выполнена простым нажатием соответствующей клавиши. Различают несколько типов пробелов, которые имеются в палитре символов (рис. 8).

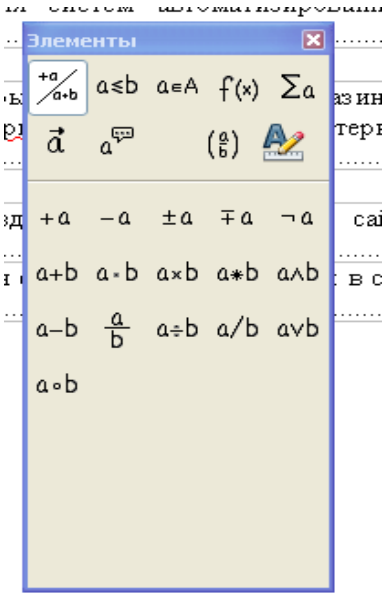


Рис. 8. Окно редактора формул.

Для вставки символов или набора простейших формул следует обратиться к значку **Ω** на панели инструментов (или воспользоваться меню **Вставка – Специальные символы**) (рис. 9).

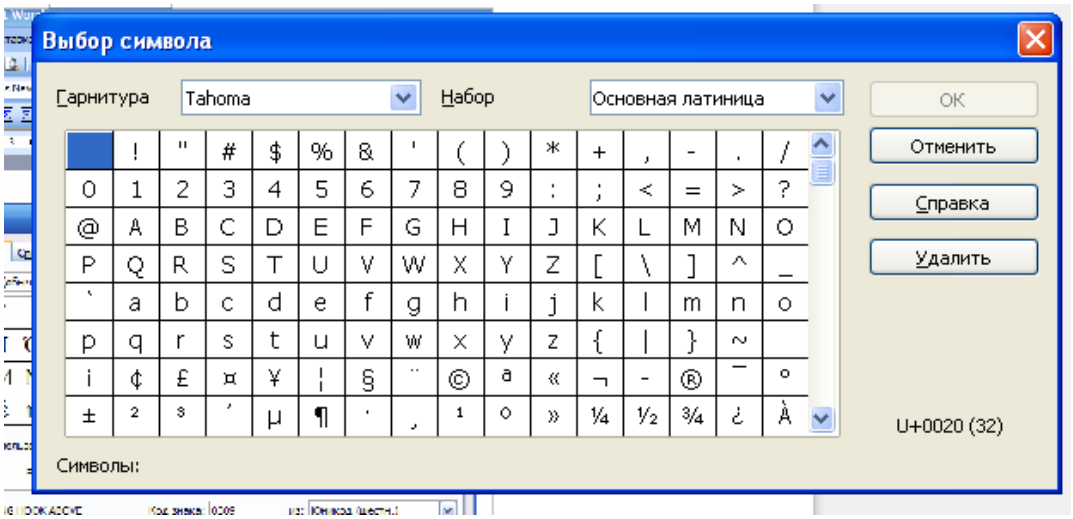


Рис. 9. Окно, содержащее символы для вставки

Программа LibreOffice Writer предназначена для выполнения работ по созданию документов, включающих разнообразные элементы (рисунки, графики, формулы, обычные или электронные таблицы, фрагменты БД и т. д.), имеющие иерархическую организацию (главы, части, параграфы и т. п.) с обеспечением работы на уровне отдельных компонентов, документа в целом, объединяющих информацию нескольких файлов в виде главного документа.

Абзац — это поле документа, набор и котором ведется без нажатия клавиши Enter. В текстовом процессоре Writer текст — это последовательность абзацев, разделяющихся специальным символом «конец абзаца», который вводится при нажатии клавиши Enter. Переход на следующую строку внутри абзаца происходит автоматически при полном заполнении текущей строки.

Страница характеризуется размером бумажного листа и параметрами размещения текста: полями, отступами от колонтитулов, способами вертикального выравнивания, ориентации текста. Набор текста в Writer осуществляется в режиме автоматизированной верстки страниц. Размер строки зависит от параметров абзаца и формата символов (меню Формат, команды Абзац и Шрифт). Размер страницы определяется параметрами команды Параметры страницы (меню Файл).

Раздел — это область документа, которая характеризуется определенным форматом печатной страницы; видом и содержанием колонтитулов; способом нумерации страниц; видом сносок в тексте и т. д. Количество разделов в документе не ограничено.

Для создания и редактирования документов используются элементы дружественного интерфейса: различные окна, меню, панели инструментов, система помощи и др.

С понятием форматирования документа в Writer связывается операция форматирования абзацев.

К форматированию абзаца относятся следующие категории: абзац, табуляция, обрамление и заливка, нумерация, кадр. Абзац может содержать текст, графику, таблицы и другие объекты. Форматирование абзаца выполняется командой Абзац меню Формат. Отдельные установки формата могут быть

сделаны также с помощью кнопок панели форматирования и горизонтальной линейки форматирования.

Диалоговое окно команды Абзац содержит две вкладки и позволяет задать следующие параметры:

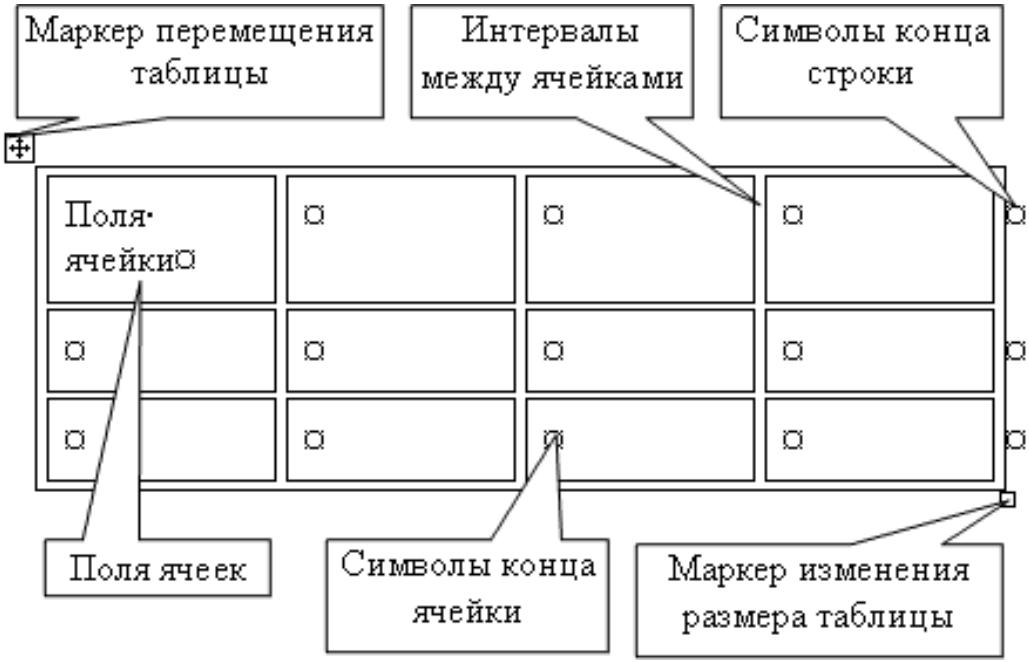
- отступ — горизонтальное расположение текста относительно полей документа;
- интервал перед — дополнительный вертикальный интервал, вставляемый перед абзацем;
- интервал после — дополнительный вертикальный интервал, вставляемый после абзаца;
- межстрочный интервал — вертикальное расстояние между строками текста в абзаце;
- выравнивание — выравнивание текста абзаца;
- запрет висячих строк — предотвращает вывод одиночной последней строки абзаца в начале новой страницы или вывод одиночной первой строки абзаца в конце страницы;
- не разрывать абзац — все строки абзаца должны быть выведены на одной странице;
- не отрывать от следующего — не позволяет вставлять разрыв страницы между данным и следующим абзацем;
- с новой страницы — абзац выводится с новой страницы;
- запретить нумерацию строк — если строки документа нумеруются, абзац исключается из нумерации;
- запретить автоматический перенос слов — к абзацу не применяется автоматическая расстановка переносов.

Левым отступом абзаца называется расстояние, на которое левая граница текста абзаца отстоит от левого поля страницы (положительный отступ) или на которое она заходит в левое поле (отрицательный отступ). То же самое относится к правому отступу абзаца. Полями называются расстояния между областью, в которой может располагаться текст, и краями бумаги; размер полей устанавливается при настройке параметров страницы. Межстрочным интервалом называется суммарная высота каждой строки текста строки в абзаце.

Перед непосредственным форматированием абзаца нужно выделить его целиком или частично. Чтобы отформатировать несколько соседних абзацев, нужно выделить хотя бы некоторую часть каждого из них.

Таблица Writer состоит из строк и столбцов ячеек. Таблицы Writer могут содержать цифры, текст и рисунки. Таблицы Writer используются для упорядочения и представления данных. Они позволяют выстроить числа в столбцы, а затем отсортировать их, а также выполнить различные вычисления.

Компоненты таблицы Writer



Границы и линии сетки

Таблица Writer имеет границу в виде тонкой сплошной линии черного цвета. Граница сохраняется при печати, а в случае удаления границы линии сетки отображаются на экране. Удаление (восстановление) границы осуществляется командой Формат / Границы и заливка, на вкладке Границы или командой Внешние границы на панели инструментов. Линии сетки не печатаются, но их тоже можно удалить (восстановить) командой Таблица / Скрыть сетку (Отображать сетку).

Концевые символы

Символ ячейки и символ строки являются непечатаемыми знаками, которые обозначают, соответственно, конец ячейки и конец строки.

Поля ячеек и интервалы между ячейками

Поля ячеек – это расстояние между границей ячейки и текстом внутри ячейки. Интервалы между ячейками и поля ячеек можно изменить в окне диалога Параметры таблицы, которое можно вызвать командой Таблица / Свойства таблицы, нажав кнопку Параметры.

Маркер перемещения и маркер изменения размера таблицы

Маркер перемещения таблицы служит для перемещения таблицы в другое место страницы, а маркер изменения размера таблицы позволяет изменить размер таблицы.

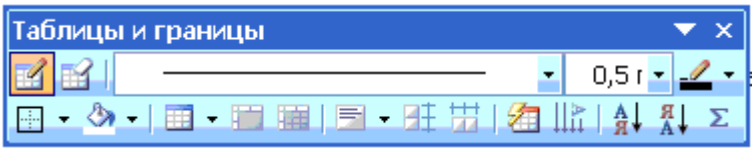
Создание таблицы Writer

Создание новой таблицы Writer можно осуществить тремя способами:

1. Нарисовать
2. Вставить
3. Создание на основе существующих данных (текста, чисел)

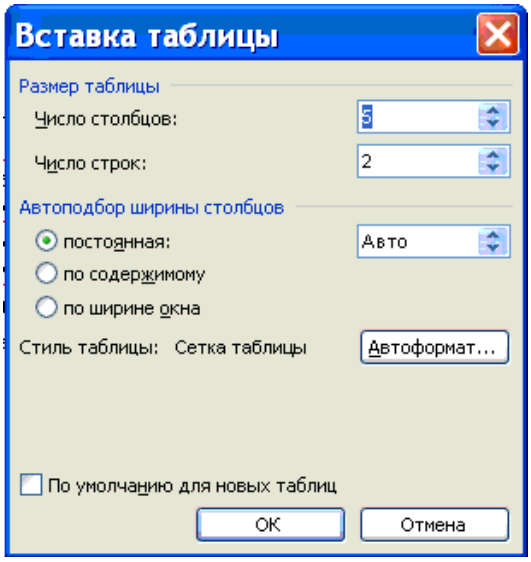
1. Нарисовать (создать) таблицу Writer

Для создания таблицы Writer со сложным заголовком целесообразно использовать способ Нарисовать таблицу. Для этого надо выбрать команду Таблица / Нарисовать таблицу. Появится плавающая панель инструментов Таблицы и границы. С помощью этой панели можно создать таблицу и осуществить ее редактирование и форматирование.



2. Вставка (создание) таблицы Writer

Чтобы быстро создать простую таблицу в Writer, необходимо воспользоваться командой Таблица/Вставить/Таблица. Появится диалоговое окно Вставка таблицы.



В этом окне можно задать размер (количество столбцов и строк), установить Автоподбор ширины столбцов, выбрать один из вариантов Автоформатирование. Установив флажок По умолчанию для новых таблиц, можно сохранить выбранный формат и в дальнейшем использовать его по умолчанию.

3. Преобразование существующего текста в таблицу

При преобразовании текста в таблицу необходимо указать, в каком месте должен начинаться каждый столбец. Для этого используют символы разделителей. В качестве разделителя может быть выбран знак абзаца, знак табуляции, точка с запятой или другой.

Ввод текста в ячейку

Для ввода текста в ячейку, необходимо щелкнуть на ячейке и ввести текст с клавиатуры или вставить из буфера обмена при копировании текста. Если текст не помещается в строке , то он переносится на другую строку и увеличивает высоту строки.

Для изменения ориентации текста в ячейке необходимо установить курсор в ячейку и в меню Формат выбрать команду Направление текста. Для изменения

выравнивания текста в ячейке на панели инструментов Таблицы и границы выберите параметр выравнивания по вертикали и горизонтали.

Для перемещения, копирования и удаления текста в ячейках необходимо выделить этот текст. Выделенный текст можно удалять клавишей Delete или Backspace, а также копировать и перемещать как с помощью буфера обмена, так и методом перемещения при помощи мыши (при нажатой левой или правой клавиши).

Форматирование текста в ячейках осуществляется методами форматирования обычного текста. Добавить текст перед таблицей в начале страницы можно, если установить курсор в начале первой строки и нажать клавишу Enter.

Редактирование таблиц Writer

К операциям редактирования таблиц Writer относится:

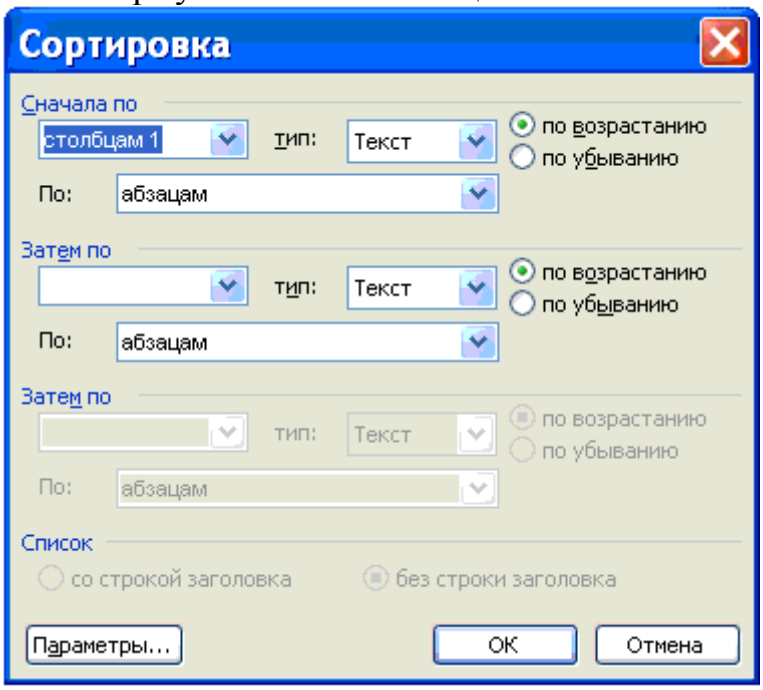
- Вставить и удалить строки и столбцы
- Объединить и разбить ячейки
- Разбить таблицу

Для редактирования элементов (ячеек, строк, столбцов) необходимо выделить эти элементы, а затем использовать меню Таблица или контекстное меню.

Для форматирования таблицы используется команда Автоформат в меню Таблицы, а также панель инструментов Таблицы и границы.

Электронные таблицы Writer

С помощью таблиц Writer можно решить некоторые задачи, которые характерны для электронных таблиц. К этим задачам относятся различные вычисления и сортировка элементов таблицы. Эти задачи выполняются командами Сортировка и Формула в меню Таблицы.



Рассмотрим способы вычислений в таблице Writer

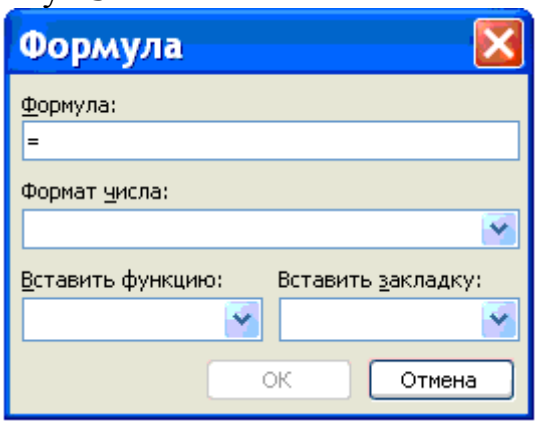
1. Сумма строки или столбца чисел

- Выделить ячейку, в которой будет отображаться сумма
- В меню Таблица необходимо выбрать команду Формула
- Если выделенная ячейка находится в самом низу столбца чисел, Writer выводит формулу =SUM(ABOVE), а если выделенная ячейка находится с правого края строки чисел, Writer предлагает формулу =SUM(LEFT).

2. Выполнение вычислений в таблице Writer

- Выделить ячейку, в которую будет помещен результат
- В меню Таблица необходимо выбрать команду Формула
- Если Writer предлагает формулу, которая не подходит для вычислений, то ее необходимо удалить
- В списке "Вставить функцию" выберите функцию. Для ссылки на ячейки введите в формулу адреса этих ячеек в скобках, например для суммирования содержимого ячеек B5 и C7 введите формулу =SUM(b5,c7).

При изменении ссылок на ячейки результаты вычислений можно обновить, выделив поле и нажав клавишу F9



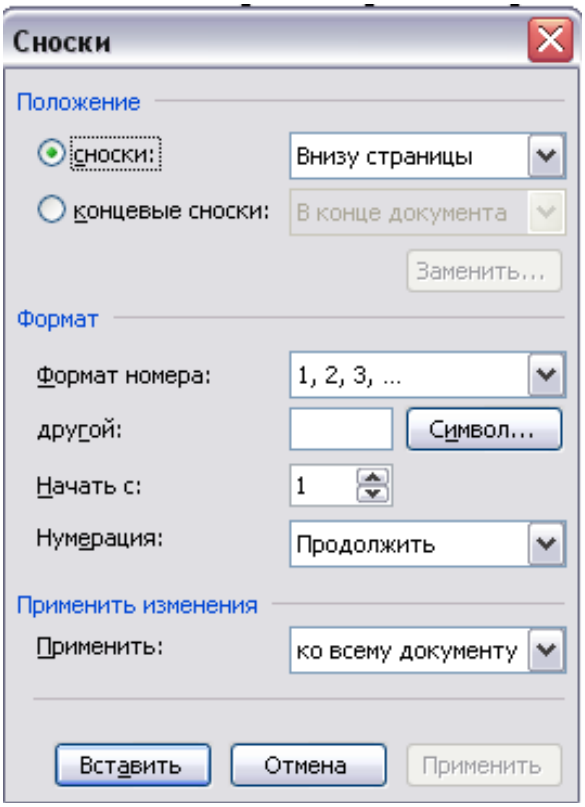
Создание сносок, ссылок и оглавлений.

Сноски бывают двух видов: обычные сноски и концевые.

Обычные сноски вставляются в конец страницы, концевые в конец документа.

Сноски находятся в меню Вставка→Ссылки→Сноски.

Так выглядит Окно «Сноски»:



Перекрестные ссылки

Перекрестные ссылки предназначены для связи объекта ссылки с абзацем, заголовком структурной части документа, закладкой, сноской, рисунком, таблицей, формулой. В зависимости от типа объекта изменяется содержание ссылки.

Например, если объект – таблица, то перекрестная ссылка может включать название таблицы, стандартное слово **Таблица** и номер таблицы, номер страницы, на которой расположена таблица, а также указатель типа выше/ниже относительно перекрестной ссылки. Наиболее часто перекрестные ссылки используются в главных документах, объединяющих вложенные документы.

Порядок создания перекрестной ссылки:

1. Создать объекты ссылки: закладки, названия таблиц, формул, рисунков и т. п.
2. Установить курсор в место вставки ссылки.
3. Ввести поясняющий ссылку текст, например, Данные приведены в, См.
4. Выполнить команду меню **Вставка → Перекрестная ссылка**:

- выбрать **Тип ссылки** (объект) и объект ссылки;
- выбрать вид ссылки – **Вставить ссылку на**;
- указать способ вставки – **Как гиперссылка**;
- добавить слова «**выше**» или «**ниже**»;
- нажать кнопку **Вставить**.

При изменении местоположения объекта, на который сделана ссылка, автоматически изменяется перекрестная ссылка. Перекрестную ссылку можно удалить после ее выделения – клавиша **Del**.

Для смены объекта, на который сделана перекрестная ссылка, следует выделить перекрестную ссылку и повторно выполнить команду меню **Вставка → Перекрестная ссылка**, изменить ее параметры.

Форматирование таблицы в Writer

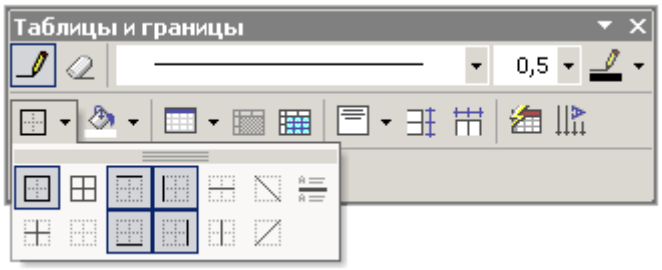
Давайте немного приукрасим таблицу, изменив линии ячеек.

Образец

	Янв	Фев	Мар	Итого
Север	7	7	5	19
Запад	6	4	7	17
Юг	8	7	9	24
Итого	21	18	21	60

1. Выберите в раскрывающемся списке Тип линии панели инструментов Таблицы и границы двойную линию.

2. Щелкните в той же панели инструментов на стрелке справа от кнопки настройки границ. Раскрывшаяся палитра показывает, что активная ячейка не имеет границ, выполненных двойной линией.



Настройка границ таблицы в Writer

Примечание Окно диалога Автоформат, в приложении Writer XP имеет более сложную структуру, чем в предыдущих версиях Office. Число имеющихся готовых вариантов формата таблица здесь значительно больше, чем раньше. Кроме того, появилась возможность самостоятельно создавать и изменять стили таблиц.

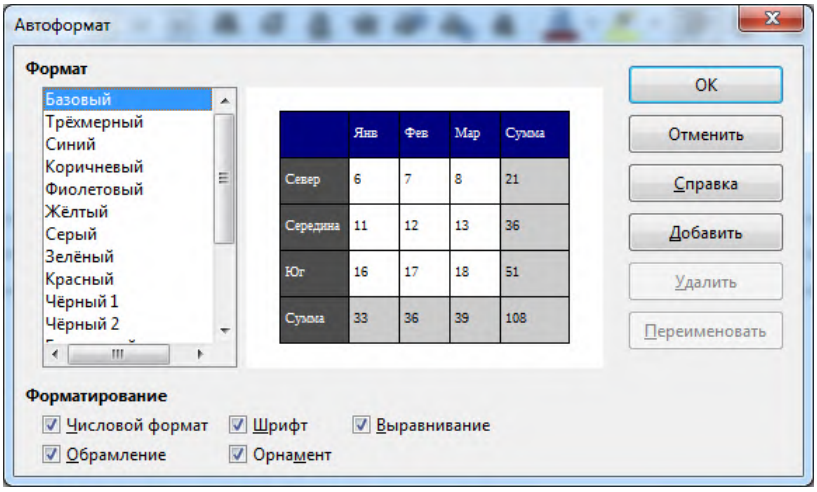
3. Щелчком на кнопке Нижняя граница замените нижнюю одинарную границу ячейки на двойную (рис. 1).

4. Выделите всю таблицу, в списке Тип линии выберите одинарную линию.

5. В списке **Толщина линии** выберите толщину 3 пункта.

6. Щелчком на кнопке **Внешние границы** палитры настройки грат ниц очертите таблицу жирной линией. Ручной настройкой границ ячеек можно создать сколь угодно сложное оформление таблицы. Но, как правило, пользователей устраивает один из стандартных стилей, предлагаемых инструментом автоформата таблиц.

7. Щелкните в любой точке, не выходящей за пределы таблицы и выберите команду **Таблица > Автоформат**. Откроется окно диалога, показанное на рис. 2.

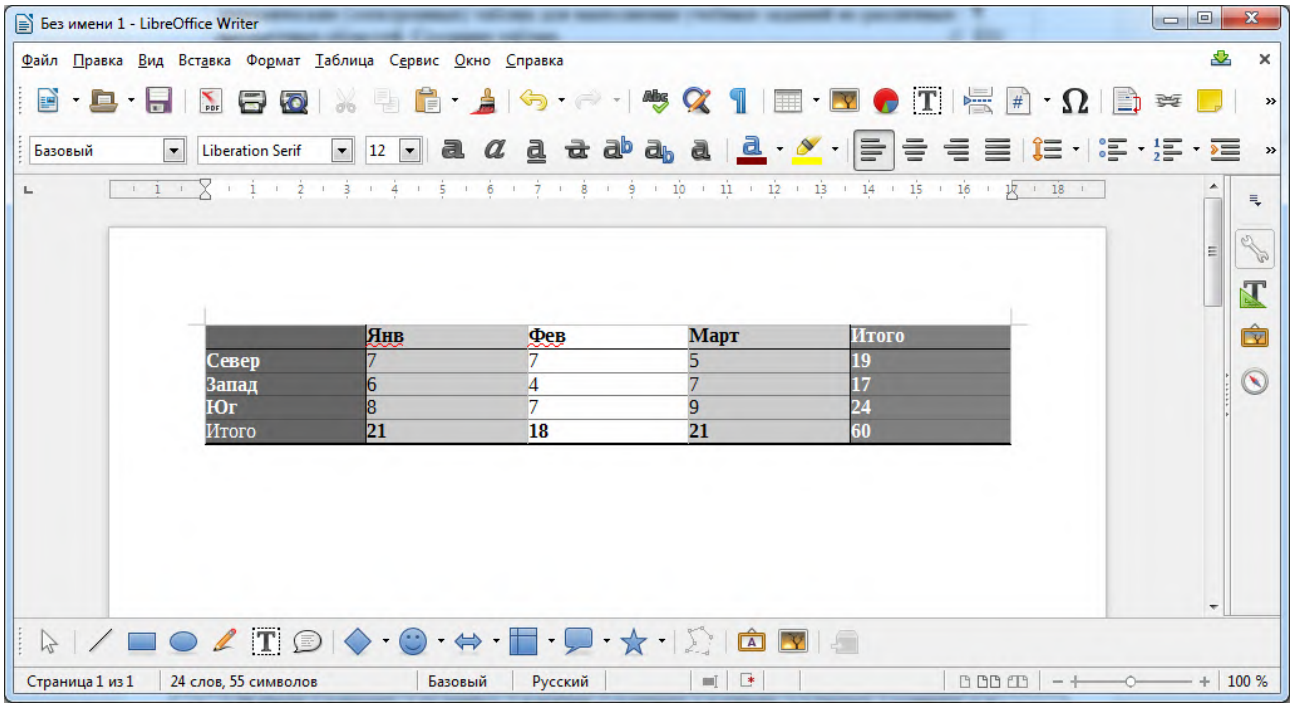


Выбор автоформата таблицы в Writer

8. Перебирайте пункты списка **Форматы** и наблюдайте предлагаемое оформление в разделе **Образец** окна диалога.

Отыскав подходящий вариант, попробуйте установить или сбросить некоторые флажки двух нижних разделов окна диалога. Это позволит модифицировать отдельные элементы выбранного автоформата. Затем щелкните на кнопке ОК.

9. Для таблицы этого упражнения можно выбрать автоформат **Серый** и не менять состояние флажков. Результат такого форматирования таблицы показан на рис. 3. Он не совсем удачен, так как таблица имеет сложную структуру заголовков и ячеек. Для более простых таблиц автоформат дает прекрасные результаты.



Окончательный вид таблицы в Writer

Практическая часть

Задание 1.

Вставка различных сносок в документ.

1. Откройте документ со своими лабораторными работами.
2. Поставьте курсор после заголовка Лабораторная работа № 14.
3. Вызовите меню Сноска (Вставка→Ссылки→Сноски).
4. Установите положение СНОСКА внизу страницы.
5. Формат номера римские цифры и нажмите кнопку вставить.
6. Курсор автоматически перейдет в конец страницы. Напишите текст сноски: «Моя лабораторная работа № 14».
7. Перейдите к заголовку третьей лабораторной работы (поставьте курсор в конец заголовка).
8. Вызовите меню Сноска.
9. Выберите положение в конце документа, формат номера ДРУГОЙ и, нажав кнопку символ, выберите символ ♣ и нажмите кнопку ВСТАВИТЬ.
10. В текст сноски введите: «Это моя третья лабораторная работа в WRITER»

Задание 2

Создать таблицы:

№ п/п	r_x	$cos\ \alpha_x$	α_x	$inv\ \alpha_x$	$\frac{1}{2}\psi_x$*	$sin\ \psi_x$	x_c	$cos\ \psi_c$	y_c
1	78,20119	0,931267	21,366814	0,018306	0,028197	0,028193	2,204747	0,999602	0
2	79	0,921850	22,801944	0,022432	0,032323	0,032317	2,553517	0,999448	0,788595
3	80	0,910327	24,449420	0,027936	0,037827	0,037818	3,025438	0,999285	1,772668
4	81	0,899089	25,961422	0,033787	0,043687	0,043673	3,536793	0,999046	2,752622
5	82	0,888124	27,361552	0,039951	0,049682	0,049662	4,085353	0,998758	3,728064
6	83	0,877424	28,866834	0,046402	0,056293	0,056263	4,669852	0,998416	4,761421
7	84	0,866978	29,890660	0,053118	0,063009	0,062926	5,289255	0,998016	5,726200
8	85	0,856779	31,043170	0,060181	0,069972	0,069915	5,942768	0,997553	6,684897
9	86	0,846816	32,132970	0,067273	0,077164	0,077087	6,629520	0,997024	7,636989
10	87	0,837083	33,166640	0,074683	0,084578	0,084477	7,349170	0,996426	8,582299
11	87,95119	0,828029	34,103202	0,081920	0,091811	0,091682	8,063547	0,995788	9,473665

Просроченная задолженность потребителей электроэнергии и топлива в общем объеме неплатежей покупателей промышленной продукции в 2017 г.

(млрд. руб.)

A	B	C	D	E	F	G
	Просроченная задолженность покупателей промышленной продукции			*Удельный вес просроченной задолженности в общем объеме неплатежей покупателей промышленной продукции*		*Темпы роста*
	всего	*в том числе:*				
		потребителей электроэнергии	*потребителей топлива*	*потребителей электроэнергии*	*потребителей топлива*	
янв. 17	96029,0	29451,8	22751,4			1,025
фев. 17	98649,1	36049,6	21161,1			
мар. 17	111831,0	44540,3	23854,2			
	12435,0	53072,8	24410,3			
	137332,0	59141,7	26338,0			
	138715,0	59628,0	26917,5			
	143421,0	61437,1	27457,3			
	150225,0	63225,2	30470,5			
	156363,0	63808,3	32515,9			
	161794,0	64548,9	33838,9			
	165988,0	66197,3	35063,0			
	170434,0	68135,5	36836,6			

среднее арифметическое		
минимальное		
максимальное		
средний темп роста		

2. Заполните свободные ячейки таблицы данными.
   - При вводе дат используйте заполнение серий данных.
   - Рассчитайте «Удельный вес», как отношение соответствующего показателя к общей сумме (графа «Всего»).
   - Рассчитайте «Темпы роста», как отношение текущего значения к предыдущему.
   - Для полученных расчетных рядов рассчитайте «среднее арифметическое», «минимальное» и «максимальное» значение ряда.
   - Рассчитайте «средний темп роста» показателя «Задолженность всего», как «среднее геометрическое» ежемесячных темпов роста.

3. Установите следующие форматы данных:
   - «Удельный вес» - формат «Процент» с двумя знаками после запятой.
   - «Темпы роста» - формат «Число» с тремя знаками после запятой. При этом незначащие нули, кроме первых двух после запятой должны быть погашены и числа выровнены по десятичной запятой. Задайте в формате цвет изображаемого числа – красный (red).
   - «Среднее арифметическое», «минимальное», «максимальное» - формат «Число» с двумя знаками после запятой.
     «Средний темп роста» - формат «Число» с тремя знаками после запятой.

Отчет должен содержать:

1. Название работы.
2. Цель работы.

3. Задание.
4. Результаты выполнения задания.
5. Вывод по работе.

Практическая работа № 5

Тема работы: Создание электронных таблиц. Выполнение расчетов Расчет с использованием встроенных Построение диаграмм на основе электронных таблиц функций

Цель работы: освоение принципов построения электронной таблицы. Ввод чисел, текста, формул, копирование формул, редактирование и пересчет ячеек, сохранение таблицы на диске, открытие существующей таблицы, закрытие таблицы без записи на диск, знакомство с абсолютной адресацией ячеек, добавление строк и столбцов, удаление строк, редактирование формул, сортировка табличных данных, функция ЕСЛИ, освоение приемов построения диаграмм и графиков

Оборудование, приборы, аппаратура, материалы: персональный компьютер, программа LibreOffice Calc.

1 Краткие теоретические сведения.

Что такое Calc?

Calc – это табличный процессор. Табличный процессор - это прикладная програм-ма, которая предназначена для создания электронных таблиц и автоматизированной обра-ботки табличных данных.

Что такое электронная таблица в Calc?

Электронная таблица – это электронная матрица, разделенная на строки и столбцы, на пересечении которых образуются ячейки с уникальными именами. Ячейки являются основным элементом электронной таблицы, в которые могут вводиться данные и на кото-рые можно ссылаться по именам ячеек. К данным относятся: числа, даты, время суток, текст или символьные данные и формулы.

Что такое обработка табличных данных в Calc?

К обработке данных относится:

•проведение различных вычислений с помощью формул и функций, встроен-ных в редактор;

•построение диаграмм;

•обработка данных в списках (Сортировка, Автофильтр, Расширенный фильтр, Форма, Итоги, Сводная таблица);

•решение задач оптимизации (Подбор параметра, Поиск решения, Сценарии "что - если" и другие задачи);

•статистическая обработка данных, анализ и прогнозирование (инструменты анализа из надстройки "Пакет анализа").

Таким образом, Calc являются не только средством автоматизации расчетов, но и средством моделирования различных ситуаций.

Область применения Calc: планово – финансовые и бухгалтерские расчеты, учет материальных ценностей, системы поддержки принятия решений (СППР) и другие обла-сти применения.

Создание новой рабочей книги в Calc

Обучение работе с Calc необходимо начинать с изучения окна приложения Calc. При запуске Calc открывается окно приложения, в котором отображается новая рабочая книга – Без имени 1.

Окно приложения Calc имеет пять основных областей: строка меню; •панели инструментов; •строка состояния; •строка ввода; •область окна
1 рабочей книги.

Основная обработка данных в Calc осуществляется при помощи команд из строки меню. Панели инструментов Стандартная и Форматирование являются встроенными панелями Calc, которые располагаются под строкой меню и содержат определенные наборы пиктограмм (кнопок). Основная часть пиктограмм предназначена для выполнения наиболее часто используемых команд из строки меню.

Строка формул в Calc используется для ввода и редактирования значений, формул в ячейках или диаграммах. Поле имени – это окно слева от строки формул, в котором выводится имя активной ячейки.

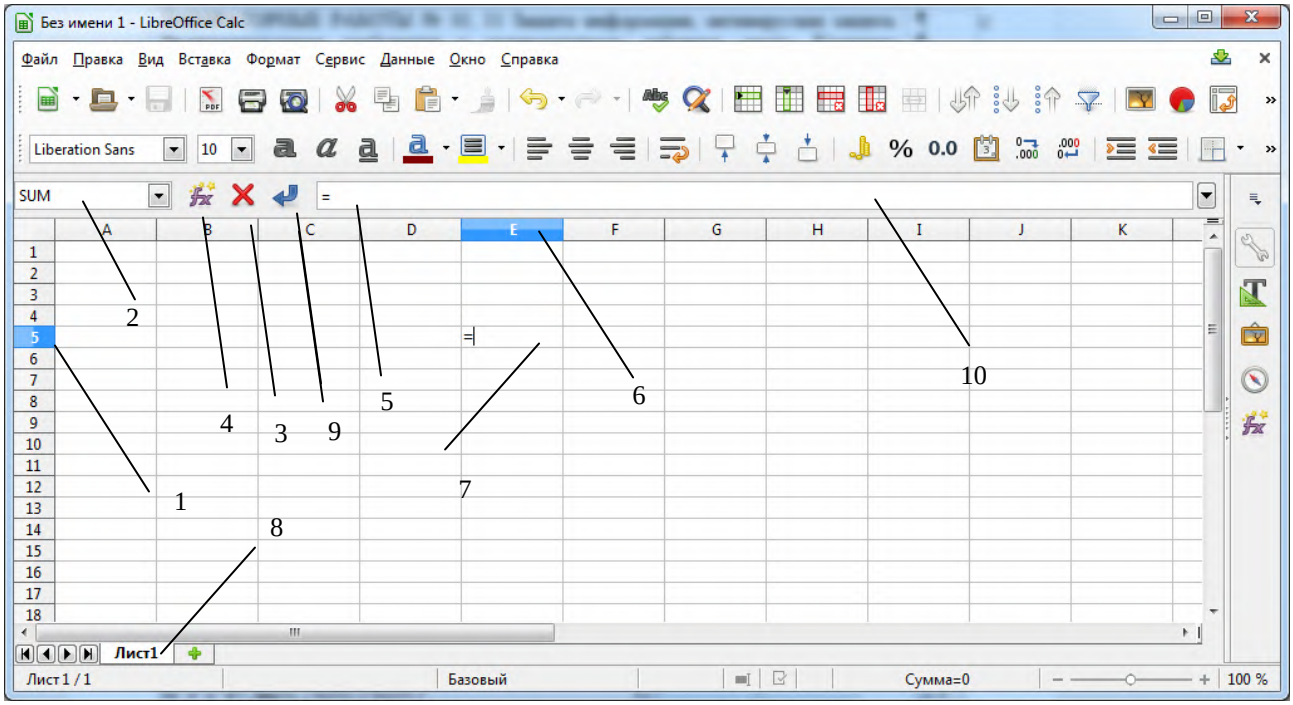


Рисунок - Рабочее окно LibreOffice *Calc:* 1 – заголовок строки; 2 – указатель адреса или имени ячейки; 3 – отмена редактирования; 4 – мастер функций; 5 – содержимое ячейки; 6 – заголовок столбца; 7 – указатель ячейки; 8 – ярлык листа; 9 – подтверждение редактирования; 10 – строка формул

Строка состояния окна приложения Calc расположена в нижней части экрана. Левая часть строки состояния указывает сведения о состоянии рабочей области электронной таблицы (Готово, Ввод, Правка, Укажите). Кроме того, в левой части строки состояния кратко описываются результаты выполненной команды. В правой части строки состояния выводятся результаты вычислений (при выполнении автоматических вычислений с помо-щью контекстного меню

строки состояния) и отображаются нажатые клавиш Ins, Caps Lock, Num Lock, Scroll Lock.

Далее необходимо ознакомиться с основными понятиями окна рабочей книги. Рабочая книга (документ Calc) состоит из рабочих листов, каждый из которых является элек-тронной таблицей. По умолчанию открывается три рабочих листа или три электронных таблицы, переход к которым можно осуществить, щелкая на ярлычках, расположенных внизу книги. При необходимости в книгу можно добавить рабочие листы (электронные таблицы) или удалить их из книги.

Кнопки прокрутки ярлычков осуществляют прокрутку ярлычков рабочей книги. Крайние кнопки осуществляют прокрутку к первому и последнему ярлычку рабочей книги. Внутренние кнопки осуществляют прокрутку к предыдущему и следующему ярлычку рабочей книги.

Основные понятия электронной таблицы: заголовок столбца, заголовок строки, ячейка, имя ячейки, маркер выделения, маркер заполнения, активная ячейка, строка формул, поле имени, активная область листа.

Рабочая область электронной таблицы состоит из строк и столбцов, имеющих свои имена. Имена строк – это их номера. Нумерация строк начинается с 1 и заканчивается максимальным числом, установленным для данной программы. Имена столбцов – это буквы латинского алфавита сначала от A до Z , затем от AA до AZ, BA до BZ и т.д.

Максимальное количество строк и столбцов электронной таблицы определяется особенностями используемой программы и объемом памяти компьютера, например, в табличном процессоре Calc 256 столбцов и более 16 тысяч строк.

Пересечение строки и столбца образует ячейку электронной таблицы, имеющую свой уникальный адрес. Для указания адресов ячеек в формулах используются ссылки (например, A6 или D8).

Ячейка – область, определяемая пересечением столбца и строки электронной таблицы, имеющая свой уникальный адрес.

Адрес ячейки определяется именем (номером) столбца и именем (номером) строки, на пересечении которых находится ячейка, например, A10. Ссылка – указание адреса ячейки.

Активная ячейка - это выделенная ячейка, имя которой отображается в поле имени.

Маркером выделения называется полужирная рамка вокруг выделенной ячейки. Маркер заполнения - это черный квадрат в правом нижнем углу выделенной ячейки.

Активная область листа - это область, которая содержит введенные данные.

В электронных таблицах можно работать как с отдельными ячейками, так и с группами ячеек, которые образуют блок.

Блок ячеек – группа смежных ячеек, определяемая с помощью адреса.

Адрес блока ячеек задается указанием ссылок первой и последней его ячеек, между которыми ставится разделительный символ – двоеточие. Если блок

имеет вид прямоугольника, то его адрес задается адресами левой верхней и правой нижней ячеек, входящих в блок.

Блок используемых ячеек может быть указан двумя путями: либо заданием с клавиатуры начального и конечного адресов ячеек блока, либо выделением соответствующей части таблицы при помощи левой клавиши мыши.

Пример задания адресов ячейки и блоков в электронной таблице:

•адрес ячейки, находящейся на пересечении столбца F и строки 9, выражается ссылкой F9;

•адрес блока, образованного в виде части строки 1 - B1:E1;

•адрес блока, образованного в виде столбца C - C1:C21;

•адрес блока, образованного в виде прямоугольника - A3:G10

Текст - это любая последовательность введенных в ячейку символов, которая не может быть интерпретирована Calc как число, формула, дата, время суток. Введенный текст выравнивается в ячейке по левому краю.

Числовые данные – это числовые константы: 0 - 9, +, -, /,*, E, %, точка и запятая. При работе с числами необходимо уметь изменять вид вводимых чисел: число знаков после запятой, вид целой части, порядок и знак числа.

Формулы – это выражение, начинающееся со знака равенства и состоящее из числовых величин, адресов ячеек, функций, имен, которые соединены знаками арифметических операций. К знакам арифметических операций, которые используются в Calc относятся: сложение; вычитание; умножение; деление; возведение в степень.

Функции Calc — это специальные, заранее созданные формулы для сложных вычислений, в которые пользователь должен ввести только аргументы.

Аргументы функции Calc - задают значения или ячейки, используемые функцией, они всегда заключены в круглые скобки. Открывающая скобка ставится без пробела сразу после имени функции. Например, в формуле «=СУММ(A2;A9)», СУММ — это имя функции, а A2 и A9 — ее аргументы.

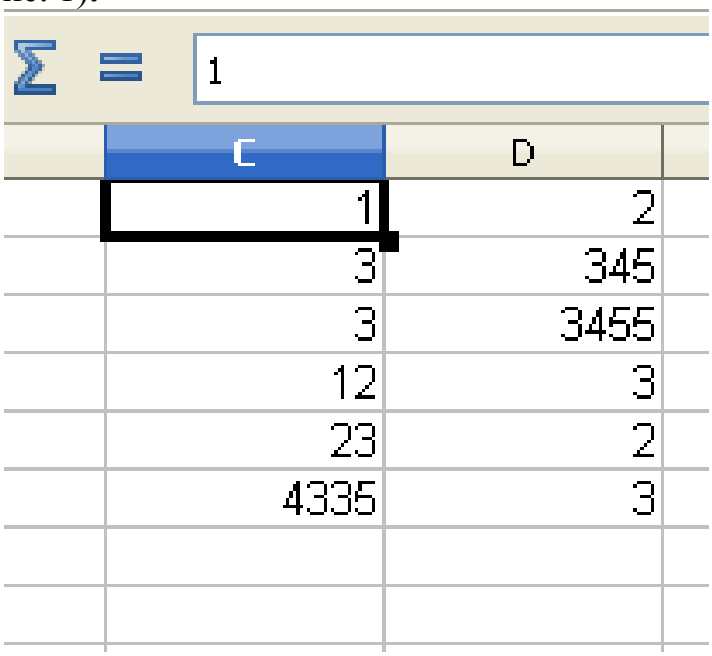
Электронная таблица состоит из строк и столбцов. Строки нумеруются, а столбцы обозначаются **латинскими** буквами. На пересечении строк и столбцов находятся **ячейки**. Каждая ячейка имеет уникальный **адрес,** состоящий из обозначения столбца и номера строки. Например: **A1, A5, B4** и т.д. В ячейке могут храниться: число, текст или *формула*. Именно благодаря тому, что в ячейке можно записывать формулы, электронные таблицы могут использоваться для выполнения расчетов. Calc по содержимому, которое пользователь вводит в ячейку, пытается определить, что в ней будет. Так, например, если первым символом вводится цифра, знак плюс или минус, то считается, что в ячейке будет записано число. Для отличия формулы от текста формулу записывают, начиная со знака равно (=). Например**:**

=1,25*(B2+C2-A2)/E2^2

означает, что нужно сложить содержимое ячеек B2 и C2, вычесть содержимое ячейки A2. Полученный результат следует умножить на коэффициент 1,25 и разделить на содержимое ячейки E2, возведенное к квадрат. В приведенном примере использованы все арифметические операции и круглые скобки, которые меняют последовательность выполнения операций. Обращаем внимание, что *целая часть от дробной* отделяется не точкой, как это принято в

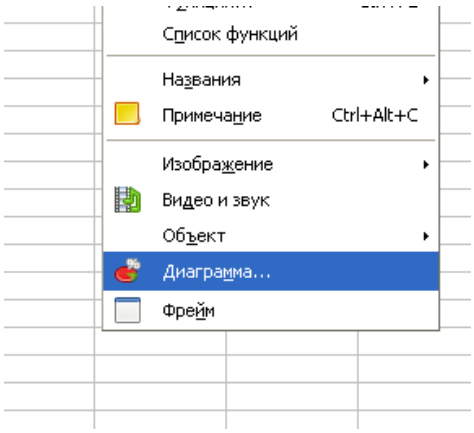
большинстве языков программирования, а *запятой* (задается в настройках Windows).

LibreOffice *Calc* обладает достаточно сильными средствами создания таблиц и диаграмм. Для наглядности результаты расчетов обычно представляют в виде графиков. Рассмотрим процедуру создания графика. Сначала создадим столбцы с данными (рис. 1).



Создание столбцов с данными для построения графика

Затем данные следует выделить. Следующим этапом – выбрать **Диаграмма** в меню **Вставка** (рис. 2). Далее запустится мастер создания диаграмм. Работа этого мастера состоит из 4 шагов. На первом шаге нужно выбрать тип диаграммы, которых огромное количество. Для определенности выберем **точечную**. Этот тип диаграммы позволяет проследить связь между двумя параметрами. На втором шаге следует определить ряды. В большинстве случаем, мастер все делает автоматически. Но если что-то по умолчанию не правильно определено, то, очевидно, нужно доопределить. В этом случае на третьем шаге можно создать подписи, а на четвертом указать, куда их вставлять. Созданная диаграмма связана с данными и при изменении последних редактируется и диаграмма. Готовую диаграмму можно форматировать, например, используя контекстное меню.



II Практическая часть

Задание 1

1. Проверьте наличие на диске вашей папки и создайте ее, если она отсутствует. Для этой цели воспользуйтесь программой **Проводник**.

2. Вызовите Calc и ознакомьтесь с меню и панелями инструментов, которые выведены на экран. Состав панелей инструментов у каждого пользователя может быть свой. Управления панелями инструментов осуществляется через п. меню **Вид – Панели инструментов**.

3. Перемещайте курсор по панели инструментов и задерживайте его на некоторое время у каждого инструмента. Обратите внимание, что при этом появляется краткая подсказка о назначении инструмента.

4. Перемещаясь по меню, ознакомьтесь с составом его пунктов. Познакомьтесь со справочной системой, если вы еще с ней не знакомы (приемы работы с файлами помощи во всех приложения Windows однообразны).

Создание таблицы

5. Начиная с клетки **A1**, создайте следующую электронную таблицу (рис. 1). Названия столбцов таблицы и фамилии сотрудников, числа и формулы вводите с первой позиции. Обратите внимание, что числа автоматически будут выровнены по правой границе, а текст - по левой. В колонке D запишите формулы. На рисунке показана формула для ячейки D2

D2 = =B2+C2

	A	B	C	D
1	Фамилия	Зарплата	Премия	Всего
2	Иванов	1000	300	1300
3	Петров	800	600	1400
4	Мешков	1500	400	1900
5	Сидоров	2000	250	2250
6	Гусев	1200	850	2050
7	Бобров	1800	450	2250
8	Дунаев	2300	900	3200
9	Итого	10600	3750	14350

Диапазон **D3:D8** заполняются путем копирования формулы из ячейки D2. Для этого нужно выделить блок ячеек, начиная с ячейки D2 до D8 включительно (**Shift** и стрека вниз) и вызвать операцию **Правка – Заполнить - Вниз***)* или с использованием мыши путем протаскивания маленького квадрата в правом нижнем углу ячейки **D2**. Просмотрите формулы для всех сотрудников. Обратите внимание на автоматическое изменение адресов.

6. Клетка **B9** рассчитывается по формуле **=СУММ(B2:B8)** Ее можно ввести самостоятельно, а можно выделить диапазон ячеек **B2:B8** и нажать на инструмент со знаком суммирования Σ. В клетки **C9** и **D9** эта формула копируется (выделение блока, а затем – п. **Правка – Заполнить - Вправо** или с использованием мыши путем протаскивания вправо квадрата в правом нижнем углу ячейки. Опробуйте оба способа копирования формулы).

Сохранение таблицы

7. Сохраните таблицу на диске **в вашей папке** под именем **Задача1.XLS**. Для этого воспользуйтесь п. меню **Файл – Сохранить**.

8. Поменяйте у нескольких сотрудников заработную плату. Для редактирования содержимого ячеек используйте клавишу **F2** или дважды щелкните мышкой на соответствующей ячейке. Обратите внимание на автоматическое изменение вычисляемых ячеек.

9. Сохраните таблицу под прежним именем (п. **Файл – Сохранить** или инструмент).

10. Внесите еще несколько изменений в таблицу и постарайтесь их запомнить.

11. Закройте таблицу *без записи* ее на диск (п. **Файл - Закрыть**).

Открытие существующей таблицы

12. Откройте созданную вами таблицу (п. **Файл – Открыть** или инструмент). Обратите внимание, ваши изменения, сделанные при выполнении п.11, в таблице отсутствуют.

Сдайте работу преподавателю и закройте таблицу.

Задание 2.

1. Откройте таблицу **Задание 1** и сразу сохраните ее под именем **Задача 2**. *Напоминаем, что в данной работе внешнему виду таблицы не придавайте значения, все внимание концентрируйте на построении формул и выполняемых операциях.*

2. Вставьте три строки перед строкой заголовка таблицы. Для этого установите курсор в ячейку **A1** и используйте п. меню **Вставка – Строки**. В ячейку **A1** введите текст «**Расчетная ведомость за январь**», в ячейку **A2** – текст «**Ставка Налога**», в ячейку **A3** - «**Удерж. в ПФ**», в ячейку **B2** число **12%**, в

E5 = =D5*B2

	A	B	C	D	E	F
1	Расчетная ведомость за январь					
2	СтавкаНал	12%	15%			
3	Удерж.ПФ	1%				
4	Фамилия	Зарплата	Премия	Всего	Налог	Выплатить
5	Иванов	1000	300	1300	156	1144
6	Петров	800	600	1400	168	1232
7	Мешков	1500	400	1900	228	1672
8	Сидоров	2000	250	2250	270	1980
9	Гусев	1200	850	2050	246	1804
10	Бобров	1800	450	2250	270	1980
11	Дунаев	2300	900	3200	384	2816
12	Итого	10600	3750	14350	1722	12628

ячейку **C2** число **15%**, в ячейку **B3** –число **1%**.

3. Создайте *новые столбцы* «**Налог**» и «**Выплатить**». При этом для столбца **«Налог»** задайте формулу, в которой используйте значение ячейки **B2**,

поставив знак доллара перед обозначением столбца и перед номером строки (**B2**), как это показано на рисунке. Это будет ***абсолютная ссылка*** на ячейку. При копировании формулы для всех сотрудников *адрес этой ячейки останется без изменения*.

<u>Внимание!</u> Количество и расположение столбцов после выполнения этой лабораторной работы показано на рисунке в лаб. работе № 3.4.

4. Формулы для вычисления столбца **F** и итоговых сумм создайте самостоятельно, руководствуясь смыслом задачи.

5. *Удалите одну* строку из таблицы (уволен сотрудник). Проверьте формулы итоговой строки, обратите внимание на изменение диапазонов в формулах.

6. *Дополните таблицу еще тремя строками*, включив их между, например, 7 и 8 строками (приняты на работу три новых сотрудника). Заполните эти строки - фамилии, заработную плату и премию введите, а расчетные формулы скопируйте путем протаскивания. Проверьте итоговую строку и обратите внимание на изменение диапазона суммируемых ячеек в формулах - они соответствуют всем ячейкам, которые находятся *выше итоговой* строки, то есть программа сама откорректировала вашу формулу.

ВНИМАНИЕ, "тонкий" момент.

7. Установите курсор *на итоговую строку*. Вставьте одну пустую строку перед итоговой и заполните ее информацией. Проверьте формулы в итоговой строке и обратите внимание - *диапазоны суммируемых ячеек* ***не изменились***. У нас появился новый сотрудник, но данные по нему в итоги не входят! **<u>Запомните</u>,** *если вставка строки происходит после последней строки диапазона суммирования, то формула не изменяется, а если во внутрь диапазона - формула изменяется автоматически.* Удалите добавленную строку или исправьте формулы суммирования.

8. Вставьте *перед* колонкой «Налог» еще две новые колонки «**Удер. в ПФ»** и «**Налог. база»**. Установите, что в пенсионный фонд удерживается 1% от начисленной заработной платы и премии. При этом в формулах ссылайтесь на ячейку **B3**.

9. *Отчисления в пенсионный фонд не входят в налогооблагаемую базу.* Внесите необходимые изменения во все формулы, зависящие от этого.

10.Измените алгоритм расчета подоходного налога с учетом прогрессивной шкалы налогообложения.

<u>Примечание.</u> На самом деле подоходный налог рассчитывался с *совокупного годового* дохода в размере 13%. В нашем примере, преследуются исключительно учебные цели.

Если налогооблагаемая база меньше или равна 30 000 рублей, то принимается ставка 12%, в противном случае налог равен - 3600 плюс 15%.от суммы, которая превышает 30000 руб. Формула должна использовать функцию **ЕСЛИ**:

=ЕСЛИ(условие; формула1; формула2)

Расчет по ***формуле 1*** *выполняется в том случае, если условие* ***истинно****, в противном случае для расчета используется* ***формула 2****.*

Так, если налогооблагаемая база первого сотрудника находится в ячейке **G5**, то формула будет выглядеть следующим образом: **=ЕСЛИ(G5<30000; G5*B2; 3600+C2*(G5-30000))**

ВНИМАНИЕ! У вас могут быть другие координаты ячеек.

11.Установите 1-2 сотрудникам заработную плату более 30000. Проверьте правильность расчета налога.

12.Вставьте перед столбцом фамилий новый столбец "**Таб.номер**" и заполните (автоматически) его значениями: 100, 101,102 и т.д. Для этого, установите табельный номер первого сотрудника равным 100. Выделите диапазон, начиная с этой ячейки, до номера последнего сотрудника и воспользуйтесь п. **Правка – Заполнение -Прогрессия**.

13.Арифметическую прогрессию можно создать и другим способом. Задайте первому сотруднику табельный номер 100, второму 110 и выделите эти две ячейки. Захватив точку в правом нижнем углу выделенного диапазона, протащите значения до последнего сотрудника - получится арифметическая прогрессия с шагом 10.

14.Сохраните исправленную таблицу (п. **Файл - Сохранить**).

Сортировка таблицы

15.**Отсортируйте** таблицу в алфавитном порядке по фамилиям сотрудников. Для этого выделите фрагмент, начиная с табельного номера первого сотрудника до ячейки "Выплатить" последнего сотрудника и воспользуйтесь п. **Данные- Сортировка**. Для усвоения материала самостоятельно выполните сортировку по другим полям.

16.В окне сортировки можно задать до трех уровней сортировки. Опробуйте сортировку *по нескольким полям* одновременно, например, сделайте в таблице несколько однофамильцев с разной зарплатой и отсортируйте таблицу по возрастанию фамилии и убыванию заработной платы.

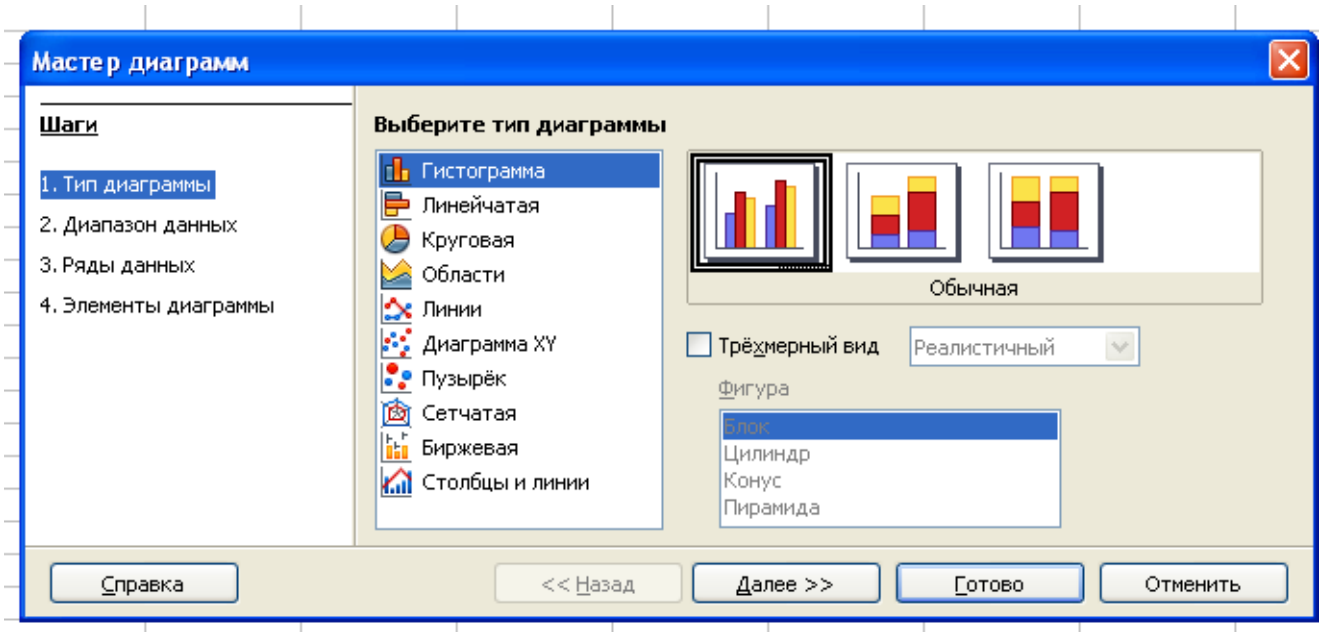
17.Сохраните исправленную таблицу в вашей папке.

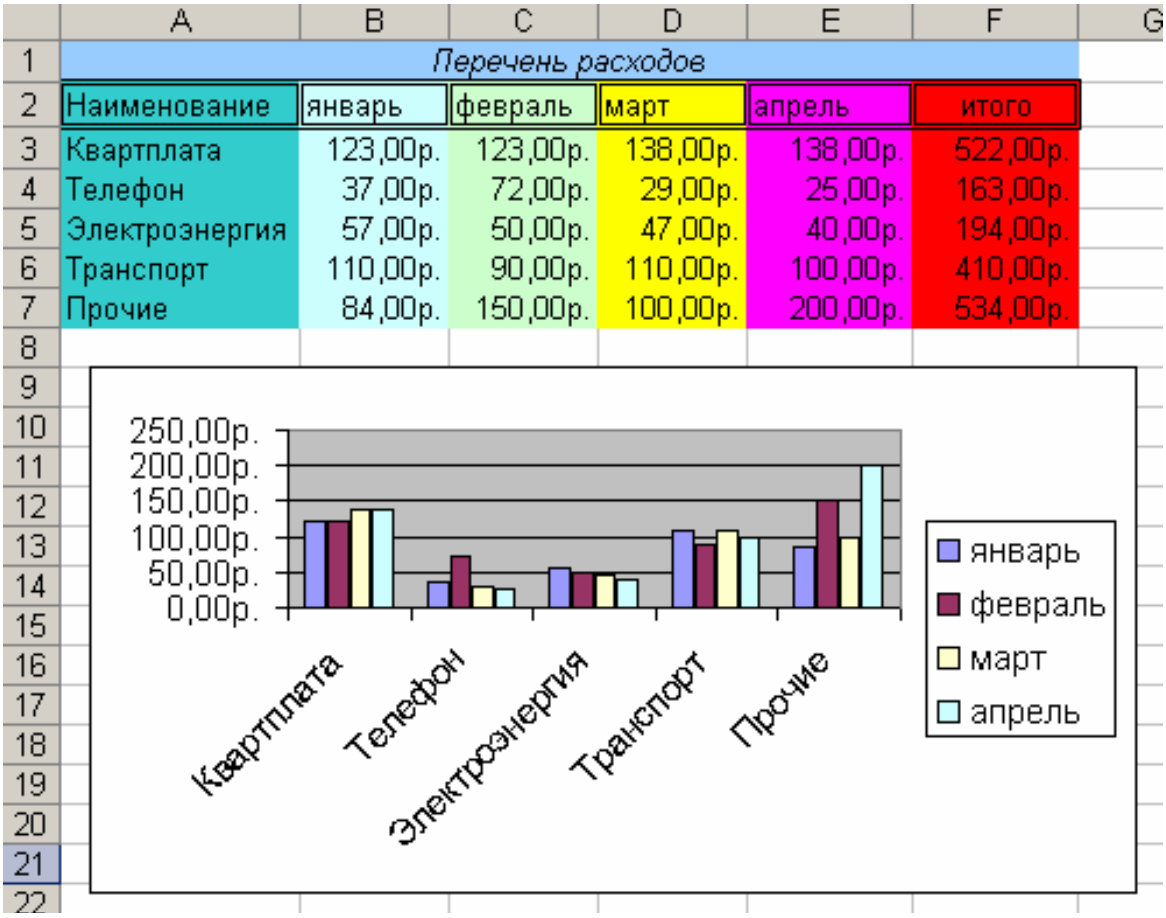
18.Сохраните таблицу и сдайте работу преподавателю.

Задание 3.

Построить таблицу, рассчитать «Итого»; построить гистограмму.

Рекомендации. Построить таблицу, оформить соответствующим образом, используя меню **Формат ячеек.** Рассчитать значения столбца «итого», используя инструмент «автосумма». Затем выделить прямоугольную область ячеек **A2:F7**. Выбрать **Гистограмма** в окне **Мастер диаграмм** (рис. 3). И остается только нажимать на кнопку **Далее,** и только в заключении, когда кнопка **Далее** станет не активной, нажать кнопку **Готово**. Гистограмма создана (рис. 4).





	A	B	C	D	E	F
1	Перечень расходов					
2	Наименование	январь	февраль	март	апрель	итого
3	Квартплата	123,00р.	123,00р.	138,00р.	138,00р.	522,00р.
4	Телефон	37,00р.	72,00р.	29,00р.	25,00р.	163,00р.
5	Электроэнергия	57,00р.	50,00р.	47,00р.	40,00р.	194,00р.
6	Транспорт	110,00р.	90,00р.	110,00р.	100,00р.	410,00р.
7	Прочие	84,00р.	150,00р.	100,00р.	200,00р.	534,00р.

Результат построения гистограммы

Задание. Построить график функции, которая задается уравнением в полярных координатах: $\rho = a\sin(k\varphi)$, *где* $a = 2$, $k = 3$. ☺☺

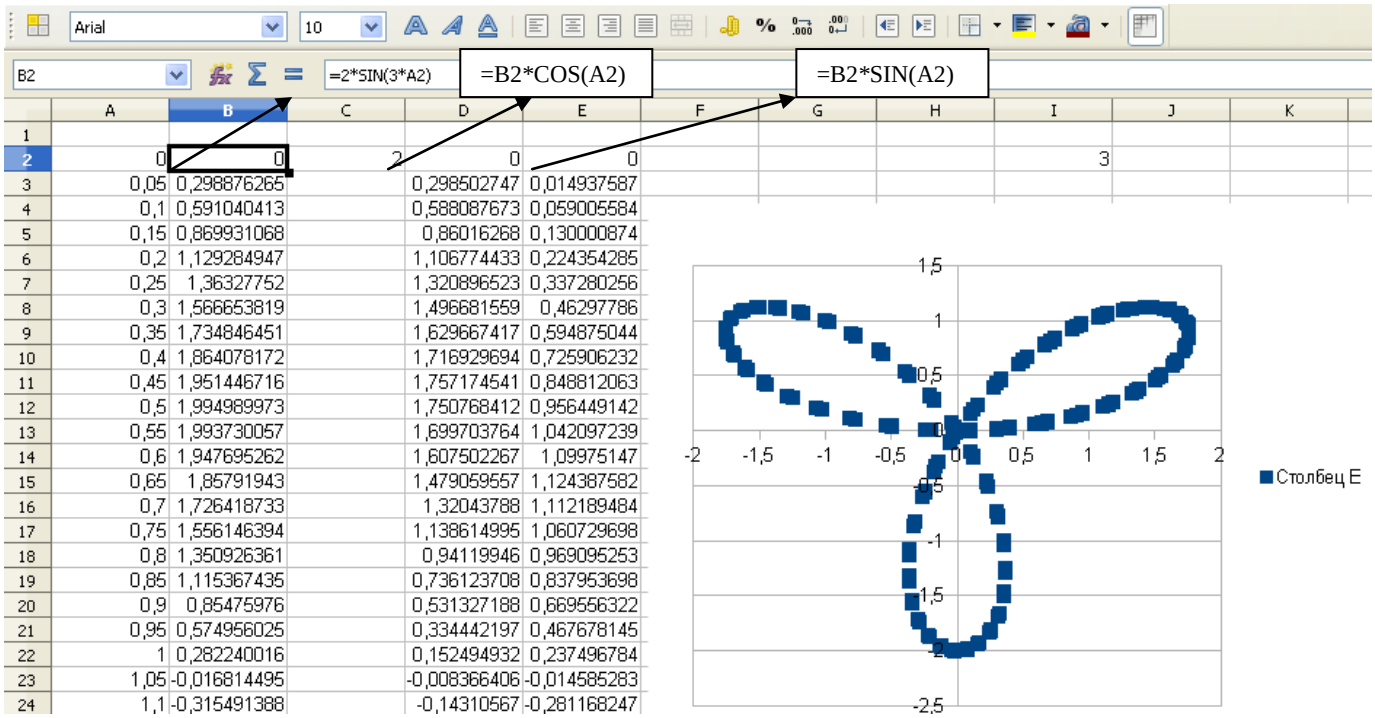
Рекомендации. Разместить в ячейке **C2** значение коэффициента a (равного 2), в ячейке **I2** – значение коэффициента k (равного 3). Подготовить диапазон изменения координаты φ. Для этого в ячейку **A2** записать 0, в ячейку **A3** записать

0,05 (в числовом формате). Затем выделить содержимое двух ячеек и произвести растаскивания (см. выше), например, до ячейки с адресом **A150**.

1. Рассчитать значение функции на данном диапазоне в полярных координатах $\rho = \rho\ (\varphi)$. Для этого в ячейку **B2** записать формулу: =2*SIN(3*A2). Вычислить и провести «растяжку» до **B150**.

2. Рассчитать значения ***x*** (столбец **D**) и ***y*** (столбец **E**) в декартовой системе координат по формулам: $x = \rho\cos\varphi$, $y = \rho\sin\varphi$. Для этого в ячейку **D2** ввести формулу: =B2*COS(A2), а в ячейку **E2** – формулу: =B2*SIN(A2). Вычислить и «растянуть» результаты аналогично п.1.

3. Выделить диапазон области определения и области значения функции (необходимые диапазоны столбцов **D** и **E,** то есть от **D2:E2** до **D150:E150**) и воспользоваться мастером построения диаграмм (**Вставка – Диаграмма**). Как и в предыдущем примере в **Мастере диаграмм** – **Диаграмма XY – Готово** без каких либо дополнительных установок (рис. 5).



	A	B	C	D	E
1					
2	0	0	2	0	0
3	0,05	0,298876265		0,298502747	0,014937587
4	0,1	0,591040413		0,588087673	0,059005584
5	0,15	0,869931068		0,86016268	0,130000874
6	0,2	1,129284947		1,106774433	0,224354285
7	0,25	1,36327752		1,320896523	0,337280256
8	0,3	1,566653819		1,496681559	0,46297786
9	0,35	1,734846451		1,629667417	0,594875044
10	0,4	1,864078172		1,716929694	0,725906232
11	0,45	1,951446716		1,757174541	0,848812063
12	0,5	1,994989973		1,750768412	0,956449142
13	0,55	1,993730057		1,699703764	1,042097239
14	0,6	1,947695262		1,607502267	1,09975147
15	0,65	1,85791943		1,479059557	1,124387582
16	0,7	1,726418733		1,32043788	1,112189484
17	0,75	1,556146394		1,138614995	1,060729698
18	0,8	1,350926361		0,94119946	0,969095253
19	0,85	1,115367435		0,736123708	0,837953698
20	0,9	0,85475976		0,531327188	0,669556322
21	0,95	0,574956025		0,334442197	0,467678145
22	1	0,282240016		0,152494932	0,237496784
23	1,05	-0,016814495		-0,008366406	-0,014585283
24	1,1	-0,315491388		-0,14310567	-0,281168247

Рисунок 5. - Результат выполнения задания построение «розы».

Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание.
4. Вывод по работе

Практическая работа № 6

Тема работы: Создание и редактирование изображений с помощью графического редактора Создание презентации

Цель работы: выработать практические навыки создания презентаций, настройки эффектов анимации, управления показом презентации при помощи гиперссылок.

Оборудование, приборы, аппаратура, материалы: персональный компьютер, программа LibreOffice Impress.

Краткие теоретические сведения.

Мультимедиа технологии - интерактивные (диалоговые) системы, обеспечивающие одновременную работу со звуком, анимированной компьютерной графикой, видеокадрами, изображениями и текстами.

Интерактивность – возможность диалога компьютера с пользователем на основе графического интерфейса с управляющими элементами (кнопки, текстовые окна и т.д.).

Компьютерная презентация является одним из типов мультимедийных проектов – последовательности слайдов (электронных карточек), содержащих мультимедийные объекты.

Применяется в рекламе, на конференциях и совещаниях, на уроках и т.д.

Переход между слайдами или на другие документы осуществляется с помощью кнопок или гиперссылок.

Создание презентаций осуществляется в программе LibreOffice Impress.

Impress - программа подготовки презентаций, входит в состав LibreOffice. Способна создавать PDF файлы из презентаций, а кроме того, экспортировать их в формат Adobe Flash (SWF), что даёт возможность просматривать их на любом компьютере с инсталлированным Flash-проигрывателем. Может показывать, редактировать и сохранять файлы в нескольких форматах, включая формат .ppt, который используется в Microsoft PowerPoint

Impress позволяет открывать файлы презентаций/шаблонов, созданных как с помощью самого LibreOffice Impress, так и с помощью Microsoft Power-Point (любой версии).

Что бы запустить Impress в операционной системе Windows, щелкните на кнопке **Пуск**, откройте меню **Программы** и выберите пункт **LibreOffice 4.3**. В появившемся на экране подменю щелкните на элементе пункт **LibreOffice Impress**.

После открытия программы **Impress** автоматически запускается мастер создания презентаций, который позволяет:

- ♦ создать пустую презентацию;
- ♦ создать презентацию на основе существующего шаблона;
- ♦ открыть уже существующую презентацию.

В появившемся окне мастера презентаций, на первом шаге (рис. 1) создания новой презентации, вы сможете оформить ее по своему усмотрению. Используя

шаблон, вы получите заготовку для презентации, в которой будет подобран фон, цвет, шрифт и т.д.

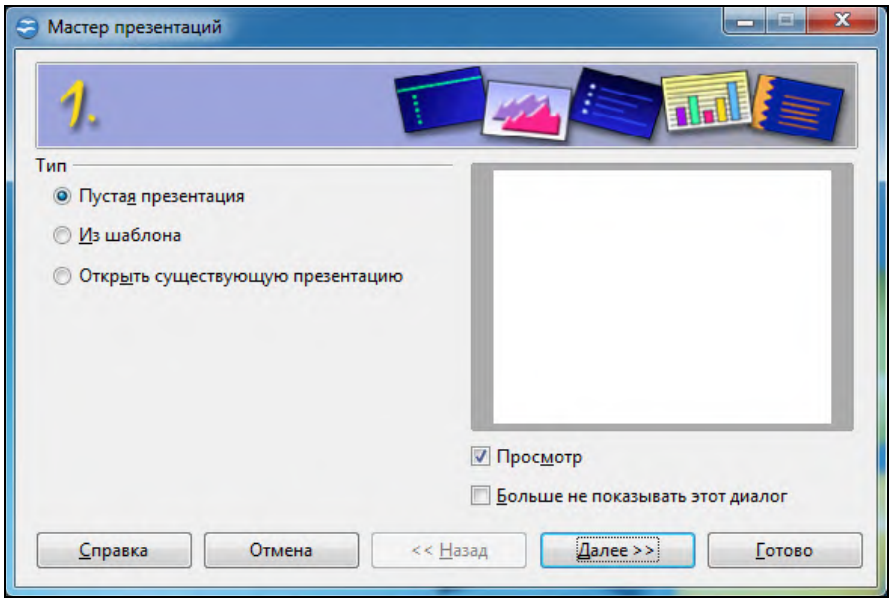


Рисунок 1 - Мастер презентаций. Шаг первый.

На втором шаге (рис. 2) создания презентации мастер предложит вам определиться со стилем слайда.

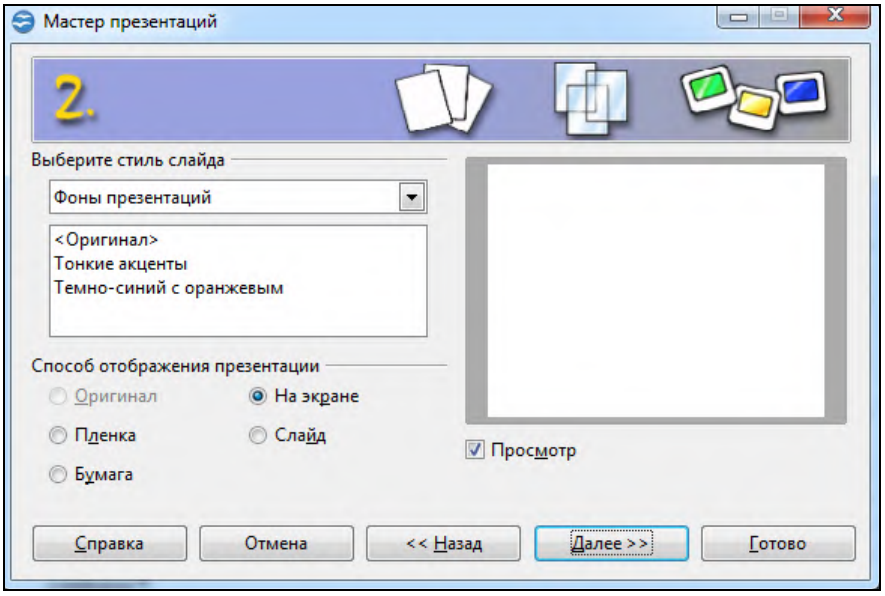


Рисунок 2 - Мастер презентаций. Шаг второй.

На третьем шаге (рис. 3) вы сможете определить, каким образом будет происходить смена слайдов в вашей презентации и определиться с типом показа презентации. То есть определиться, будет ли ваша презентация демонстрироваться автоматически согласно заданному времени (смена слайдов происходит друг за другом, после того, как вся презентация показана, пауза, а по истечению паузы показ презентации начинается снова).

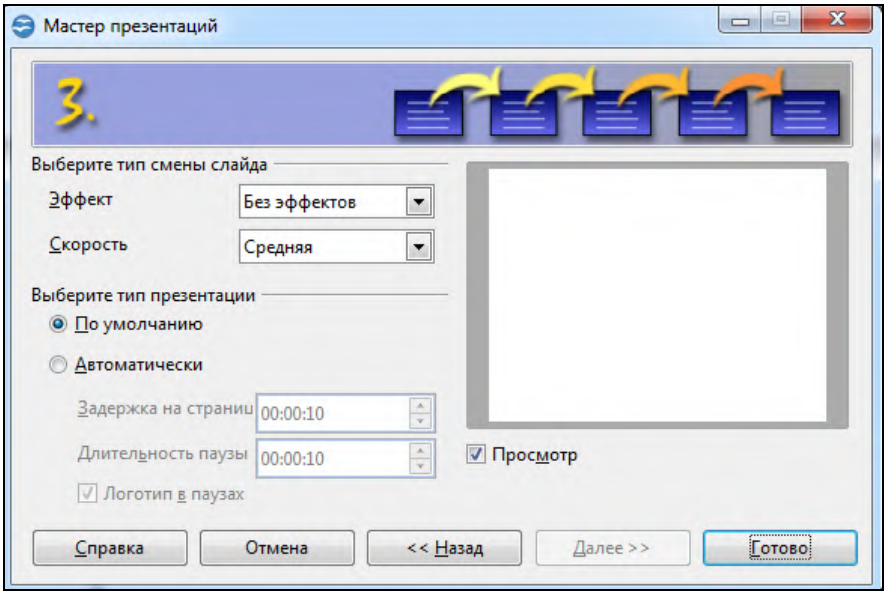


Рисунок 3 - Мастер презентаций. Шаг третий.

После того, как вы произведете все нужные настройки, необходимо нажать кнопку **Готово** — презентация будет создана.

В LibreOffice Impress есть несколько режимов просмотра презентаций — режим рисования, режим тезисов, режим структуры, режим примечаний, сортировщик слайдов. Переключаться между ними можно с помощью вкладок, которые расположены непосредственно над текущим слайдом.

Как правило, презентация создается в **Режиме рисования**, который имеет три области: панель **Слайды**, панель **Задачи** и непосредственно сам слайд.

Режим структуры отображает текст слайда в виде структуры, то есть, перейдя в этот режим можно увидеть название каждого слайда и его содержимое (тестовое). Данный режим удобен для создания содержания, для перемещения слайдов и текста.

Режим примечаний предназначен для создания тестовых примечаний к слайду. Область для заметок может перемещаться и изменяться.

Режим тезисов отображает один или несколько слайдов (в зависимости от выбранного макета на панели задач), при этом можно заполнить колонтитул, указать дату и т.п.

Режим Сортировщик слайдов позволяет увидеть эскизы созданных слайдов презентации. Здесь же имеется возможность добавлять, удалять, перемещать слайды.

Автоматически выводится по 4 слайда в строку, но это не всегда бывает удобно, поэтому в LibreOffice Impress можно указать количество слайдов, которое стоит выводить в одну строку (для этого нужно указать нужное количество в поле **Слайдов на строку**, которая расположена на панели инструментов **Сортировщик слайдов**).

Что бы добавить слайд в презентацию в **режиме Рисования** на панели **Слайды** выделите тот, после которого вы хотите добавить новый слайд, нажмите по нему правой кнопкой мыши, и в появившемся диалоговом меню выберите **Новый слайд**.

Перемещать слайды (менять их местами) можно в режимах **Рисования** и **Сортировщика слайдов**. При этом принцип перемещения слайдов в этих режимах одинаков

1. Выделяем слайд/слайды, которые мы будем перемещать.
2. Перетаскиваем их мышью в нужное место. При перемещении слайда/слайдов под указателем мышки будет отображаться пунктирный прямоугольник, который и говорит о том, что слайд/слайды перемещаются**. Большой жирный указатель** будет показывать то место, где будет располагаться наш слайд/слайды после перемещения.

На панели задач находятся пять вкладок:

- **Фоны страниц.** Здесь вы можете применить к вашей презентации шаблон, в котором подобран фон, цвет и шрифт.
- **Макеты.** Здесь представлены макеты слайдов.
- **Дизайн таблицы.** С помощью данной вкладки вы можете вставить в слайд таблицу и предать ей оригинальный дизайн.
- **Эффект.** С помощью данного меню вы можете снабдить оригинальной анимацией любой из элементов вашей презентации.

Смена слайда. В данном меню задается анимация смены слайдов, так же можно задать звуковое сопровождение.

Основные правила разработки и создания презентации

Правила шрифтового оформления:

- Шрифты с засечками читаются легче, чем гротески (шрифты без засечек);
- Для основного текста не рекомендуется использовать прописные буквы.
- Шрифтовой контраст можно создать посредством: размера шрифта, толщины шрифта, начертания, формы, направления и цвета.

Правила выбора цветовой гаммы:

- Цветовая гамма должна состоять не более чем из двух-трех цветов.
- Существуют не сочетаемые комбинации цветов.
- Черный цвет имеет негативный (мрачный) подтекст.
- Белый текст на черном фоне читается плохо (инверсия плохо читается).

Правила общей композиции:

- На полосе не должно быть больше семи значимых объектов, так как человек не в состоянии запомнить за один раз более семи пунктов чего-либо.
- Логотип на полосе должен располагаться справа внизу (слева наверху и т. д.).
- Логотип должен быть простой и лаконичной формы.
- Дизайн должен быть простым, а текст — коротким.

- Изображения домашних животных, детей, женщин и т.д. являются положительными образами.
- Крупные объекты в составе любой композиции смотрятся довольно неважно. Аршинные буквы в заголовках, кнопки навигации высотой в 40 пикселей, верстка в одну колонку шириной в 600 точек, разделитель одного цвета, растянутый на весь экран — все это придает дизайну непрофессиональный вид.

Единое стилевое оформление

- стиль может включать: определенный шрифт (гарнитура и цвет), цвет фона или фоновый рисунок, декоративный элемент небольшого размера и др.;
- не рекомендуется использовать в стилевом оформлении презентации более 3 цветов и более 3 типов шрифта;
- оформление слайда не должно отвлекать внимание слушателей от его содержательной части;
- все слайды презентации должны быть выдержаны в одном стиле;

Содержание и расположение информационных блоков на слайде

- информационных блоков не должно быть слишком много (3-6);
- рекомендуемый размер одного информационного блока — не более 1/2 размера слайда;
- желательно присутствие на странице блоков с разнотипной информацией (текст, графики, диаграммы, таблицы, рисунки), дополняющей друг друга;
- ключевые слова в информационном блоке необходимо выделить;
- информационные блоки лучше располагать горизонтально, связанные по смыслу блоки — слева направо;
- наиболее важную информацию следует поместить в центр слайда;
- логика предъявления информации на слайдах и в презентации должна соответствовать логике ее изложения.
- Помимо правильного расположения текстовых блоков, нужно не забывать и об их содержании — тексте. В нем ни в коем случае не должно содержаться орфографических ошибок. Также следует учитывать общие правила оформления текста.

Практическая часть

Задание 1. С помощью справочной системы выясните назначение пунктов меню панели инструментов LibreOffice Impress. Результаты представьте в таблице.

Задание 2. Создайте презентацию на тему, выбранную из списка, предложенного преподавателем, объемом минимум десять слайдов. Обязательным условием является использование различных эффектов смены,

анимации отдельных частей слайдов, использование текста, рисунков и диаграмм, а так же желательно применение звукового сопровождения.

Темы презентаций:

1. Измерение количества информации
2. Квест-технология
3. Виды носителей информации
4. Электронная почта
5. Архиваторы
6. История развития операционных систем
7. Что умеет компьютер?
8. Защита от несанкционированного доступа к информации
9. Локальные сети
10. Мультимедийные технологии
11. Принтеры
12. Виды информации
13. Графические способы представления данных в Excel
14. Зачем компьютеру программное обеспечение?
15. Создание презентаций в Microsoft Power Point
16. Базы данных
17. Основные понятия компьютерной графики
18. Архитектура ЭВМ и вычислительных систем
19. Этапы развития вычислительной техники
20. Вред и польза компьютера
21. Работа с объектами текстового документа
22. Антивирусы
23. Компьютерные мышки
24. География Интернета
25. Интернет - за и против.

Задание 3. Преобразуйте презентацию следующим образом:

- замените стандартный текст в слайдах шаблона вашим текстом;
- перейдя в режим Сортировщик слайдов, ознакомьтесь с вариантами:
- оформления слайдов;
- стандартных цветовых схем;
- эффектов смены слайдов и их звукового сопровождения;
- озвучьте первый слайд презентации с помощью звукового музыкального файла, второй — с помощью звукозаписи речевого комментария;
- ознакомьтесь с вариантами эффектов анимации текста и графических объектов слайдов;
- вставьте в последний слайд гиперссылку, позволяющую вернуться в начало презентации;
- сохраните презентацию в своей рабочей папке в двух форматах: презентации (ПР6.odp) и демонстрации (ПР6.odp);

- последовательно запустите на выполнение оба файла, отметьте различия операций запуска;
- ознакомьтесь с вариантами выделения отдельных элементов слайда в момент его демонстрации с помощью ручки, фломастера, маркера, расположенных в левом нижнем углу демонстрируемого слайда;
- установите автоматические режимы анимации объектов и смены слайдов презентации;
- запустите на выполнение слайд-фильм в режиме презентации и отрегулируйте временные интервалы показа слайдов, эффекты анимации и звука;
- запустите на выполнение слайд-фильм в режиме демонстрации.

Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

Контрольные вопросы

1. Что такое мультимедиа технологии? Их назначение.
2. Для чего нужны компьютерные презентации?
3. Перечислите основные правила разработки и создания презентаций:
   - правила шрифтового оформления;
   - правила выбора цветовой гаммы;
   - правила общей композиции;
   - правила расположения информационных блоков на слайде.

Практическая работа № 7-9

Тема работы: Создание и заполнение таблиц. Установка связей. Создание запросов. Создание форм и отчетов

Цель работы: 1) изучении е технологии создания базы данных с помощью СУБД LibreOffice Base;

2) развивать навыки структуризации данных;

3) воспитывать ответственное отношение к информации.

Оборудование, приборы, аппаратура, материалы: персональный компьютер, программа LibreOffice Base.

Краткие теоретические сведения.

Развитие информационных технологий привело к созданию компьютерных баз данных. Создание баз данных, а также операции поиска и сортировки данных выполняются специальными программами — *системами управления базами данных (СУБД).*

Система управления базами данных (СУБД) - это программа, позволяющая создавать базы данных, а также обеспечивающая обработку (сортировку) и поиск данных.

СУБД обеспечивает программные средства для создания, загрузки, запроса и обновления данных, контролирует действия, связанные с вводом-выводом данных, решает вопросы совместного их использования и защиты.

СУБД служит для поддержания базы данных в актуальном состоянии и обеспечивает эффективный доступ пользователей к содержащимся в ней данным в рамках предоставленных пользователям полномочий.

По степени универсальности различают два класса СУБД:

1) системы общего назначения;

2) специализированные системы.

СУБД общего назначения не ориентированы на какую-либо предметную область или на информационные потребности какой-либо группы пользователей. Каждая система такого рода реализуется как программный продукт, способный функционировать на некоторой модели ЭВМ в определенной операционной системе, и поставляется многим пользователям как коммерческое изделие.

Такие СУБД обладают средствами настройки на работу с конкретной базой данных.

Использование СУБД общего назначения в качестве инструментального средства для создания автоматизированных информационных систем, основанных на технологии баз данных, позволяет существенно сокращать сроки разработки, экономить трудовые ресурсы. Этим СУБД присущи развитые функциональные возможности и даже определенная функциональная избыточность.

Специализированные СУБД создаются в редких случаях при невозможности или нецелесообразности использования СУБД общего назначения.

СУБД общего назначения - это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.

- Они позволяют определять структуру создаваемой базы, инициализировать ее и производить начальную загрузку данных.

- Они управляют полномочиями пользователей на доступ к БД, организуют параллельный доступ к ней нескольких пользователей.

- Они обеспечивают защиту логической и физической целостности данных - защиту от разрушений.

СУБД поддерживают один из возможных типов моделей данных - сетевую, иерархическую или реляционную, которые являются одним из важнейших признаков классификации СУБД.

СУБД обеспечивают многоцелевой характер использования базы данных, защиту и восстановление данных. Наличие развитых диалоговых средств и языка запросов высокого уровня делает СУБД удобным средством для конечного пользователя.

Единовременно может быть открыта только одна база данных, содержащая обязательное *окно базы данных* и *окна для работы с объектами базы данных.* В каждый момент времени одно из окон является активным и в нем курсором отмечается активный объект.

Окно базы данных — один из главных элементов интерфейса Access. Здесь систематизированы все объекты БД: *таблицы, запросы, формы, отчеты, макросы и модули.*

Таблица. В базах данных вся информация хранится в двумерных таблицах. Это *базовый* объект БД, все остальные объекты создаются на основе существующих таблиц *(производные* объекты). Каждая строка в таблице — *запись* БД, а столбец — *поле.*

Запись содержит набор данных об одном объекте, а поле — однородные данные обо всех объектах.

Запросы. В СУБД запросы являются важнейшим инструментом. Главное предназначение запросов — отбор данных на основании заданных условий. С помощью запроса из базы данных можно выбрать информацию, удовлетворяющую определенным условиям.

Формы. Формы позволяют отображать данные, содержащиеся в таблицах или запросах, в более удобном для восприятия виде. При помощи форм можно добавлять в таблицы новые данные, а также редактировать или удалять существующие. Форма может содержать рисунки, графики и другие внедренные объекты.

Отчеты. Отчеты предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде.

4. Задания.

<u>Задание 1. Создание базовых таблиц.</u>

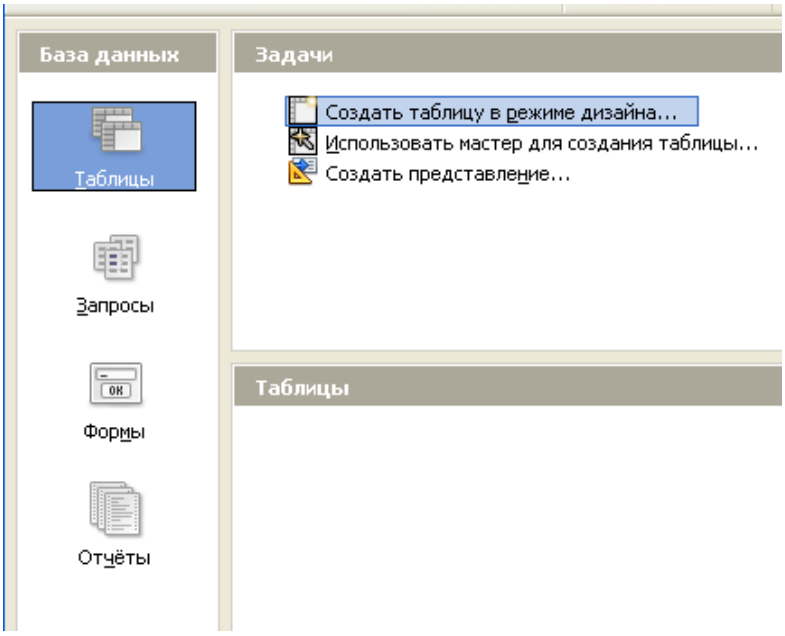
Руководитель торгового предприятия, реализующего научную литературу, заказал разработку базы данных, основанной на двух таблицах. Одна таблица содержит данные, которые могут отображаться для клиентов при оформлении

закупки, - в ней указаны розничные цены на книги. Вторая таблица предназначена для анализа результатов деятельности предприятия – в ней содержатся закупочные оптовые цены и краткая информация о поставщиках (покупатели не имеют доступа к данным этой таблицы).

1. Запустите программу **LibreOffice Base** (***Пуск→ Программы→*** **LibreOffice 3.6→ LibreOffice Base**).

2. Нажмите ***Готово***.

3. В окне ***Сохранить как*** выберите папку ***Мои документы (на Рабочем столе не сохранять, т.к. после перезагрузки ПК с него всё удаляется)*** и дайте файлу имя: ***Книготорговля***. Убедитесь, что в качестве типа файла выбрано ***Базы данных ODF***, и щелкните на кнопке ***Создать***. Откроется окно новой базы – ***Книготорговля***.



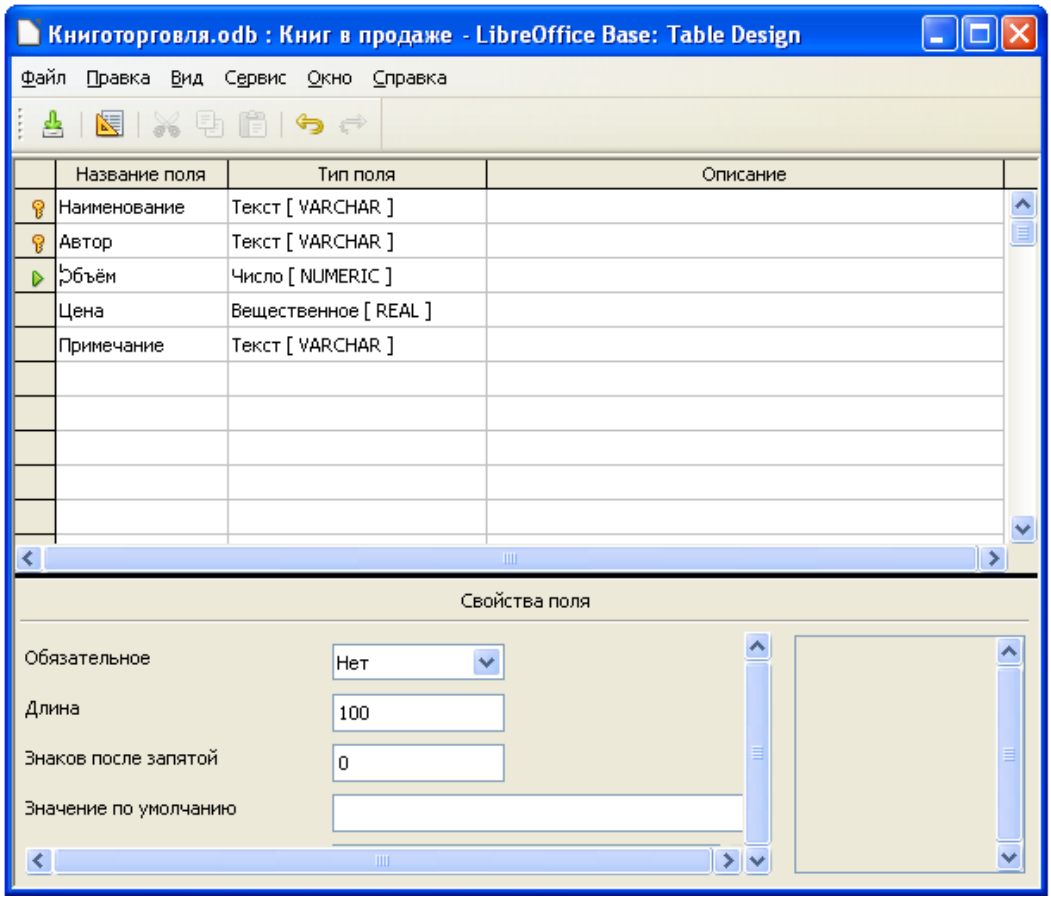
4. Откройте панель ***Таблицы***.

5. Дважды щелкните на значке ***Создать таблицу в режиме дизайна*** – откроется бланк создания структуры таблицы.

6. Для первой таблицы введите следующие поля:

Название поля	**Тип поля**
Наименование	Текст[VARCHAR]
Автор	Текст[VARCHAR]
Объем	Число[NUMERIC]
Цена	Вещественное[REAL]
Примечание	Текст[VARCHAR]

7. Щелкните на поле ***Объем***. В нижней части бланка задайте свойство *Знаков после запятой* равным ***0***.



8. Для связи с будущей таблицей поставщиков надо задать *ключевое поле*. На уникальность может претендовать поле ***Наименование***, но в больших базах данных возможно появление разных книг с одинаковым названием. Поэтому мы используем комбинацию полей ***Наименование*** и ***Автор***. Выделите оба поля в верхней части бланка (при нажатии клавиши **SHIFT**). Щелчком правой кнопки мыши откройте контекстное меню и выберите в нем пункт ***Первичный ключ***.

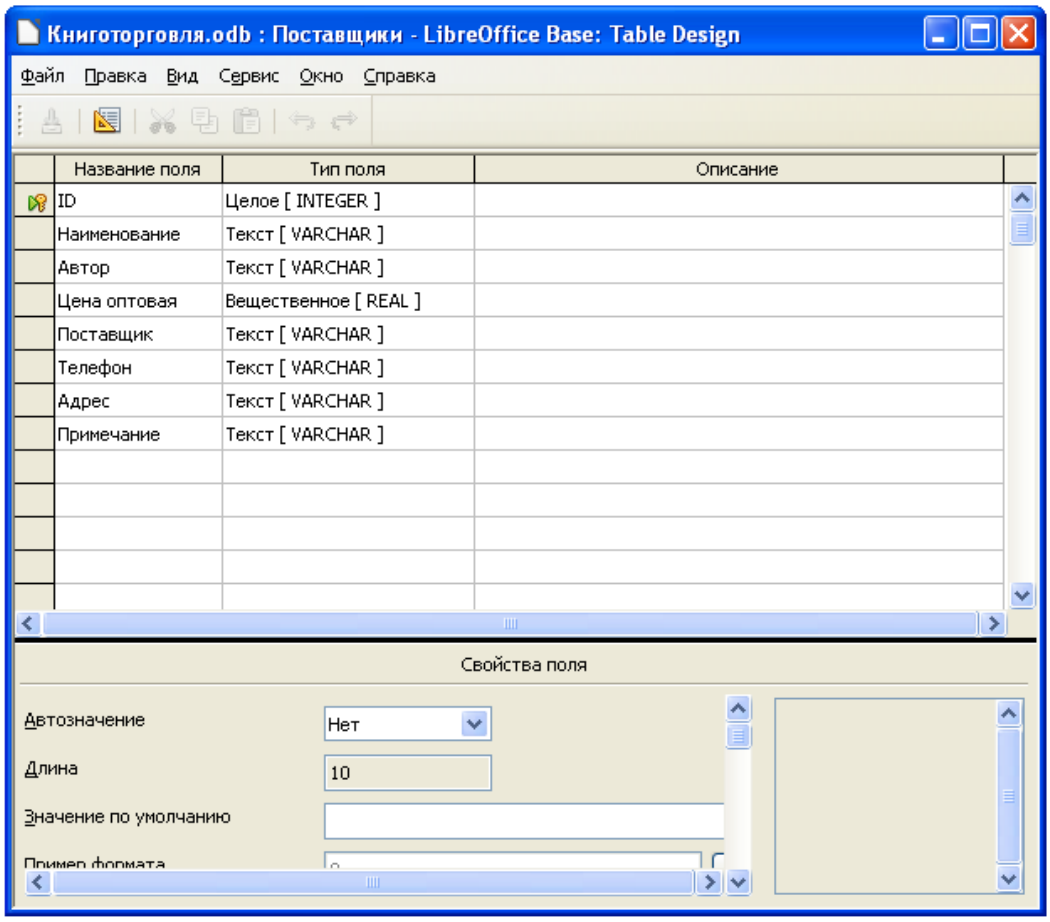
9. Закройте окно ***Конструктора***. При закрытии окна дайте таблице имя ***Книги в продаже***.

10. Повторив действия пунктов 5-9, создайте таблицу ***Поставщики***, в которую входят следующие поля:

Имя поля	**Тип поля**
Наименование	Текст[VARCHAR]
Автор	Текст[VARCHAR]
Цена оптовая	Вещественное[REAL]
Поставщик	Текст[VARCHAR]
Телефон	Текст[VARCHAR]
Адрес	Текст[VARCHAR]
Примечание	Текст[VARCHAR]

Обратите внимание на то, что поле номера телефона является текстовым, не смотря на то, что обычно номера телефонов записывают цифрами. Это связано с тем, что они не имеют числового содержания. Номера телефонов не сравнивают

по величине, не вычитают из одного номера другой и т.д. Это типичное текстовое поле. При сохранении согласиться на создание ключевого поля.



Данные для заполнения таблиц:

Таблица Книги в продаже:

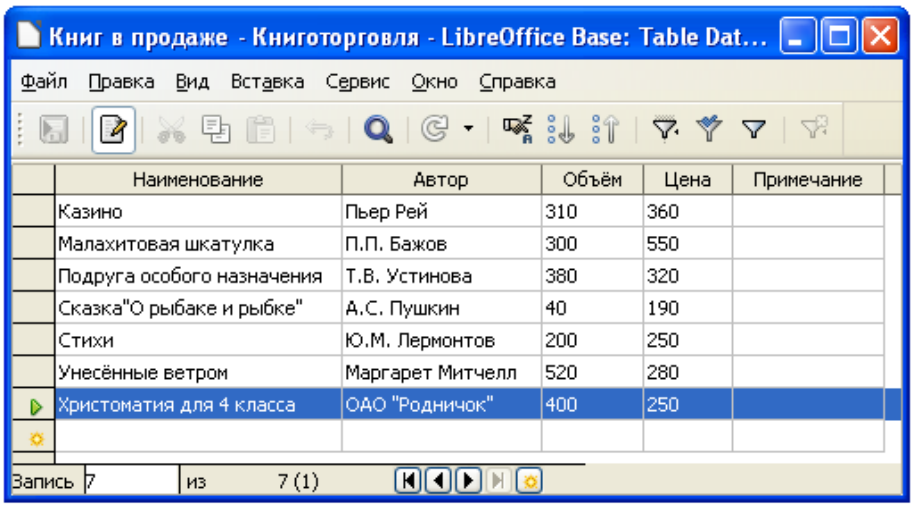
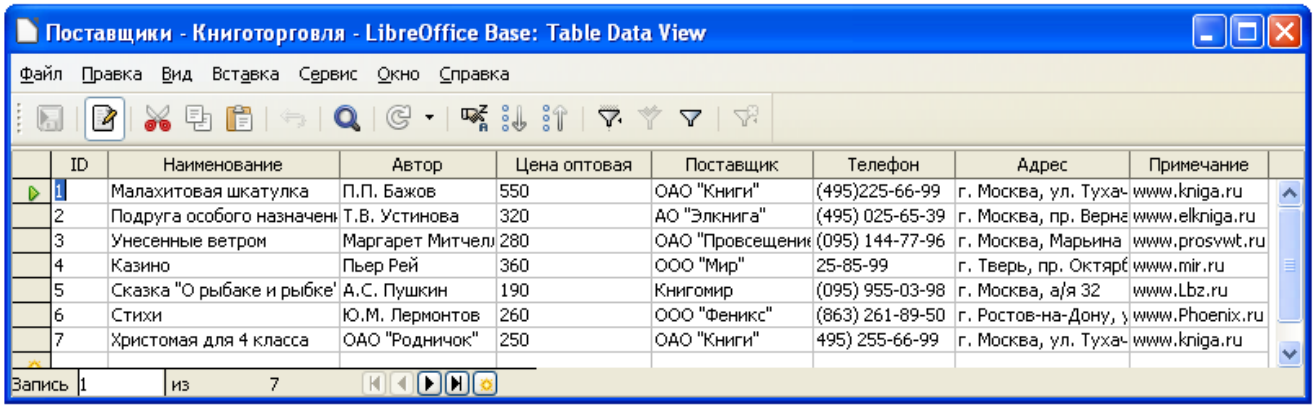


Таблица Поставщики

Наименование	Автор	Цена оптовая	Поставщик	Телефон	Адрес	Примечание
Малахитовая шкатулка	П.П. Бажов	550,00 р.	ОАО "Книги"	(495) 255-66-99	г. Москва, ул. Тухачевского, 34	www.kniga.ru
Подруга особого назначения	Т.В. Устинова	320,00 р.	АО "Элкнига"	(495) 025-65-39	г. Москва, пр. Вернатского, 356	www.elkniga.ru
Унесенные ветром	Маргарет Митчелл	280,00 р.	ОАО "Провсещение"	(095) 144-77-96	г. Москва, Марьина роща, 41	www.prosvwt.ru
Казино	Пьер Рей	360,00 р.	ООО "Мир"	25-85-99	г. Тверь, пр. Октярбря, 46	www.mir.ru
Сказка "О рыбаке и рыбке"	А.С. Пушкин	190,00 р.	Книгомир	(095) 955-03-98	г. Москва, а/я 32	www.Lbz.ru
Стихи	Ю.М. Лермонтов	260,00 р.	ООО "Феникс"	(863) 261-89-50	г. Ростов-на-Дону, ул. Советская, 57	www.Phoenix.ru
Христомая для 4 класса	ОАО "Родничок"	250,00 р.	ОАО "Книги"	495) 255-66-99	г. Москва, ул. Тухачевского, 34	www.kniga.ru

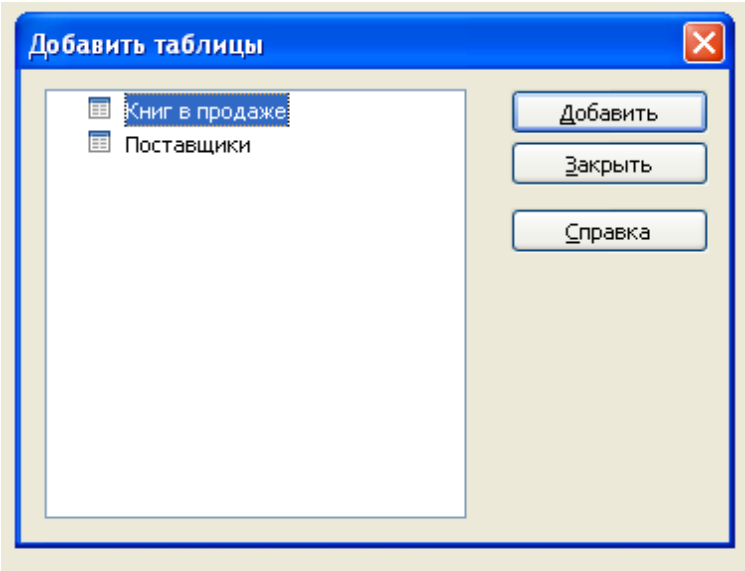
1. В окне Книготорговля: база данных откройте по очереди созданные таблицы и наполните их экспериментальным содержанием (3-10 записей). Закончив работу, закройте таблицы.



<u>Задание 2. Создание межтабличных связей.</u>

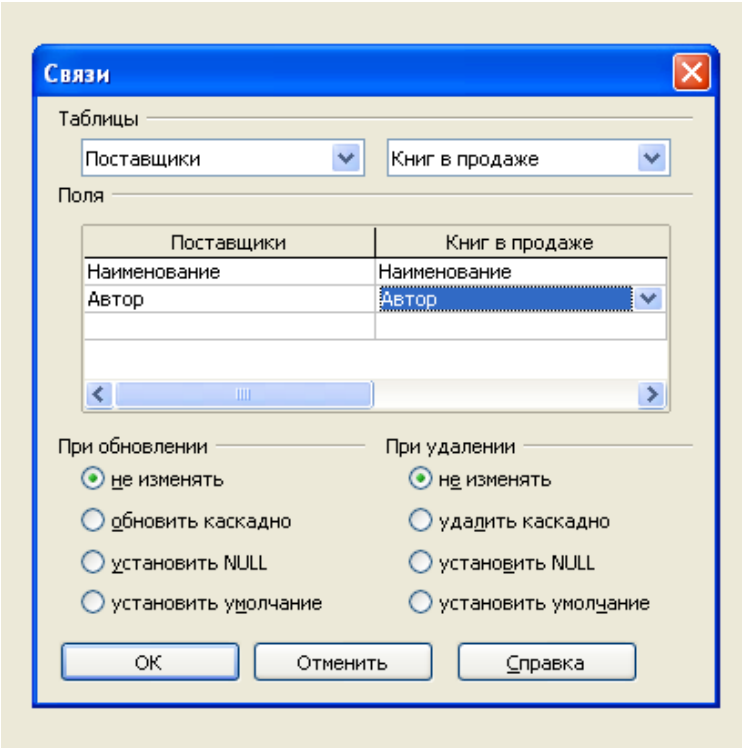
1. В окне Книготорговля: база данных откройте панель ***Таблицы***. Убедитесь, что на ней присутствуют значки ранее созданных таблиц ***Книги в продаже*** и ***Поставщики***.

2. Найдите команду строки меню: ***Сервис→Связи...***. Одновременно с открытием окна ***Схема данных*** открывается диалоговое окно ***Добавление таблицы***.

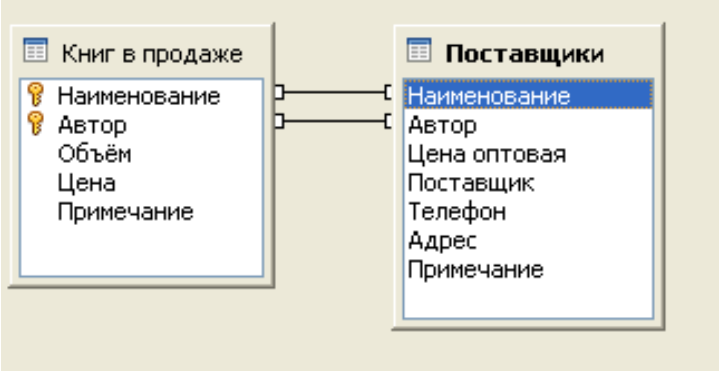


3. Щелчком на кнопке **Добавить** выберите таблицы *Книги в продаже* и *Поставщики* – в окне схема данных откроются списки полей этих таблиц.

4. Разыщите на панели инструментов кнопку **Создать связь**.
5. Откроется окно ***Связи,*** приведите его к следующему виду:



7. Закройте диалоговое окно ***Связи*** и в окне ***Схема данных*** рассмотрите образовавшуюся связь. Убедитесь в том, что линию связи можно выделить щелчком левой мыши, а щелчком правой кнопки мыши открывается контекстное меню, позволяющее разорвать связь или отредактировать ее.

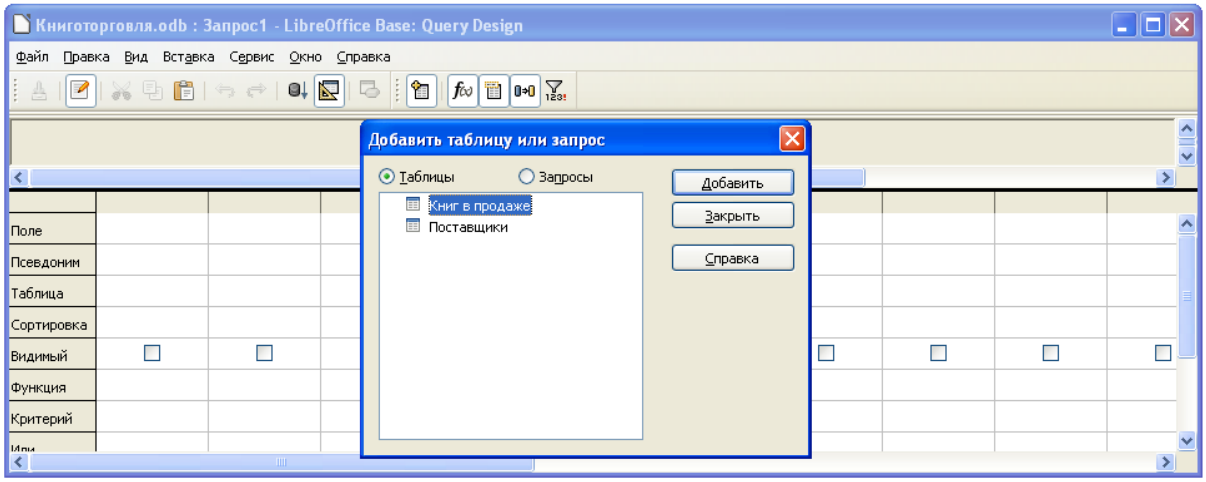


8. Закройте окно ***Схема данных***. Сохраните изменения.

<u>Задание 3. Создание запроса на выработку.</u>

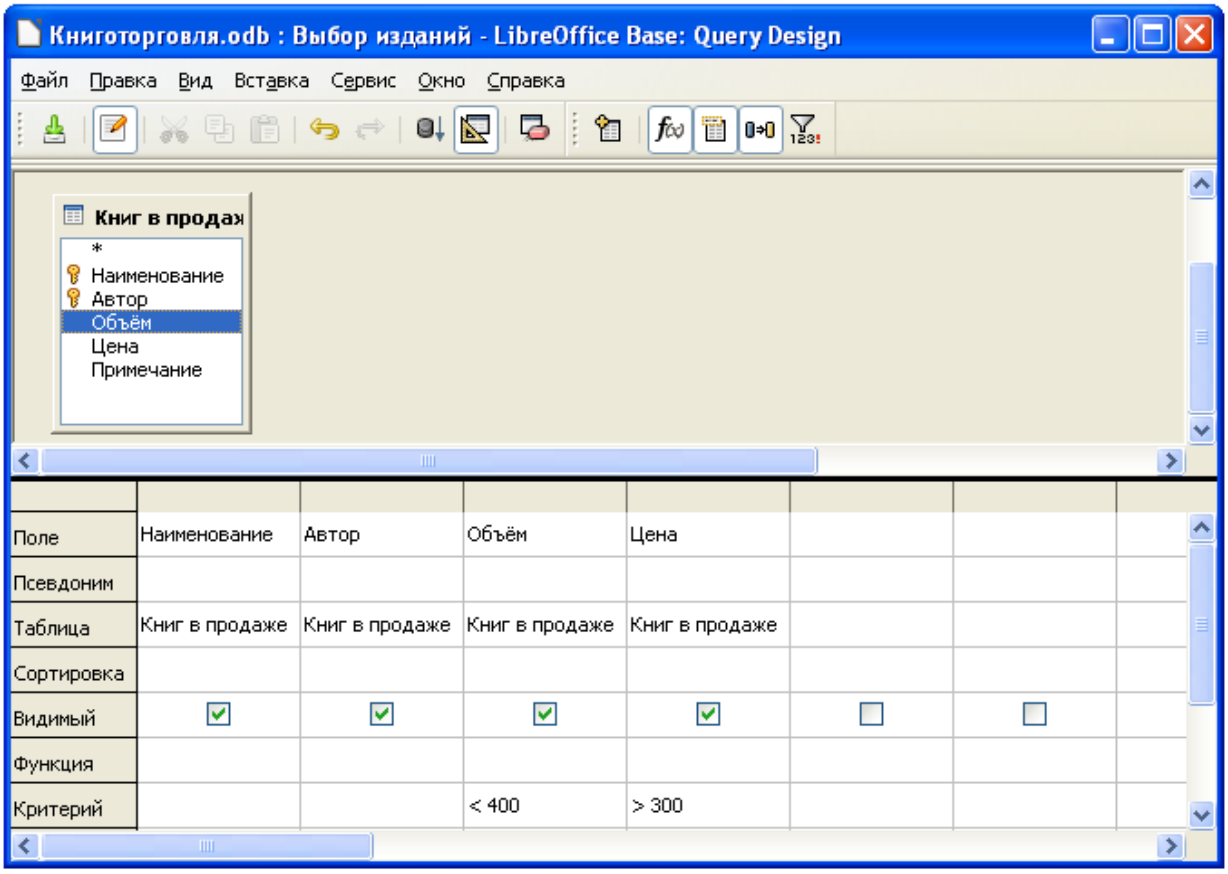
В этом упражнении мы создадим запрос на выборку книг, имеющих объем не менее 400 страниц при цене менее 80 рублей.

1. В окне Книготорговля: база данных откройте панель ***Запросы***. Дважды щелкните на значке ***Создать запрос в режиме дизайна…*** – откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно ***Добавить таблицу или запрос***.



2. В окне ***Добавить таблицу или запрос*** выберите таблицу *Книги в продаже* и щелкните на кнопке ***Добавить***. Закройте окно добавление таблицы.

3. В списке полей таблицы ***Книги в продаже*** выберите поля, включаемые в результирующую таблицу: ***Наименование***, ***Автор***, ***Объем***, ***Цена***. Выбор производите двойными щелчками на именах полей.



4. Задайте критерий для поля ***Объём***. В соответствующую строку введите: **<400**. Из таблицы будут выбираться не все издания, а только те, объем которых не превышает 400 страниц.

5. Задайте критерий для поля ***Цена***. В соответствующую строку введите: **>300**. Теперь из таблицы будут выбираться только изделия, имеющие цену более 300 рублей.

6. Закройте бланк запроса по образцу. При закрытии запроса введите его имя – ***Выбор изданий***.

7. В окне Книготорговля: база данных откройте только что созданный запрос и рассмотрите результирующую таблицу. Её содержательность зависит от того, что было введено в таблицу Книги в продаже при ее наполнении в задании 1. Если ни одно издание не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.

8. По окончании исследований закройте все открытые объекты и сохраните изменения.

<u>Задание 4. Создание запросов с «параметром».</u>

Выше мы рассмотрели, как действует условие отбора, но должны отметить его существенный недостаток. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий издания, имеющие цену менее 80 рублей, то пользователь базы уже не в состоянии отобрать книги, цена которых менее 150 рублей, поскольку у него нет соответствующего запроса.

Специальный тип запросов, называемый запросами «с параметром», позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этим приемом обеспечивается гибкость работы с базой.

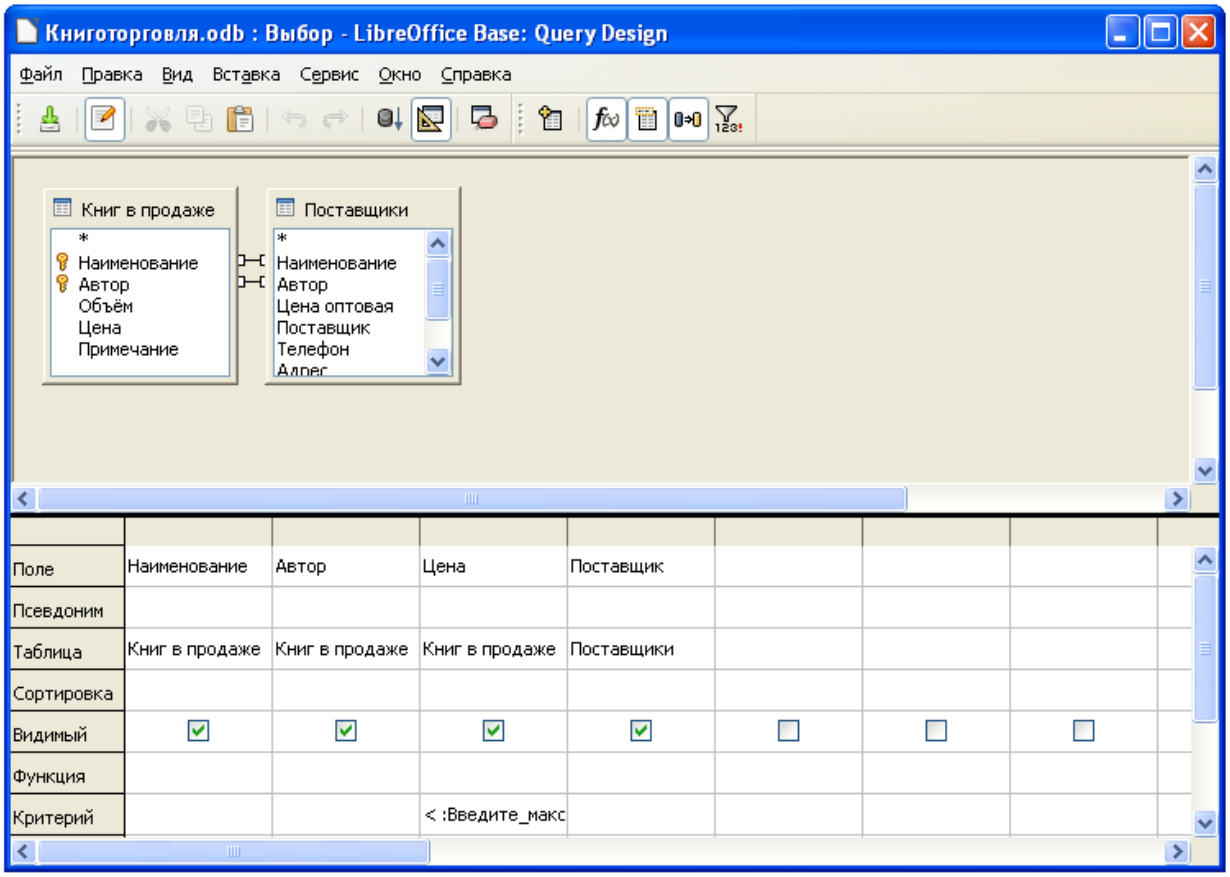
Создадим простой запрос, позволяющий отбирать издания, предельную цену которых пользователь может задать сам при запуске запроса.

1. В окне Книготорговля: база данных откройте панель **Запросы**. Дважды щелкните на значке **Создать запрос в режиме дизайна…** – откроется бланк запроса по образцу.

2. Согласно заданию 3, создайте запрос на выборку, основанный на таблице Книги в продаже и Поставщики, в который войдут следующие поля:

- Наименование;
- Автор;
- Цена;
- Поставщик.

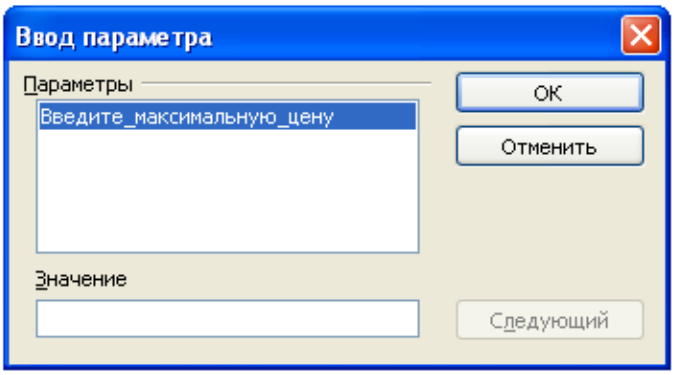
3. Строку **Критерий** для поля ***Цена*** надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение. Введите следующее выражение: **< :Введите_максимальную_цену**.



4. Введите запрос. При закрытии сохраните его под именем ***Выбор книг***.

5. В окне ***Книготорговля***: ***база данных*** откройте панель ***Запросы*** и запустите запрос ***Выбор книг*** – на экране появится диалоговое окно ***Ввод параметра***.

6. Введите какое-либо число и щелкните на кнопке **ОК**. В зависимости от того, что реально содержится в таблице ***Книги в продаже***, по результатам запроса будет сформирована результирующая таблица.

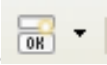


7. Закройте все объекты базы данных. Сохраните изменения.

<u>Задание 5. Создание простой формы.</u>

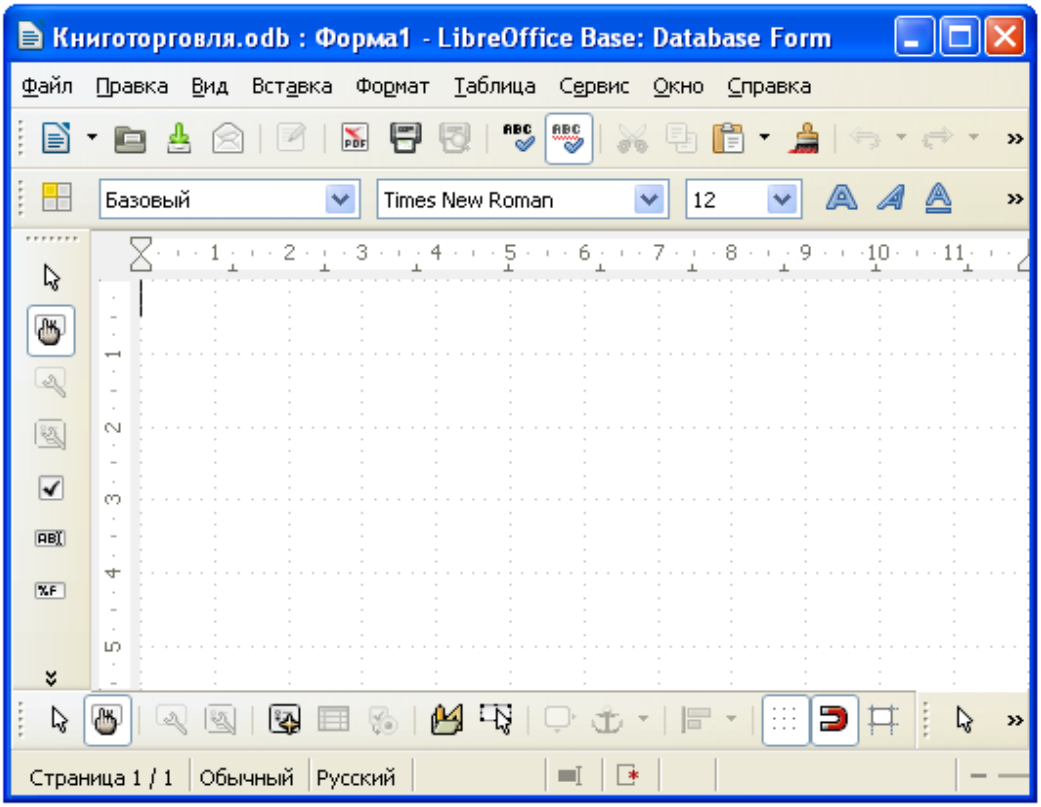
1.В диалоговом окне базы данных выберите вкладку ***Таблицы***, если находитесь в другой вкладке.

2.Выделите таблицу «***Книги в продаже***». (форму можно строить и при закрытой таблице, и при открытой.)

3.Щелкните по кнопке раскрытия списка, расположенной на панели инструментов  , и выберите элемент ***Форма***.

4.Сохраните форму под именем «***Продажа книг***».

Закройте форму, ответив утвердительно на вопрос о сохранении, если таковой последует.



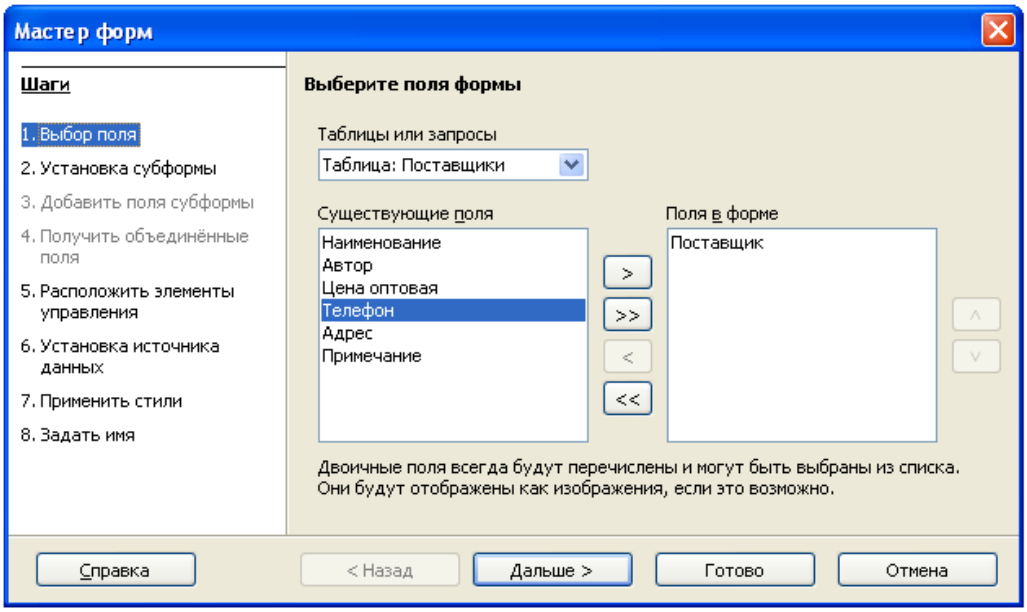
<u>Задание 7. Создание простой формы.</u>

1. В окне базы данных выберите вкладку ***Формы***.

2.Щелкните по кнопке ***Использовать мастер для создания формы*** .

3.В поле ***Таблицы/запросы*** выберите таблицу «Книги в продаже», в поле ***Существующие поля*** выберите поля «Категория», «Наименование», «Автор» и «Цена» перенесите их стрелкой в поле ***Поля в форме***.

4. В поле ***Таблицы/запросы*** выберите таблицу «Поставщики», в поле ***Существующие поля*** выберите поле «Поставщик» и перенесите его стрелкой в поле ***Поля в форме***. Щелкните по кнопке ***Применить стили***.



5. Выберите требуемый стиль (например, ***Вода***). Щелкните по кнопке ***Далее***.

6. Задайте имя формы: «***Поставщики***». Щелкните по кнопке ***Готово***.

В результате получите форму, в которой можно менять существующие данные и вводить новые значения сразу в две таблицы.

7. Закройте форму. Сохраните изменения.

Задание 8. Создание отчета.

1.Откройте вкладку ***Отчеты***, если находитесь в другом окне.

2.Щелкните по кнопке ***Создать отчёт в режиме дизайна***.

3.В появившемся диалоговом окне ***Добавить поле: Книг в продаже*** выберите все поля таблицы.

4.Закройте диалоговое окно.

5. Сохраните отчет под именем «Книги». Закройте отчет.

Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание.
4. Вывод по работе

Контрольные вопросы

1. Что такое база данных?
2. Чем отличается база данных от СУБД.
3. Основные объекты СУБД.
4. Два режима создания таблиц.
5. Назначение запросов.
6. Назначение форм.
7. Назначение отчетов.

Практическая работа № 10

Тема работы: Решение прикладных математических задач.

Цель работы: научиться решать прикладные математические задаче

Оборудование, приборы, аппаратура, материалы: персональный компьютер, программа LibreOffice Calk.

I. Теоретическая часть

В задачах линейного программирования всегда необходимо найти минимум (или максимум) линейной функции многих переменных при **линейных** ограничениях в виде равенств или неравенств.

$$L = c_0 + c_1x_1 + c_2x_2 + \cdots c_nx_n \to \min$$
$$a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n \le b_1$$
$$\vdots$$
$$a_{k1}x_1 + a_{k2}x_2 + \cdots a_{kn}x_n \le b_k$$
$$x_1, x_2, \cdots, x_n \ge 0$$

В задачи целочисленного программирования добавляется ограничение, что все $\mathbf{x_i}$ должны быть **целыми**.

II. Практическая часть

Задание 1. Решить задачу линейного программирования:

$$L = 5x_1 - 2x_3 \to \min$$
$$-5x_1 - x_2 + 2x_3 \le 2$$
$$-x_1 + x_3 + x_4 \le 5$$
$$-3x_1 + 5x_4 \le 7$$

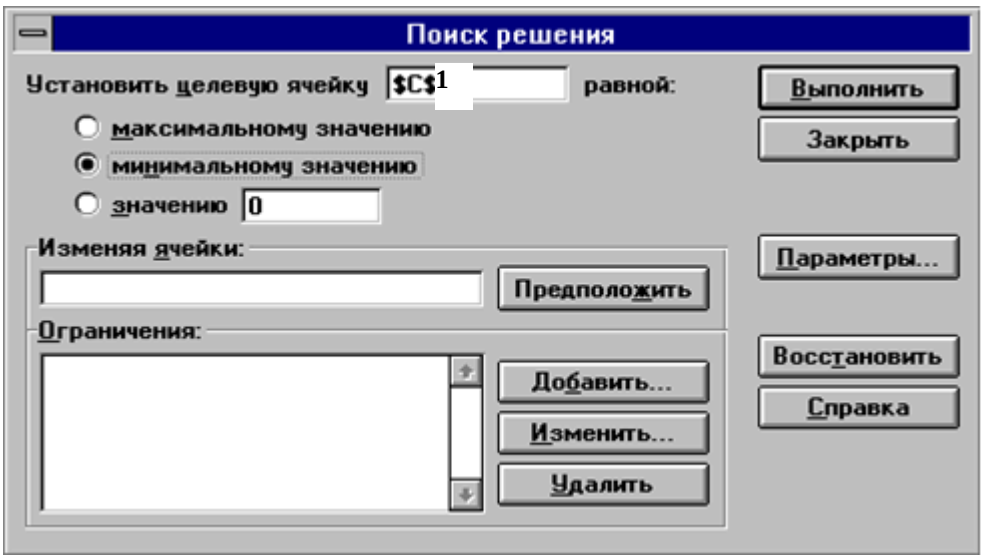
Для решения подобных задач предназначена команда **Поиск решения** из меню **Сервис**.

Пусть значения $\mathbf{x_1}$, $\mathbf{x_2}$, $\mathbf{x_3}$, $\mathbf{x_4}$ хранятся в ячейки **A1:A4**, а значение функции **L** - в ячейке **C1**. Введем ограничения:

C2 = -5*A1 - A2 + 2*A3
C3 = -A1 +A3 + A4
C4 = -3*A1 + 5*A4.

Таким образом, было задано условие исходной задачи линейного программирования.

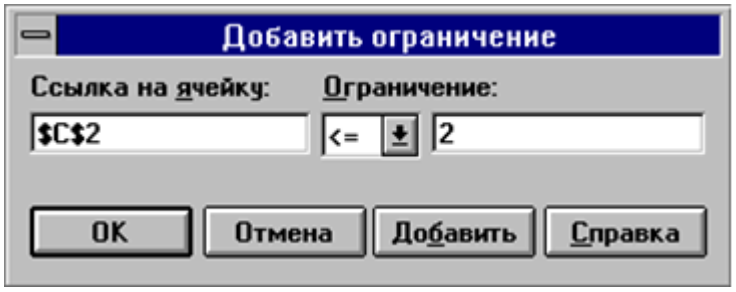
Выполним команду из главного меню **Сервис → Поиск решения** .



Устремим целевую функцию в ячейке **C1** к минимуму. Для этого введем в поле **Установить целевую функцию** значение **C1** и установим опцию **"равной минимальному значению"**.

В поле **Изменяя ячейки** необходимо указать адреса ячеек, в которых хранятся изменяемые значения. В нашем случае это ячейки **A1:A4**.

Для добавления ограничений необходимо щелкнуть по кнопке **Добавить**, появится диалоговое окно **Добавить ограничение** .

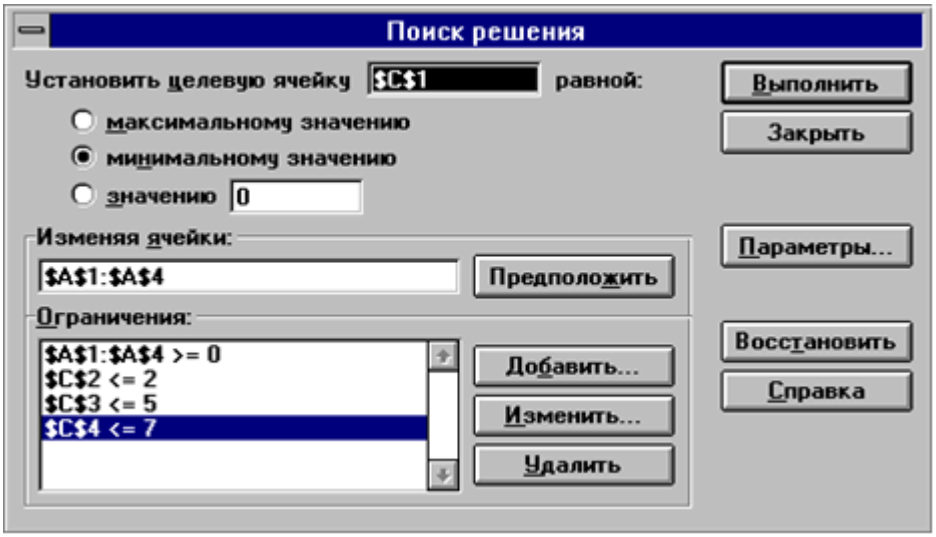


В поле ввода **Ссылка на ячейку** необходимо ввести адрес ячейки, где хранится ограничение, затем, щелкнув по стрелке, выбрать знак и ввести значение ограничения в поле **Ограничение**.

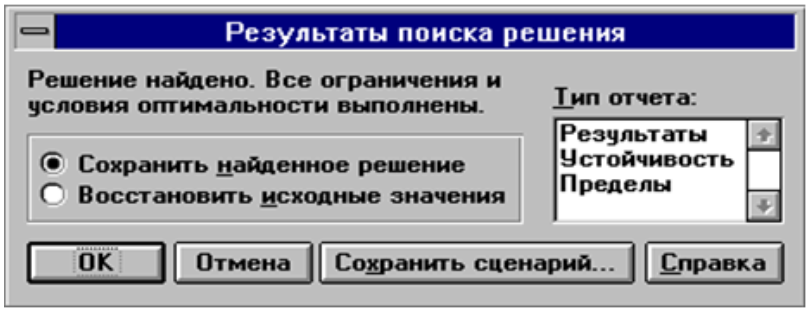
Щелчок по кнопке **OK** означает ввод очередного ограничения и возврат к диалоговому окну **Поиск решения**.

Щелчок по кнопке **Добавить** вводить очередное ограничение, находясь в окне **Добавить** ограничение.

В нашем случае окно будет иметь вид, изображенный на рисунке



Щелчок по кнопке **Выполнить** начнет процесс решения задачи, завершится который появлением диалогового окна, изображенного на рисунке.



Щелчок по кнопке **OK** приведет к появлению в ячейке **C1** значения целевой функции **L**, а в ячейках **A1:A4** - значений переменных x_1-x_4, при которых целевая функция достигает минимального значения.

Если задача не имеет решения или неверно были заданы исходные данные, в окне **Результаты поиска решения** может появиться сообщение о том, что решение не найдено.

Итак, назначение основных кнопок и окон диалогового окна **Поиск решения**:

- Поле **Установить целевую ячейку** - определяет целевую ячейку, значение которой необходимо максимизировать или минимизировать, или сделать равным конкретному значению.
- Опции **"минимальному значению"**, **"максимальному значению"** и **"значению"**, определяют, что необходимо сделать со значением целевой ячейки - максимизировать, минимизировать или сделать равным конкретному значению.
- Поле **Изменяя ячейки** определяет изменяемые ячейки. Изменяемая ячейка - это ячейка, которая может быть изменена в процессе поиска решения для

достижения нужного результата в ячейке из окна **Установить целевую ячейку** с удовлетворением поставленных ограничений.

- Кнопка **Предположить** отыскивает все неформульные ячейки, прямо или непрямо зависящие от формулы в окне **Установить целевую ячейку**, и помещает их ссылки в окно **Изменяя ячейки**.
- Окно **Ограничения** перечисляет текущие ограничения в данной задаче. Ограничение есть условие, которое должно удовлетворяться решением; ограничения перечисляются в виде ячеек или интервалов ячеек, обычно содержащих формулу, которая зависит от одной или нескольких изменяемых ячеек, чье значение должно попадать внутрь определенных границ или удовлетворять равенству.
- кнопки **Добавить, Изменить, Удалить** позволяют добавить, изменить или удалить ограничение.
- Кнопка **Выполнить** запускает процесс решения определенной задачи.
- Кнопка **Закрыть** закрывает окно диалога, не решая проблемы. Сохраняются лишь изменения, сделанные при помощи кнопок **Параметры, Добавить, Изменить** и **Удалить**. Не сохраняются изменения, произведенные после использования данных кнопок.
- Кнопка **Параметры** выводит окно диалога **Параметры поиска решения**, в котором можно контролировать различные аспекты процесса отыскания решения, а также загрузить или сохранить некоторые параметры, такие, как выделение ячеек и ограничений, для какой-то конкретной задачи на рабочем листе.
- Кнопка **Сбросить** очищает все текущие установки задачи и возвращает все параметры к их значениям по умолчанию.

С помощью решающего блока можно решить множество различный оптимизационных задач (задач на максимум и минимум) с ограничениями любого типа. При решении задачи целочисленного программирования необходимо добавить ограничение, показывающее, что переменные целочисленные. При решении других оптимизационных задач вводят целевую функцию и ограничения.

Задание для самостоятельной работы:

1 $$W = 2x_1 - x_2 + x_4 \to \min$$
$$\begin{cases} x_1 + x_2 + x_3 - x_4 \le 1 \\ x_1 - x_2 + x_3 - x_4 \le 0 \\ 2x_1 + x_2 + x_3 - x_4 \ge 3 \end{cases}$$

2 $$W = 3 + 2x_2 + x_3 \to \max$$
$$\begin{cases} x_1 - x_2 + 2x_3 + x_4 \ge 1 \\ 2x_1 - x_2 + x_3 - x_4 \ge 1 \\ x_1 - 2x_2 + x_3 - x_4 \ge -1 \\ x_1 + x_2 + x_3 + 2x_4 \le 5 \end{cases}$$

3

$$W = x_3 + 3x_4 \to \min$$
$$\begin{cases} x_1 + x_2 - x_3 - x_4 \le 2 \\ x_1 - x_2 - x_3 + x_4 \ge 0 \\ -x_1 - x_2 + 2x_3 - x_4 \ge -3 \\ x_1 \ge 1 \end{cases}$$

4

$$W = x_1 - x_2 - 2x_4 \to \max$$
$$\begin{cases} 2x_1 - x_2 + 2x_3 - x_4 \le 4 \\ x_1 - 2x_2 + x_3 - 2x_4 \ge 2 \\ x_1 - x_4 \ge 1 \\ x_2 + x_3 \le 1 \end{cases}$$

5

$$W = -x_2 - 2x_3 + x_4 \to \min$$
$$\begin{cases} 3x_1 - x_2 \le 2 \\ x_2 - 2x_3 \le -1 \\ 4x_3 - x_4 \le 3 \\ 5x_1 + x_4 \ge 6 \end{cases}$$

6

$$W = -4 - 2x_1 - x_2 - x_3 \to \min$$
$$\begin{cases} x_1 - 2x_2 + 3x_3 - 4x_4 \ge -10 \\ x_1 + x_2 - x_3 - x_4 \le -4 \\ x_1 - x_2 + x_3 - x_4 \ge -6 \\ x_1 + x_2 + x_3 + x_4 \le 10 \end{cases}$$

7

$$W = 2 + 2x_2 - x_3 + 3x_4 \to \max$$
$$\begin{cases} -x_1 + x_2 - 2x_4 \ge -1 \\ x_1 + x_3 + x_4 \ge 1 \\ x_2 + x_3 - x_4 \ge 1 \\ x_3 \le 4; x_4 \le 10 \end{cases}$$

8

$$W = x_1 - 10x_2 + 100x_3 \to \max$$
$$\begin{cases} x_1 + x_2 + x_3 \le 1 \\ x_1 - x_2 - x_3 \le 2 \\ -x_1 + 2x_3 \le 0 \\ x_1 + 2x_3 \le 5 \end{cases}$$

9

$$W = x_1 + 2x_2 + 3x_3 + 1 \to \max$$
$$\begin{cases} x_1 + x_2 + x_3 \le 3 \\ x_1 + x_2 \le 2 \\ x_1 + x_3 \le 2 \\ x_2 + x_3 \le 2 \end{cases}$$

10

$$W = 2 + x_1 - x_2 + 2x_3 \to \min$$
$$\begin{cases} x_1 + x_2 \ge 2 \\ x_1 - x_2 \le 0 \\ x_1 + x_3 \ge 2 \\ x_1 + x_2 - x_3 \le 3 \end{cases}$$

11

$$W = 2 + x_1 + 2x_2 \to \min$$
$$\begin{cases} x_1 + x_2 \ge 2 \\ x_1 + 2x_2 \ge 3 \\ -2x_1 + 2x_2 \ge -4 \\ x_1 + 3x_2 \ge 4 \\ 3x_1 + x_2 \ge 4 \end{cases}$$

12

$$W = x_1 + x_3 \to \min$$
$$\begin{cases} x_1 + x_2 + x_3 \le 4 \\ x_1 - x_2 + x_3 \le 2 \\ x_1 + x_2 - x_3 \ge 2 \\ -x_1 + x_2 + x_3 \le 2 \end{cases}$$

13

$$W = 2 + x_1 + 3x_3 - x_4 \to \min$$
$$\begin{cases} x_1 + x_2 + x_3 - x_4 \le 2 \\ x_1 + x_2 - x_3 + x_4 \le 2 \\ x_1 - x_2 + x_3 + x_4 \le 2 \\ -x_1 + x_2 + x_3 + x_4 \le 2 \end{cases}$$

14

$$W = x_2 + 2x_3 - x_4 \to \min$$
$$\begin{cases} -x_1 + x_2 - 2x_4 \ge -1 \\ x_1 + x_3 + x_4 \ge 1 \\ x_2 + x_3 - x_4 \ge 1 \end{cases}$$

15

$$W = -x_1 - x_2 + 2x_4 \to \min$$
$$\begin{cases} x_1 + x_2 + x_3 + 2x_4 + 2x_5 \le 3 \\ x_1 + x_3 + x_4 + 2x_5 \le 2 \\ x_1 + x_2 + 2x_4 - x_5 \le 2 \end{cases}$$

16

$$W = x_1 + x_3 \to \max$$
$$\begin{cases} 2x_1 - 7x_2 + 22x_3 \le 22 \\ 2x_1 - x_2 + 6x_3 \le 6 \\ 2x_1 - 5x_2 + 2x_3 \le 2 \\ -4x_1 + x_2 + x_3 \le 1 \end{cases}$$

17

$$W = 3x_1 + 2x_2 + 3x_3 + 4x_4 - x_5 \to \min$$
$$\begin{cases} x_1 + x_2 + 2x_5 \ge 1 \\ x_1 + x_3 \ge 1 \\ x_1 + x_4 \ge 1 \end{cases}$$

18

$$W = -x_1 + x_2 \to \max$$
$$\begin{cases} x_1 - 2x_2 \ge 2 \\ 2x_1 - x_2 \ge 2 \\ x_1 + x_2 \ge 5 \end{cases}$$

19

$$W = x_1 - x_2 + 3x_3 + x_4 \to \max$$
$$\begin{cases} x_1 - x_2 + x_4 \le 1 \\ x_2 - x_3 + x_4 \le 1 \\ x_1 + x_3 + 2x_4 \le 2 \\ -2x_2 + x_4 \le 0 \end{cases}$$

20

$$W = x_1 + x_2 + 3x_3 - x_4 \to \max$$
$$\begin{cases} x_1 - 5x_2 + 4x_3 \le 5 \\ x_1 - 2x_2 - 3x_3 \le 4 \\ x_1 + 6x_2 + 5x_3 \le 4 \\ x_2 + x_3 \le 1 \end{cases}$$

21

$$W = x_1 + x_2 + x_3 + 1 \to \min$$
$$\begin{cases} x_1 + x_2 \ge 0 \\ x_1 + x_3 \ge 1 \\ x_2 - x_3 \ge 1 \\ x_1 + 2x_2 + 3x_3 \ge 0 \end{cases}$$

22

$$W = x_1 + x_2 + 3 \to \max$$
$$\begin{cases} x_1 - x_2 \le 1 \\ x_1 - 2x_2 \ge -2 \\ -x_1 + x_2 \ge -1 \\ 2x_1 + x_2 \ge -2 \end{cases}$$

23

$$W = -3 + x_1 + 3x_2 + 5x_3 \to \max$$
$$\begin{cases} x_1 - x_2 + x_3 \le 1 \\ 2x_1 + x_2 + x_3 \le 1 \\ x_1 + 2x_2 + x_3 \le 1 \\ x_1 + x_2 + 2x_3 \le 1 \end{cases}$$

24

$$W = -5 + x_1 - x_2 - 3x_3 \to \max$$
$$\begin{cases} x_1 + x_2 \ge 2 \\ x_1 - x_2 \le 0 \\ x_1 + x_3 \ge 2 \\ x_1 + x_2 - x_3 \le 3 \end{cases}$$

25

$$W = -3 - 2x_1 - x_3 - 2x_4 \to \min$$
$$\begin{cases} x_1 - x_2 + 2x_3 + x_4 \ge 1 \\ 2x_1 - x_2 + x_3 - x_4 \ge 1 \\ x_1 - 2x_2 + x_3 - x_4 \ge -1 \\ x_1 + x_2 + x_3 + 2x_4 \le 5 \end{cases}$$

26

$$W = x_1 + x_2 - x_3 - 2x_4 \to \min$$
$$\begin{cases} x_1 - x_2 + x_3 + x_4 \le 1 \\ -x_1 - x_4 \le 5 \\ x_2 + x_3 \le 10 \end{cases}$$

27

$$W = x_1 + x_2 + x_3 - 3x_4 \to \min$$
$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 1 \\ x_1 - x_2 - x_4 \le 2 \\ x_2 + x_3 + x_5 \le 1 \end{cases}$$

28

$$W = x_1 + x_2 + 4x_3 \to \max$$
$$\begin{cases} x_1 + x_2 + x_3 \le 1 \\ -x_1 + x_2 + x_3 \le 1 \\ -x_1 - x_2 + x_3 \le 1 \\ x_1 - x_2 + x_3 \le 1 \end{cases}$$

29

$$W = x_1 + x_2 + x_3 - x_4 \to \max$$
$$\begin{cases} 3x_1 - x_2 \le 2 \\ x_2 - 2x_3 \le -1 \\ 4x_3 - x_4 \le 3 \\ 5x_1 + x_4 \ge 6 \end{cases}$$

30

$$W = -x_1 - x_2 + 2x_3 \to \min$$
$$\begin{cases} x_1 + x_2 + x_3 \le 1 \\ x_1 - x_2 - x_3 \le 2 \\ -x_1 + 2x_3 \le 2 \end{cases}$$

Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание.
4. Вывод по работе

Практическая работа № 11

Тема работы: Работа в сети Интернет

Цель работы: освоение приемов работы с браузером Internet Explorer; изучение среды браузера и его настройка; получение навыков извлечения web-страниц путем указания URL-адресов; навигация по гиперссылкам.

Оборудование, приборы, аппаратура, материалы: персональный компьютер с выходом в Интернет, браузер Internet Explorer.

Краткие теоретические сведения.

Браузер – это программа для просмотра web-страниц.

Настройка браузера. Все браузеры позволяют выполнить некоторые настройки для оптимизации работы пользователей в Интернете. В браузере Internet Explorer основная часть настроек содержится в меню Сервис – Свойства обозревателя.

Вкладка Общие позволяет задать адрес домашней страницы, которая будет автоматически загружаться в окно браузера при его запуске, цвета гиперссылок по умолчанию, название шрифта по умолчанию. Здесь же определяется сколько дней будет храниться ссылка посещенных страниц в журнале. Кроме того, для ускорения просмотра. Все посещенные страницы помещаются в специальную папку, и с помощью кнопки Параметры можно задать разные способы обновления таких страниц.

С помощью вкладки Безопасность можно создать списки надежных узлов и узлов с ограниченными функциями. Зона Интернет будет при этом включать все остальные узлы, не вошедшие в эти две папки. Для каждой из них с помощью кнопки Другой можно изменить параметры безопасности, установленные для них по умолчанию. Здесь можно запретить выполнение сценариев, отображение всплывающих окон, загрузку файлов и т.д.

Вкладка Конфиденциальность дает возможность настроить работу с файлами cookie, с помощью которых информация о пользователе автоматически передается на сервер.

Вкладка Содержание позволяет ограничить доступ к некоторой информации (насилие, ненормативная лексика и т.д.).

Вкладка Подключения позволяет установить подключение к Интернету.

На вкладке Дополнительно можно задать некоторые дополнительные параметры работы (отключить загрузку графических изображений, отменить подчеркивание ссылок, запретить отладку сценариев и т.д.).

Вкладка Программы позволяет определить программы, которые будут по умолчанию использоваться службами Интернета (почтовые программы, html-редакторы и т.п.).

Задания

Задание 1. Изучите элементы среды Internet Explorer, возможности настройки этого браузера. Занесите в список надежных узлов сайты http://www.gismeteo.ru, http://www.yandex.ru. Запретите загрузку файлов. Заблокируйте всплывающие окна.

Задание 2. Восстановите настройки Internet Explorer по умолчанию.

Задание 3. Зайдите на сайт интернет-библиотеки по адресу http://www.internet-biblioteka.ru, зарегистрируйтесь. Изучите правила работы с библиотекой. Найдите книгу Комоловой Н. "Компьютерная верстка и дизайн. Самоучитель". Скачайте ее. Составьте список книг библиотеки по информатике. Список сохраните в своей папке в документе MS Writer под именем ПР11_1.doc.

Задание 4. Изучите новости Тульской области, открыв, например, адрес http://gagarincity.ru/smolnews/. Сохраните последние новости в документе Writer под именем ПР11_2.doc.

Задание 5. Зайдите на сайт турагентства по адресу http://agency.travelplus.ru. Изучите возможности организации тур-поездок на ближайший месяц по России. Сохраните ближайшие туры в текстовом документе под именем ПР11_3.txt.

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и порядок его выполнения.
4. Вывод по работе.

6. Контрольные вопросы

1. Что такое браузер?
2. Как осуществить настройку браузера?
3. Для чего нужна адресная строка в браузере?
4. Как осуществить поиск информации в Интернете с помощью браузера?

Практическая работа № 12-13

Тема работы: Программирование алгоритмов

Цель работы:

1. Познакомится с интерфейсом интерпретатора языка Free Pascal.
2. Научиться решать задачи с использованием алгоритмов линейной, разветвляющейся и циклической структуры.

Оборудование, приборы, аппаратура, материалы: персональный компьютер, среда программирования Free Pascal.

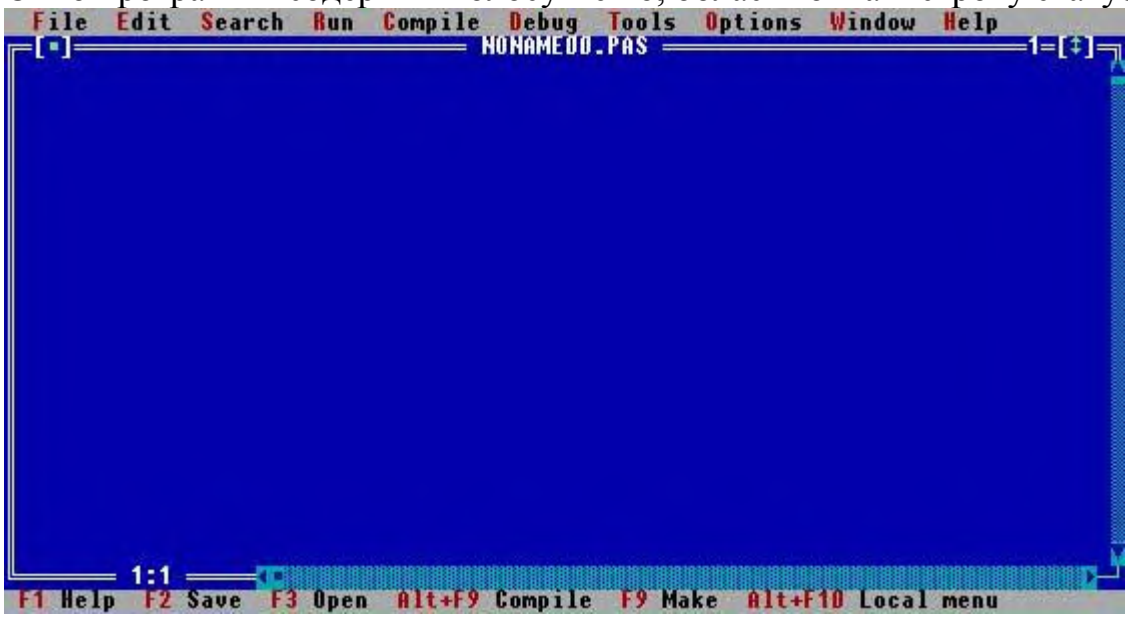
Краткие теоретические сведения

Паскаль - язык профессионального программирования, который назван в честь французского математика и философа Блеза Паскаля (1623-1662) и разработан в 1968-1971 гг. Никлаусом Виртом. Первоначально был разработан для обучения, но вскоре стал использоваться для разработки программных сред **Паскаль** - это система программирования, созданная для повышения качества и скорости разработки программ (80-е гг.).

Систему программирования Free Pascal называют интегрированной (integration - объединение отдельных элементов в единое целое) средой программирования, т.к. она включает в себя редактор, компилятор, отладчик, имеет сервисные возможности.

Для запуска интегрированной среды программирования нужно установить текущим каталог с Free Pascal и ввести команду: turbo.exe.

Окно программы содержит полосу меню, область окна и строку статуса.



Для входа в меню можно воспользоваться одним из способ:

- с помощью "мышки";
- с помощью клавиши F10;

- с помощью комбинации Alt+<выделенная буква>. О том, что мы в меню свидетельствует курсор - прямоугольник зеленого цвета.

Интегрированная среда программирования Турбо Паскаль позволяет иметь любое количество открытых окон, но в любой момент времени активным может быть только одно.

Активное окно - это окно с которым вы в настоящий момент работаете.

Общие горячие клавиши:

- F1 - выводит окно подсказки;
- F2 - сохраняет файл активного окна;
- F3 - появление диалогового окна и возможность открыть файл;
- F4 - запускает программу до строки, на которой стоит курсор;
- F5 - масштабирует диалоговое окно;
- F6 - переходит к следующему открытому окну;
- F7 - запускает программу в режиме отладки с заходом внутрь процедур;
- F8 - запускает программу в режиме отладки, минуя вызов процедур;
- F9 - компилирование программы в текущем окне;
- F10 - возвращение в меню.

Команды меню File:

- Open-F3 - открыть существующий файл (при активизации этой опции появляется окно со списком файлов, где можно выбрать необходимый),

- New - создать новый файл (очищает память редактора и переводит в режим создания нового файла, которому присваивается имя Noname.pas; имя можно изменить при записи файла на диск),
- Save-F2 - сохранить файл (переписывает файл из памяти редактора на диск),
- Save as - сохранить с новым именем,
- Save all - сохранить все в окнах (записывает содержимое всех окон редактора в соответствующие файлы),
- Change dir - смена каталога (позволяет изменить установленный по умолчанию диск или каталог),
- Print - печать файла,
- Get info - выдача информации о текущем состоянии программы и используемой памяти,
- DOS Shell - выход в DOS без выгрузки из памяти (для возврата ввести команду exit),
- Exit - выход и выгрузка из памяти.

1. Программы на языке Паскаль имеют блочную структуру:
2. Блок типа PROGRAM - имеет имя, состоящее только из латинских букв и цифр. Его присутствие не обязательно, но рекомендуется

записывать для быстрого распознавания нужной программы среди других листингов.

3. Программный блок, состоящий в общем случае из 7 разделов:

- раздел описания модулей (uses);
- раздел описания меток (label);
- раздел описания констант (const);
- раздел описания типов данных (type);
- раздел описания переменных (var);
- раздел описания процедур и функций;
- раздел описания операторов.

Общая структура программы на языке Паскаль:

```
Program ИМЯ..; {заголовок программы}
Uses ...; {раздел описания модулей}
Var ..; {раздел объявления переменных}
...
Begin {начало исполнительной части программы}
... {последовательность
... операторов}
End. {конец программы}
```

Пример программы, которая осуществляет сложение двух чисел и выводит сумму на экран:

```
Program Summa;
Uses
Crt;{Подключаем модуль Crt}
Var
  number1, {переменная, в которой будет содержаться первое число}
  number2, {переменная, в которой будет содержаться второе число}
  rezult {переменная, в которой будет содержаться результат}
    :integer; {указывает тип целых чисел}
Begin
  ClrScr;{Используем процедуру очистки экрана из модуля Crt}
  Write ('Введите первое число ');
  {Выводим на экран символы, записанные между апострофами}
  Readln (number1);
  {Введенное пользователем число считываем в переменную number1}
  Write ('Введите второе число ');
  {Выводим на экран символы, записанные между апострофами}
  Readln (number2);
  {Введенное пользователем число считываем в переменную number2}
  rezult := number1 + number2;
  {Находим сумму введенных чисел и присваиваем переменной rezult}
  Write ('Сумма чисел ', number1, ' и ', number2, ' равно ', rezult);
  {Выводим на экран строчку, содержащую ответ задачи}
```

```
  Readln;{Процедура задержки экрана}
End.
```

4. Задания

Задание 1.

Изучите внешний вид системы программирования Free Pascal. Запустите программу Free Pascal.

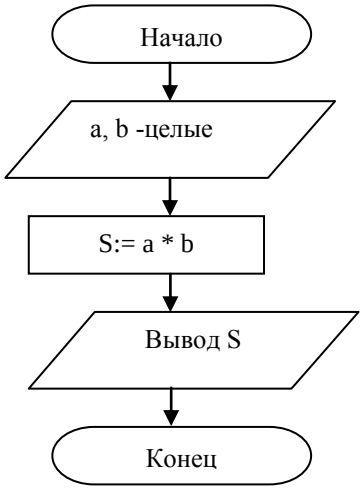
Некоторые команды и служебные клавиши интерпретатора языка Turbo Pascal.	
Выход в верхнее меню	F10
Запуск программы на выполнение	Ctrl + F9 или F10 - Run - Run
Помощь	F1
Смена алфавита	Левый (Shift+Alt)
Изменение размера окна и подключение «мыши»	Ctrl +F5
Создание нового листа	F10 – File - New
Переключение между листами	Alt + 0 или F10 – Window - List
Закрытие программы	Alt + X

Задание 2.

Задача. Даны длины сторон прямоугольника. Составить программу нахождения площади данной фигуры.

а) Ввести текст программы в среде интерпретатора языка Free Pascal. Запустить программу на выполнение. Если компилятор выдал сообщение о наличии ошибки необходимо, выполнить отладку и запустить программу заново.

Блок - схема	**Программа на языке Free Pascal.**

	**program** primer1;
	var a, b, s: integer;
	begin
	  write ('ВВЕДИ a, b ');
	  readln (a, b);
	  s: = a * b;
	  write ('S= ', s);
	  readln;
	end.

б) Часть программы, соответствующую вводу данных оформить так:

```
  write (' a=  ');
  readln (a);
  write (' b=  ');
  readln (b);
```

Привести оператор вывода к следующему виду **write** ('S= ' , a , '* ' , b , ' = ' , s, ' см ');

Запустить программу на выполнение.

Задание 3.

Изменить текст программы так, чтобы кроме значения площади прямоугольника вычислялось и выводилось на экран значение периметра фигуры.

Для этого внесите соответствующие изменения в раздел описания переменных, добавьте соответствующий оператор присваивания и операторы ввода и вывода данных.

Задание 4.

Составьте программу вычисляющую значения функции $y = \dfrac{a^2 + 1}{\sqrt{a^2 + 1}}$, при любом значении а.

Решение оформить аналогично образцу в задании 2

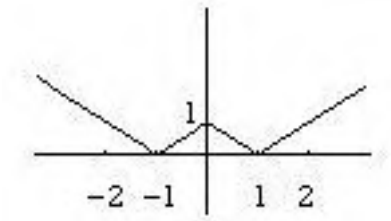
Краткие теоретические сведения.

Алгоритм называется **разветвляющимся**, если он содержит несколько ветвей, отличающихся друг от друга содержанием вычислений. Выход вычислительного процесса на ту или иную ветвь алгоритма определяется исходными данными задачи.

Перед выполнением работы необходимо ознакомится с правилами записи логических выражений, операций сравнения, операторов IF, CASE.

Пример:

Дано действительное а. Для функций f(a),график которой представлен на рисунке, вычислить f(a).

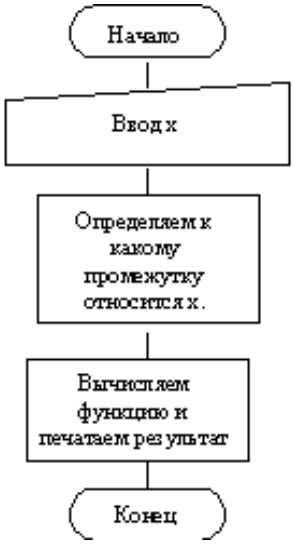


Этапы решения задачи:

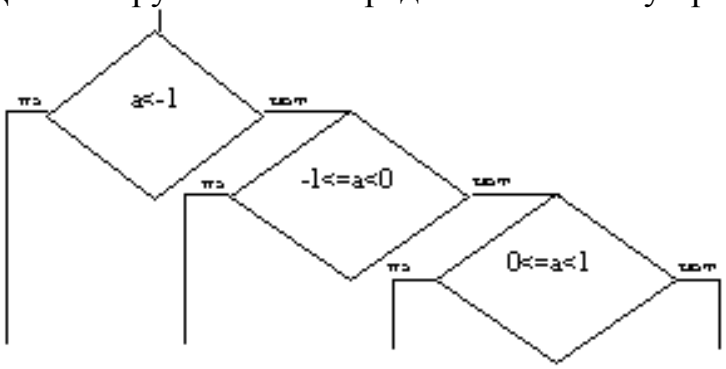
1. Математическая модель: функция вычисляется по следующей формуле

$$f(x) = \begin{cases} -x-1, & x < -1 \\ x-1, & -1 \le x < 0 \\ -x+1, & 0 \le x < 1 \\ x+1, & x \ge 1 \end{cases}$$

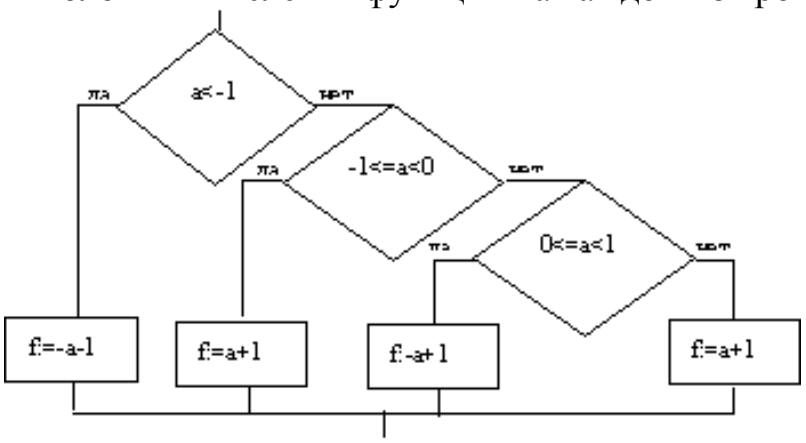
2. Составим схему алгоритма



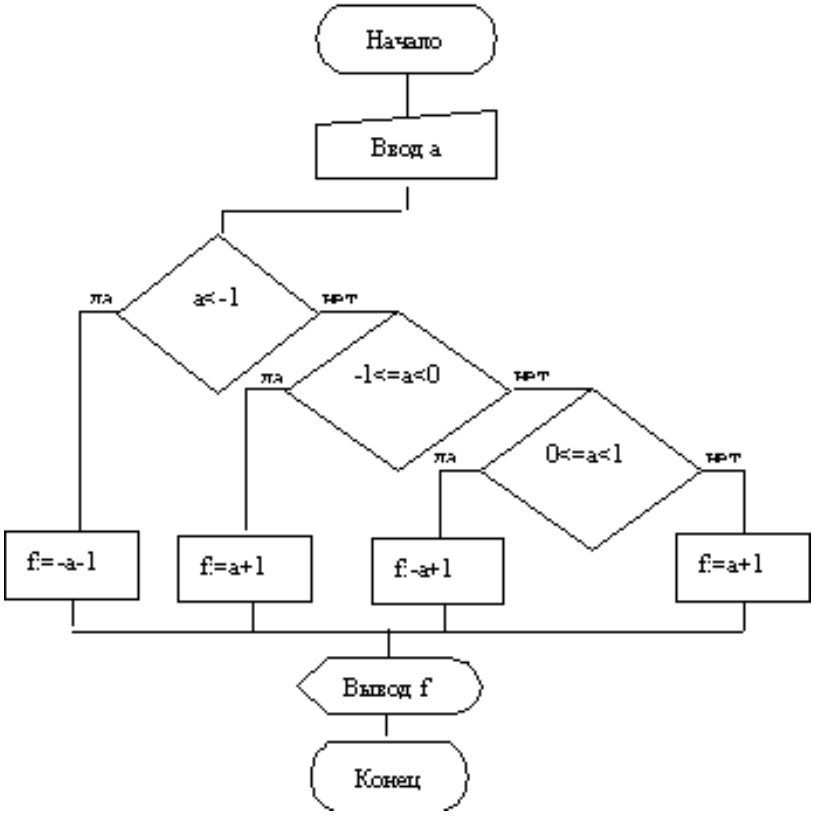
Детализируем блок "Определяем к какому промежутку относится x."



Добавим блоки вычисления функции на каждом из промежутков:



Таким образом, окончательный алгоритм выглядит следующим образом:



Дальнейшая детализация не требуется. Переводим блок-схему на язык Паскаль.

```
Program example1;
var a, f:real;
begin
   write('Введите a> '); readln(a);
     if a<-1 then f:= -a-1 else
       if (a>=-1) and (a<0) then f:= a+1 else
       if (a>=0) and (a<1) then f:= -a+1 else f:= a+1;
     writeln('F= 'f);
end.
```

Задания.

1. Даны действительные положительные числа x, y, z. Выяснить, существует ли треугольник с длинами сторон x, y, z.

2. Дано действительное a. Для функции f(a), график которой представлен на рисунке, вычислить f(a).

3. Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу (1, 3).

4. Даны действительные числа x, y. Если x, y отрицательны, то каждое значение заменить его модулем; если отрицательное только одно из них, то оба значения увеличить на 0.5; если оба значения не отрицательны и ни одно из них

не принадлежит отрезку [0.5, 2.0], то оба значения уменьшить в 10 раз; в остальных случаях x, y оставить без изменения.

5. Определить и вывести на печать номер квадранта, в котором расположена точка M(x,y), x и y заданные вещественные числа.

6. Из величин, определяемых выражениями a=sinx, b=cosx, c=ln|x| при заданном x, определить и вывести на экран дисплея минимальное значение.

7. Определить, какая из двух точек - M1(x1,y1) или M2(x2,y2) - расположена ближе к началу координат. Вывести на экран дисплея координаты этой точки.

8. Определить, какая из двух фигур (круг или квадрат) имеет большую площадь. Известно, что сторона квадрата равна a, радиус круга r. Вывести на экран название и значение площади большей фигуры.

9. Определить, попадает ли точка M(x,y) в круг радиусом r с центром в точке (x0,y0)

Краткие теоретические сведения.

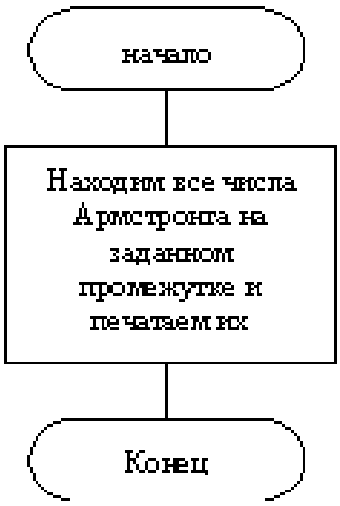
Алгоритм называется **циклическим**, если он содержит многократное выполнение одних и тех же операторов при различных значениях промежуточных данных. Число повторений этих операторов может быть задано в явной (цикл с известным заранее числом повторений) или неявной (цикл с неизвестным заранее числом повторений) форме.

Перед выполнением работы необходимо изучить различные схемы организации циклов и операторы FOR, WHILE, REPEAT.

Пример: На промежутке от 1 до M найти все числа Армстронга. Натуральное число из n цифр называется числом Армстронга, если сумма его цифр, возведенных в n-ю степень, равна самому числу.

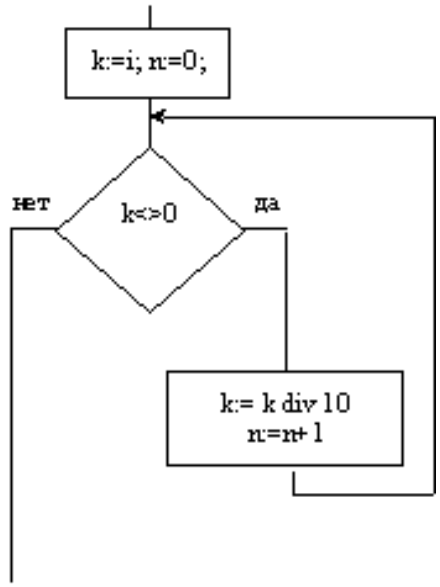
Этапы решения задачи:

1. Математическая модель: xO[1;M], x=
2. Составим блок схему программы:

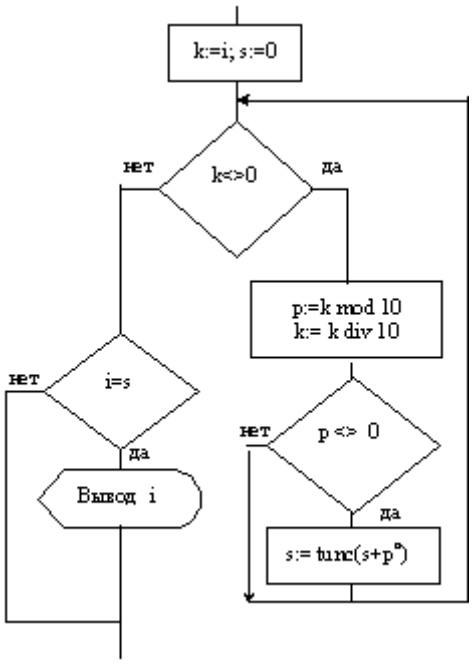


Распишем составные части блока"Находим все числа Армстронга на заданном промежутке и печатаем их"

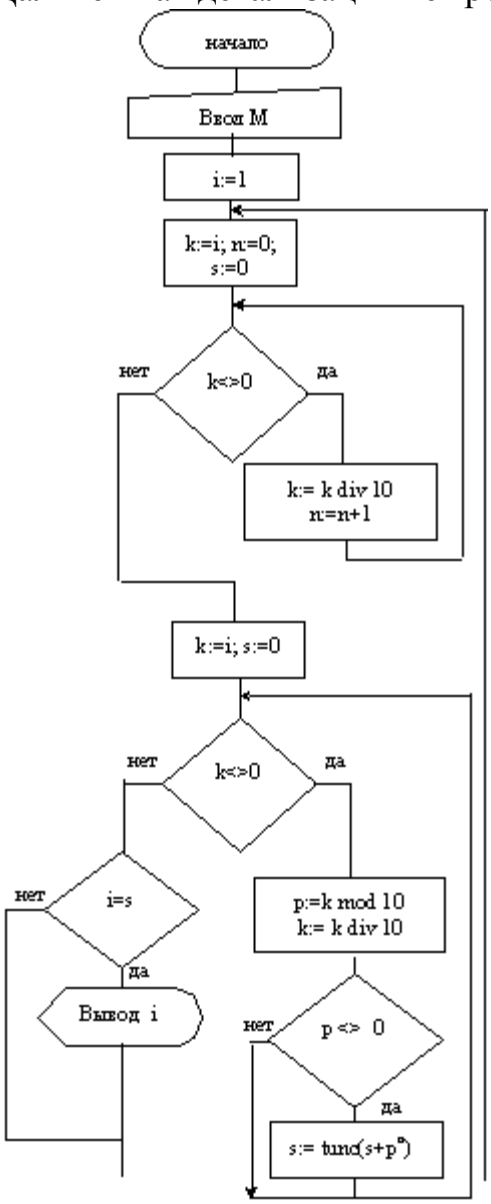
Опишем блок "Подсчитываем сколько цифр в числе i"



Опишем блок "Проверяем, является ли i числом Армстронга"



Дальнейшая детализация не требуется, запишем блок-схему целиком:



Дальнейшей детализации не требуется, переведем программу на язык Паскаль.

```
PROGRAM Primer_1;
var i,k,s,p,n: Integer;
  BEGIN
    Write('Введите M '); Readln(m);
            For i:=1 to M do
      begin
        s:=0; k:=i; n:=0;
        While k<>0 do
          begin k:=k DIV 10; n:=n+1 end;
            k:=i;
            While k<>0 do
            begin p:=k MOD 10; k:=k DIV 10;
              If p<>0 then s:=Trunc (s+Exp(n*Ln(p)))
            end;
            If s=f then WriteLn (f)
      end;
END.
```

Задания.

Вычисление выражений:

Дано натуральное n.

Вычислить:

1. $(1+\frac{1}{1^2})+(1+\frac{1}{2^2})+(1+\frac{1}{3^2})+\ldots+(1+\frac{1}{n^2})$;

2. $\frac{1}{\sin 1}+\frac{1}{\sin 1+\sin 2}+\ldots+\frac{1}{\sin 1+\ldots \sin n}$;

Дано действительное число x, натуральное число n. Вычислить:

3.x (x - n)(x - 2 n)(x - 3 n)…(x - n2);

4. $\frac{1}{x}+\frac{1}{x(x+1)}+\ldots+\frac{1}{x(x+1)\ldots(x+n)}$;

5. $\frac{x^1}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}+\ldots+\frac{x^n}{n!}$;

Дано натуральное n. Вычислить:

6. $\prod_{i=1}^{n}(2+1/i!)$;

$\sum_{i=1}^{n}(1+i)/i!$

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод по работе.

Контрольные вопросы

1. Какие команды текстового редактора вы знаете?
2. Что такое блок текста и как его выделить?
3. Какие операторы используются для программирования разветвлений?
4. Как выполняются операторы перехода?
5. Какую из функций: Sin(x), Abs(x), Trunc(x) можно заменить условным оператором if x<0 then x := -x?
6. Как записывается и как работает оператор FOR?
7. Для организации каких циклов применим оператор FOR?
8. В чем отличие оператора WHILE от оператора REPEAT?
9. Как программируются циклические алгоритмы с явно заданным числом повторений цикла?
10. Как программируются циклические алгоритмы с заранее неизвестным числом повторений цикла?
11. Напишите оператор цикла, который не выполняется ни разу.
12. Напишите оператор цикла, который выполняется неограниченное число раз.

Минобрнауки России
ФГБОУ ВО «Тульский государственный университет»
Технический колледж им. С.И. Мосина

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ И САМОСТОЯТЕЛЬНЫХ РАБОТ

по дисциплине

Основы алгоритмизации и программирования

по специальности 10.02.05 Обеспечение информационой безопасности автоматизированных систем

2024 г.

УТВЕРЖДЕНЫ

Цикловой комиссией информационных технологий

Протокол от «12» 09 2024 г. № 2

Председатель цикловой комиссии _____ И.В.Миляева

Авторы: Афанасьева С.М., канд.техн. наук

Содержание

Практическое занятие № 1 4
Разработка линейных алгоритмов и алгоритмов ветвленияя 4
Практическое занятие № 2 10
Разработка циклических алгоритмов 10
Практическое занятие № 3 14
Знакомство с инструментальной средой программирования 14
Практическое занятие № 4 16
Разработка программ последовательной структуры 16
Практическое занятие № 5 19
Разработка программ разветвляющейся структуры 19
Практические занятия № 6,7,8 24
Разработка программ с использованием цикла с предусловием. Разработка программ с использованием цикла с постусловием. Разработка программ с использованием цикла с параметром 24
Практическое занятие № 9 29
Разработка программ с использованием одномерных массивов 29
Практическое занятие № 10 35
Разработка программ с использованием двумерных массивов. Указатели 35
Практическое занятие № 11 42
Разработка программ с использованием строк 42
Практическое занятие № 12 49
Разработка программ с использованием функций 49
Практическое занятие № 13 54
Разработка программ с использованием рекурсивных функций 54
Практическое занятие № 14 56
Разработка программ работы с файлами 56
Практическое занятие № 15 61
Основы работы с интегрированной средой 61
Практическое занятие № 16 64
Создание интерфейса пользователя в среде разработки программ 64
Практическое занятие № 17 71
Организация классов и принцип инкапсуляции 71
Практическое занятие № 18 78
Программная реализация принципов наследования и полиморфизма 78
Практическое занятие № 19 84
Работа с графикой. Рисование графических примитивов 84
Практическое занятие № 20 88
Работа с графикой. Рисование анимированных объектов 88

Практическое занятие № 1

Разработка линейных алгоритмов и алгоритмов ветвленияя

Цель работы: научиться составлять базовые структоры алгоритмов – линейный ветвящиеся алгоритмы.

Краткие теоретические сведения.

Линейным называется алгоритм, в котором действия выполняются последовательно – одно за другим.

Разветвляющийся называется алгоритм, в котором действие выполняется по одной из возможных ветвей решения задачи, в зависимости от выполнения условий. В отличие от линейных алгоритмов*, в которых команды выполняются последовательно одна за другой*, в разветвляющиеся алгоритмы входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (действий) – ветка (рисунок 1).

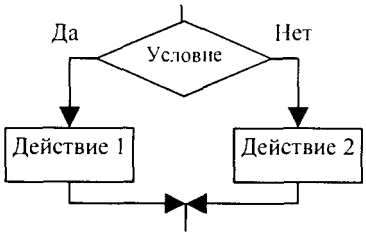


Рисунок 1

В качестве условия в разветвляющемся алгоритме может быть использовано любое понятное исполнителю утверждение, которое может соблюдаться (быть истинно) или не соблюдаться (быть ложно). Такое утверждение может быть выражено как словами, так и формулой. Таким образом, алгоритм ветвления состоит из условия и двух последовательностей команд.

Ветвящийся процесс, включающий в себя две ветви, называется простым, более двух ветвей — сложным. Сложный ветвящийся процесс можно представить с помощью простых ветвящихся процессов.

Примером может являться разветвляющийся алгоритм, изображенный на рисунке 2. Аргументами этого алгоритма являются числа *A* и *B*, а результатом — число *X*. Если условие *A* >= *B* истинно, то выполняется операция умножения чисел *(X = A* * B), в противном случае выполняется операция сложения *(X = A + B)*. В результате печатается то значение *X*, которое получается в результате выполнения одного из действий.

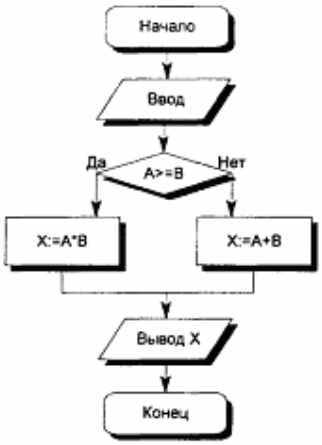


Рисунок 2

Пример выполнения задания

Задание 1: вычислить значение функции

$$y = \frac{\sin^3 c * \cos^2 a}{5 \sin^d b} + \frac{2}{15}$$ при

a = 9,5; b = 1,365; c = 6,5; d = 5.

Блок=схема алгоритма представлена на рисунке

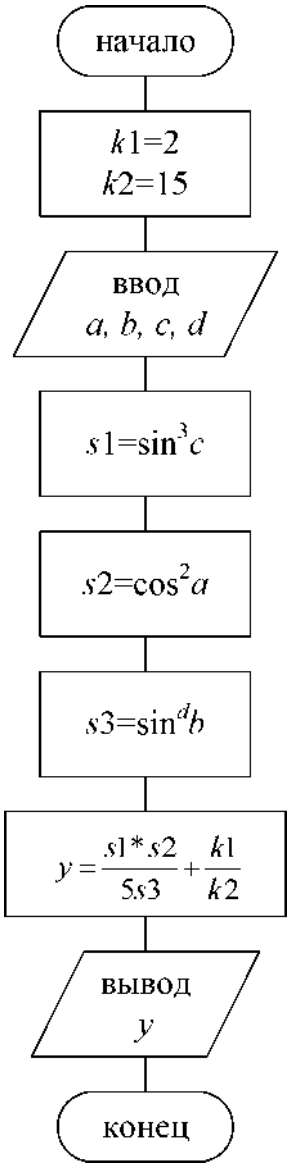


Рисунок 3

Задание 2: по заданным координатам x и y определить, где находится точка (рисунок 3): внутри заштрихованной области; вне заштрихованной области; на границе этой области.

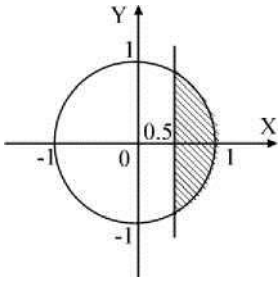


Рисунок 2 — Постановка задачи

Метод решения задачи:

1. Для решения задачи будем использовать уравнение окружности $x^2 + y^2 = R^2$. Так как R=1, то уравнение принимает вид $x^2+y^2 = 1$.
2. Определяем условие, при котором точка будет находиться внутри заштрихованной области: (x>0.5) и (x^2+y^2<1).
3. Определяем условие, при котором точка будет находиться вне заштрихованной области: (x<0.5) или (x^2+y^2>1).

Составим блок-схему решения задачи (рисунок 4).

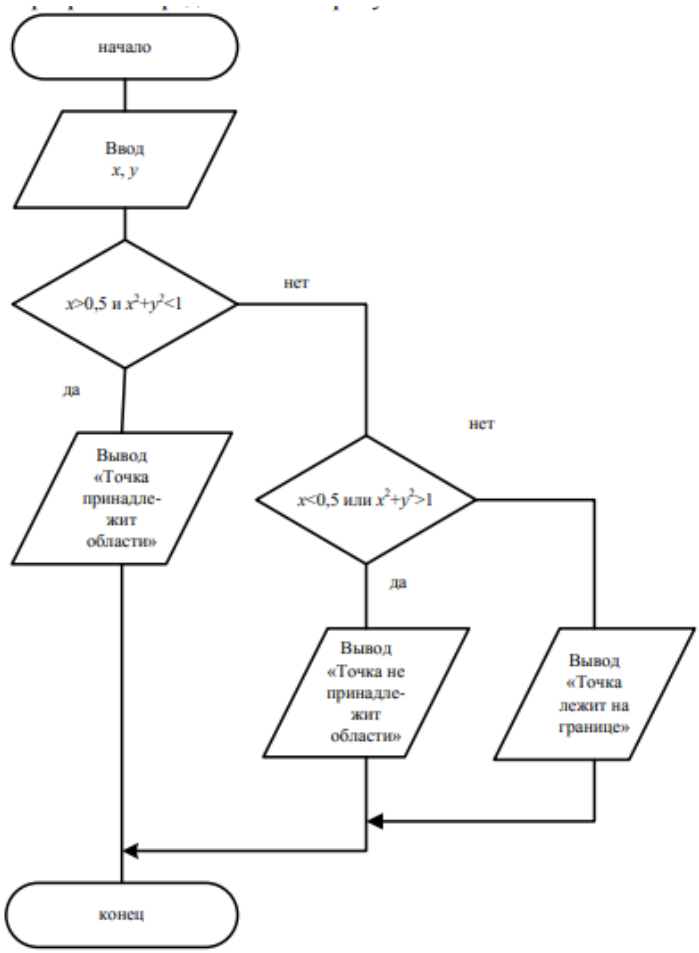


Рисунок5

Варианты заданий

Задание 1 Вычислить значение функции

1 $y = \arccos^{2} A - B \lg(A^{2} + B^{2})$ при A = 5 B = 2,35

2 $Y = \lg^{3(}A/B + tgA)$ при $A = 4$; $B = 13,6$•

3 Y = (sin3 c • cos2 a)/ arcsin d3+ 5sin d b при A = 9.5; B = 1,365; C = 6,6; D = 3

4 Y = 2A • arcsin(A2+B3) при A = 8; B = 5,6 •

5 $Y = 2B \cdot \ln(A - B^{s}) - \sin(D-C2)$ при $A = 5,6$; $B = 2,8$; $D = 3$ $C = 3,6$;

6 y = sin 2a + sin 5a - sin3a /(cos a +1 - $2\sin^{2}2a$) при a = 4 71 •

7 $Y = 1 - \sin^{2} 2a + \cos2a$ при a = 1,57 •

8 $Y = 4\cos^{a} \cos a + \cos4a$ при a = 2,09•

9 $Y = \arccos(A - B) + tg(1-B) \cdot \lg^{3}(A^{2}/B^{2})$ при $A = 5$, $B = 2,35$ •

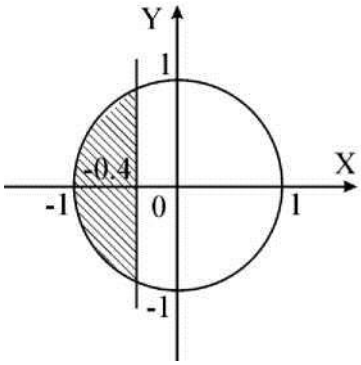
10 $Y = \lg(A/B + tgA) - cos(A+B)$ при $A = 4$; $B = 13,6$ •

Задание 2: По заданным координатам точки определить, где находится точка:

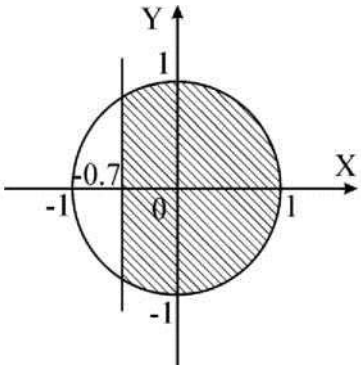
1) внутри заштрихованной области;

2) вне заштрихованной области;

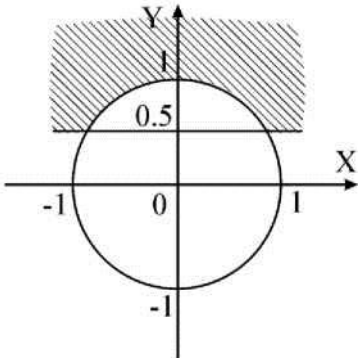
3) на границе этой области.



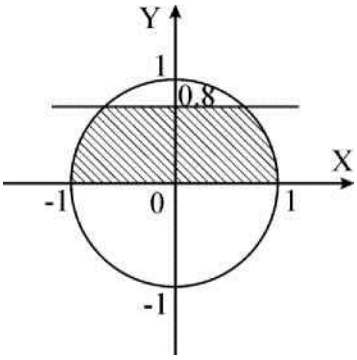
1.



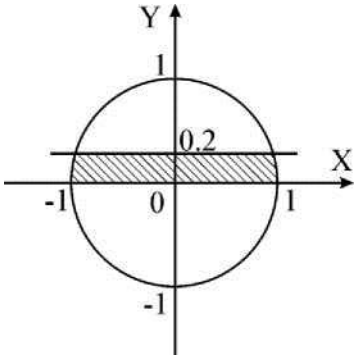
2.



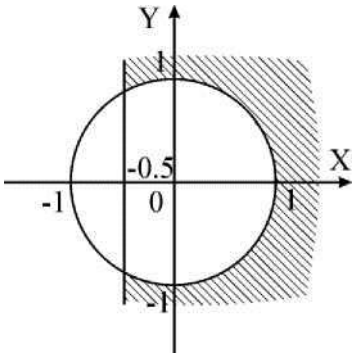
3



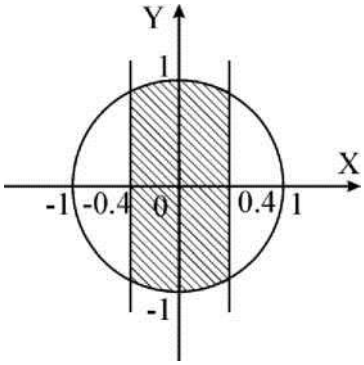
4.



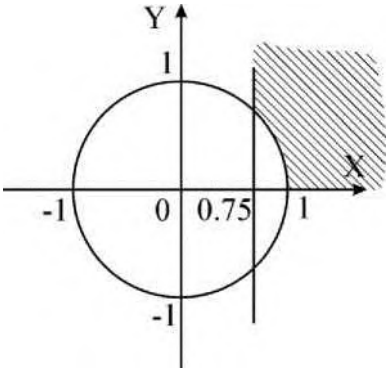
5.



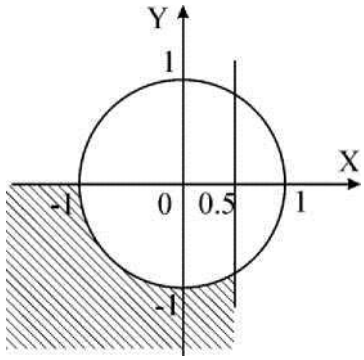
6.



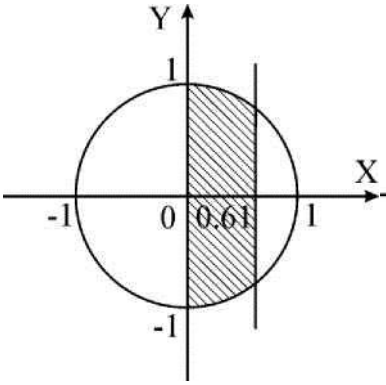
7.



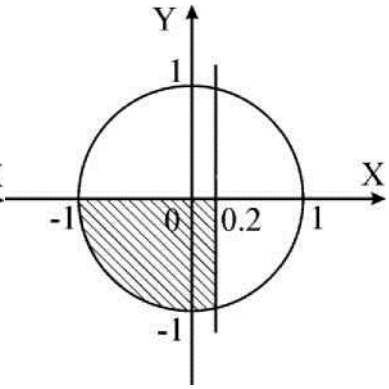
8.



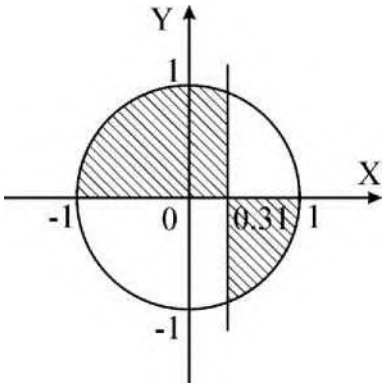
9.



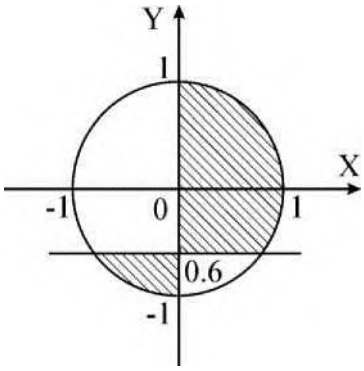
10.



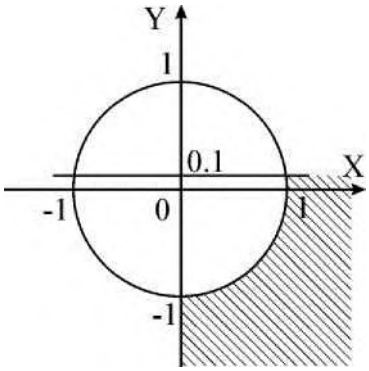
11.



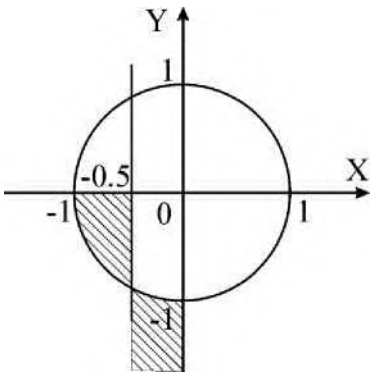
12.



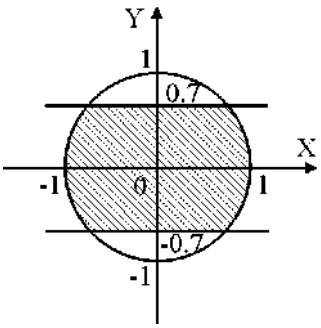
13.



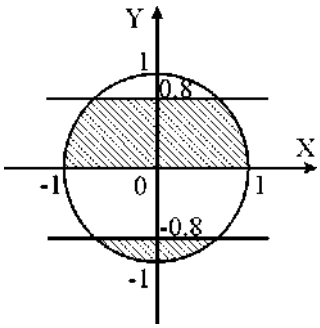
14.



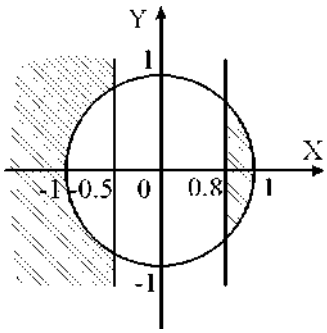
15.



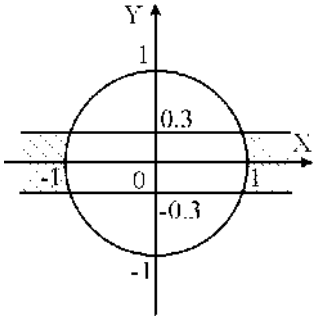
16.



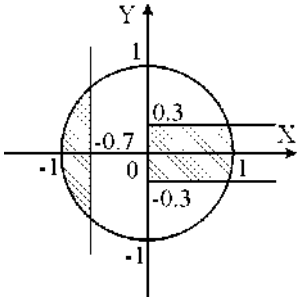
17.



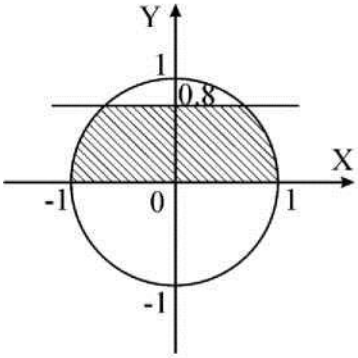
18.



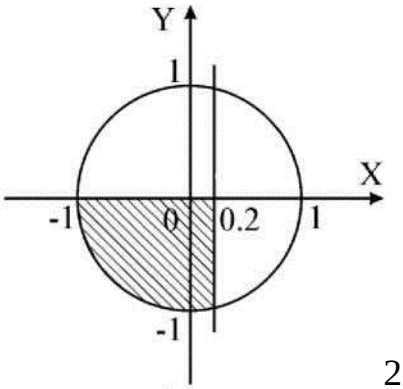
19.



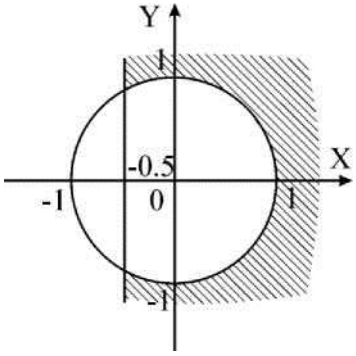
20.



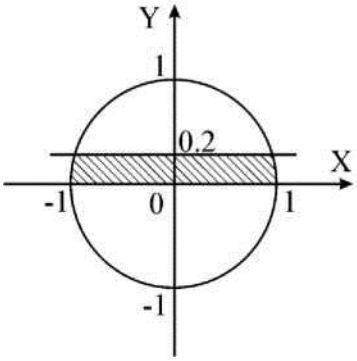
21.



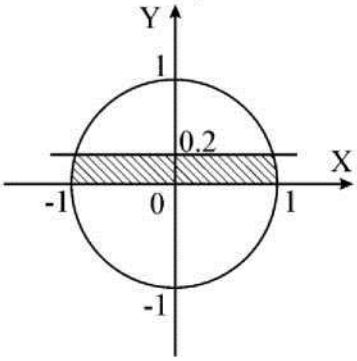
22.



23.



24.



25.

Практическое занятие № 2
Разработка циклических алгоритмов

Цель работы: научиться составлять одну из базовых структор алгоритмов – циклические алгоритмы

Краткие теоретические сведения

Цикл – процесс повторения одних и тех же операций (группы команд).

Тело цикла–это последовательность операций (команд), которая повторяется многократно заданное количество раз или до тех пор, пока не будет выполнено условие. Существуют три схемы представления циклических алгоритмов.

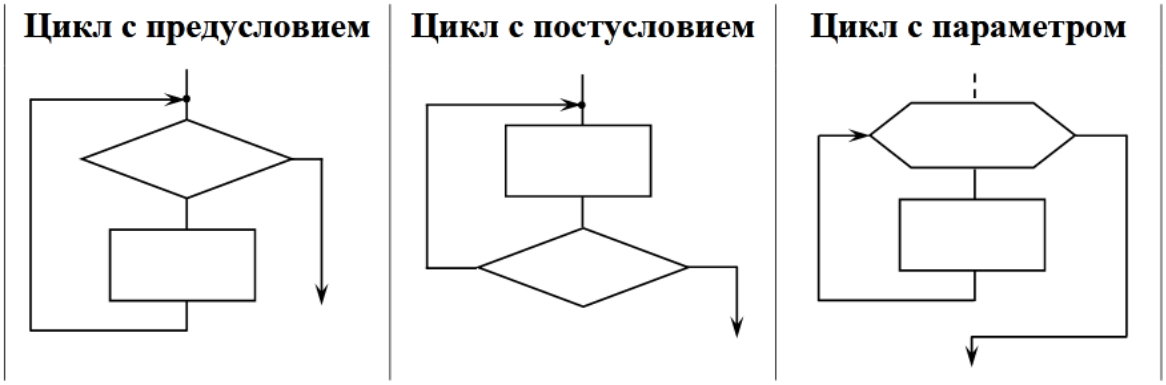


Рисунок 1

Цикл с предусловием

Особенностью первой схемы (цикл с предусловием) является то, что проверка условия проводится до тела цикла. В том случае, если условие выхода из цикла выполняется, то тело цикла не выполняется ни разу. Например, по условию необходимо вывести все точки графика $y=x^2$ на отрезке для x от -5 до 5 с шагом 1 сантиметр. Алгоритм решения задачи может быть представлен блок-схемой (рисунок 2).

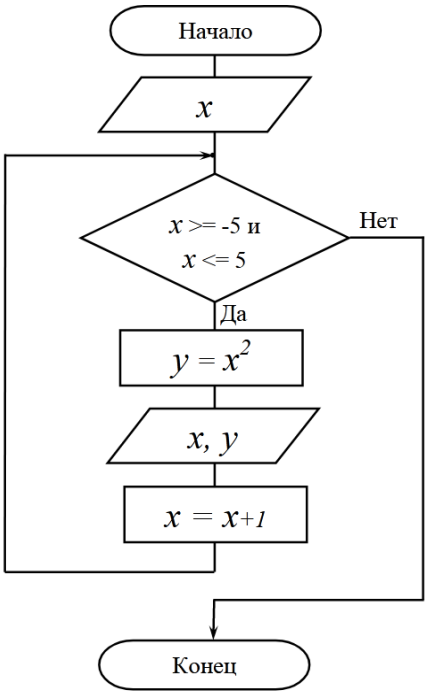


Рисунок 2

По представленному алгоритму с предусловием, если с клавиатуры будет введено стартовое значение хне удовлетворяющее условию $-5 \leq x \geq 5$, то решение задачи закончится после первого же неверного шага.

Цикл с постусловием

При решении задачи по циклическому алгоритму с постусловием в любом случае цикл будет выполнен хотябы один раз, так как первая проверка условия выхода из цикла осуществляется после того, как тело цикла выполнено. Например, блок-схема циклического алгоритма с постусловием для решения той же выше поставленной задачи представлена на рисунке 3.

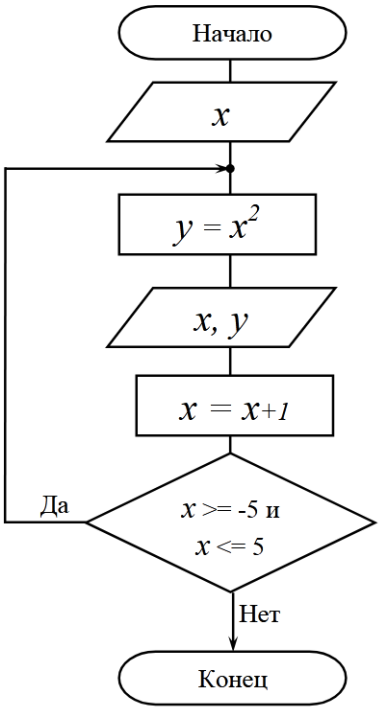


Рисунок 3

На практике обе схемы равноценны по применению частота и логичность их применения зависит от условия решаемой задачи.

Цикл с параметром

В условии многих задач четко оговаривается количество повторений операций тела цикла или, например, задано выполнить действия для всех значений переменной от первого до последнего с заданным шагом изменении.

Параметр цикла – это переменная, через которую организуется (описывается) цикл. Параметр цикла определяет количество повторений (итераций), так как у него обязательно известно - первое значение, при котором тело цикла будет выполнено первый раз, последнее значение, при котором тело цикла будет выполнено последний раз и величина изменения параметра (шаг), на которую надо увеличить параметр после каждой итерации. Шаг – числовой показатель изменения параметра после каждой итерации. Циклы с параметром более просты в понимании и организации. В блок–схеме выделена специальная фигура для обозначения цикла с параметром. В блоке через «;» прописываются все значения параметра - «первое; последнее; шаг». Применительно к решению выше поставленной задачи, параметром цикла, по условию, является *x*.

Для всех значений x от -5 до 5 через 1 сантиметр рассчитывается $y=x^2$ и выводится точка графика (координаты x, y). Блок-схема решения задачи с применением цикла с параметром, представлена на рисунке 4.

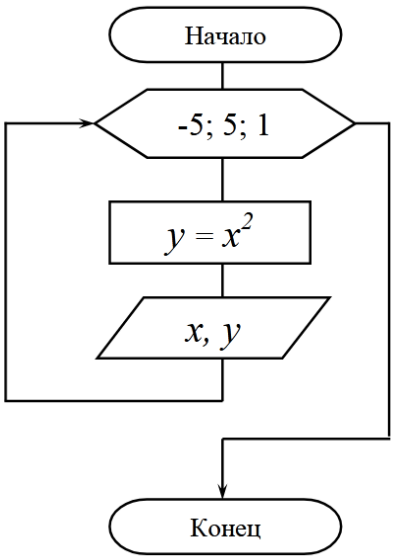


Рисунок 4

Варианты заданий

1 Составьте программу для вычисления степени числа *a* с целым показателем n.
2 Вычислите сумму всех двухзначных чисел.
3 Выведите на экран все чётные числа от 2 до n включительно и подсчитайте их количество.
4 Вычислите значение функции y=cos(x)*tg(pi*x) на интервале [0; pi), где шаг изменения равен 0.1.
5 Имеется товар в ящиках по 16, 17 и 21 кг. Как получить 185 кг этого товара, не вскрывая ящиков?
6 Найдите все делители некоторого целого числа.
7 Определите 5 первых совершенных чисел. Натуральное число называется совершенным, если оно равно сумме всех своих делителей за исключением самого себя. Например, 6 – совершенное число, т.к. 6=1+2+3, число 8 – несовершенное, т.к. 8<>1+2+4).
8 Найти сумму чётных натуральных чисел, меньших 100.
9 Найти сумму нечётных натуральных чисел, меньших 100.
10 Найти сумму целых положительных чисел, больших 20, меньших 100 и кратных 3.
11 Вычисление F = 10! выполнить каждым из трёх операторов цикла.
12 Начав тренировки, спортсмен в первый день пробежал 10 км, каждый последующий день он увеличивал дневную норму на 10% от нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?
13 Самолёт летит из пункта А в пункт Б со средней скоростью υ. Найти время полёта t, если есть встречный ветер, скорость которого $\upsilon_в$изменяется от 0 до 15 м/с с шагом $\Delta\upsilon_в$= 0.5 м/с. Расстояние между пунктами А и Б S= 437 км, υ= 720 км/ч.(Для тех, кто плохо знает физику: $t=S/(\upsilon-\upsilon_в)$)

14 Составить программу, печатающую таблицу значений градусов температуры по Цельсию и Фаренгейту. Значения градусов температуры по Цельсию изменяются от 0˚до 20˚с шагом 1˚. (℉ =℃ * 1.8 + 32)

15 Пользователь сберегательной кассы внёс в неё вклад S_0= 3000 руб. До какой суммы он возрастёт через N= 5 лет, если процент годовых начислений P= 30 % ?

16 Даны действительные числа x, y. Вывести в порядке возрастания все целые числа, расположенные между x и y, а также количество этих чисел.

17 Даны действительные числа x, y. Вывести в порядке убывания все целые числа, расположенные между x и y, а также количество этих чисел

18 . Даны действительные числа x, y. Найти произведение всех целых чисел, расположенных между x и y, а также количество этих чисел.

19 Даны действительные числа x, y. Найти сумму квадратов всех целых чисел, расположенных между x и y, а также количество этих чисел.

20 Даны действительные числа x, y. Найти сумму кубов всех целых чисел, расположенных между x и y, а также количество этих чисел.

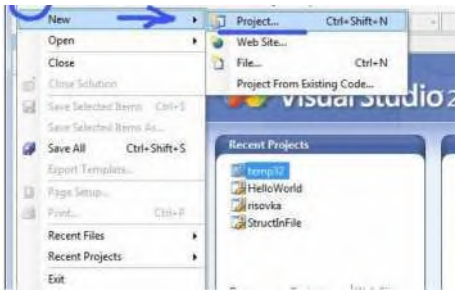
Практическое занятие № 3
Знакомство с инструментальной средой программирования

Цель работы: научиться создавать консольные приложения

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом.

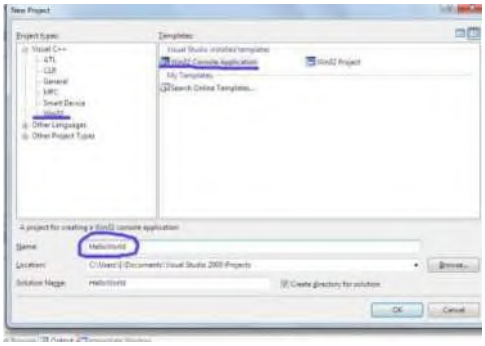
Создание проекта и добавление исходного файла состоит из следующих шагов.

1 Создание проекта, последовательно выбрав меню Файл пункты Создать_и_Проект (File-New-Project).



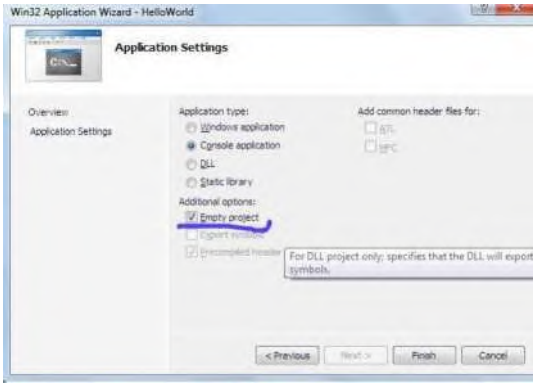
2 В области типов проектов Visual C++ выберите группу Win32 и щелкните элемент Консольное приложение (Win32 Console Application).

3 Введите имя проекта.

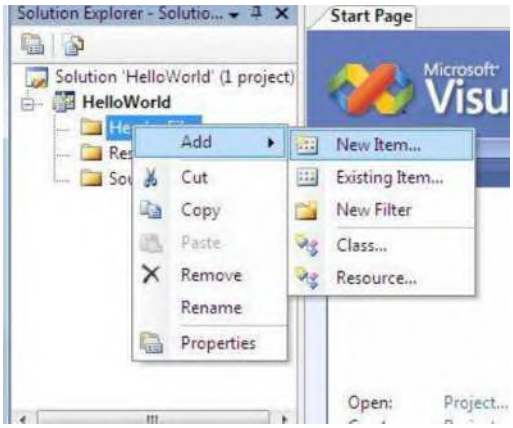


По умолчанию имя решения, содержащего проект, совпадает с именем проекта, однако можно ввести другое имя. Также можно указать другое расположение для проекта. Нажмите кнопку ОК, чтобы создать проект.

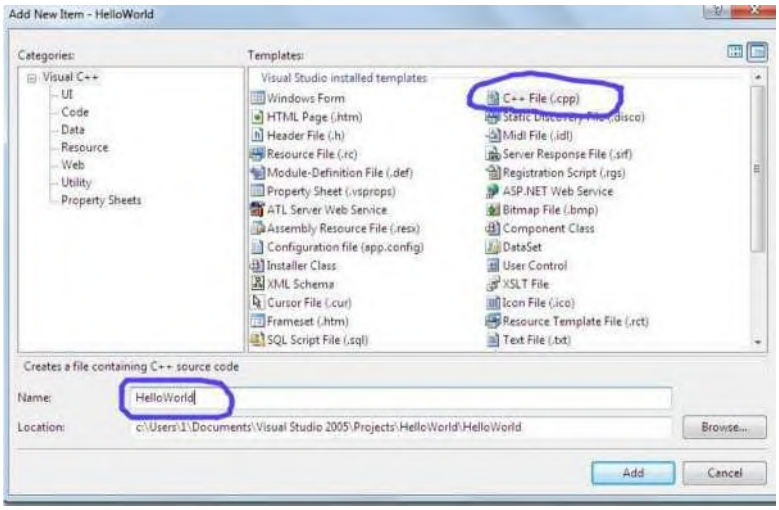
4 В мастере приложений Win32 нажмите кнопку Далее (Next), выберите вариант Пустой проект (EmptyProject)и нажмите кнопку Готово (Finish).



5 Далее появилось окно Solution Explorer, в котором должно быть название вашего проекта. Если у вас Solution Explorer не появляется, то скорее всего надо нажать View-Solution Explorer (это в основном меню). Под названием проекта отображены значки трех папок. Нас интересует Header Files. — жмем на ней правой клавишей мыши открывается контекстное меню, где надо выбрать New Item



6 Дальше появляется новое окно, в котором выбираем C++ File (.cpp), вводим название проекта и нажимаем Add.



Если все правильно, то можно приступать к написанию первой программы
Написав текст программы и можно проверить её нажав ctrl+F5.

Каждое задание необходимо решить в соответствии с изученными методами обработки данных и преобразования типов данных в языке C++.
Следует реализовать каждое задание в соответствии с приведенными этапами:

1. изучить словесную постановку задачи, выделив при этом все виды данных;
2. сформулировать математическую постановку задачи;
3. выбрать метод решения задачи, если это необходимо;
4. разработать графическую схему алгоритма;
5. записать разработанный алгоритм на языке C++;
6. разработать контрольный тест к программе;
7. отладить программу;
8. представить отчет по работе.

Практическое занятие № 4
Разработка программ последовательной структуры

Цель работы: изучить классификацию типов и их внутреннее представление в языке С++, научиться работать со стандартными функциями. Изучить структуру программы на языке С++. Ознакомиться с программированием математических формул.

Краткие теоретические сведения.

Базовые типы – это типы данных, которые предопределены *стандартом языка*, указываются зарезервированными ключевыми словами, характеризуются одним значением и внутренним представлением.

Вещественный тип – это *базовый тип данных*, который применяется для хранения дробных чисел в формате с плавающей точкой.

Логический (булевый) тип – это *базовый тип данных*, который применяется для хранения значений двузначной логики.

Неявное приведение типа – это преобразование значения переменной к новому типу, которое происходит автоматически, по правилам, заложенным в языке программирования.

Перечисляемый тип – это *производный тип данных*, он определяется как набор идентификаторов, являющихся именованными целыми константами, которым приписаны уникальные обозначения

Преобразование типов – это приведение значения переменной одного типа в *значение* другого типа.

Производные типы – это типы данных, которые задаются пользователем.

Символьный тип – это *базовый тип данных*, который применяется для хранения символов или *управляющих последовательностей* в виде кода.

Тип данных – это множество допустимых значений, которые может принимать тот или иной *объект*, а также множество допустимых операций, которые применимы к нему.

Типы класса – это типы данных, экземплярами которых являются объекты.

Целочисленный тип – это *базовый тип данных*, который применяется для хранения целых чисел.

Явное приведение типа – это преобразование значения переменной к новому типу, при котором указывается *тип переменной*, к которому необходимо привести исходную переменную.

1. Для организации хранения данных и корректного выполнения операций над ними в языках программирования определены типы данных.
2. Типы характеризуются схожим внутренним представлением данных в памяти компьютера; объемом памяти, выделяемым под данные; множеством (диапазоном) принимаемых значений; допустимыми операциями и функциями.
3. В языке С++ типы классифицируются на базовые, производные и классы.
4. Для базовых типов определены их подмножества и расширения, что обеспечивает повышение точности расчетов или экономный расход памяти.
5. Над типами данных определена операция преобразования типов. Ее следует применять с осторожностью при переходе к типу, у которого меньше по модулю границы диапазонов.

Варианты заданий

При выполнении лабораторной работы для каждого задания требуется написать программу на языке С++, которая получает на входе числовые данные, выполняет их обработку в соответствии с требованиями задания и выводит результат на экран. Ввод данных осуществляется с клавиатуры с учетом требований к входным данным, содержащихся в постановке задачи. Ограничениями на *входные данные* является допустимый*диапазон* значений используемых *числовых типов* в языке С++.

1. Найдите сумму первых трех цифр дробной части *вещественного числа*. Например, для числа 23,16809 она равна 15.

2. Составьте программу вычисления стоимости поездки на автомобиле на дачу (туда и обратно). Исходными данными являются: расстояние до дачи (в километрах); количество бензина, которое потребляет автомобиль на 100 км пробега; цена одного литра бензина. Ниже представлен рекомендуемый вид диалога во время работы программы.
   1. Вычисление стоимости поездки на дачу.
   2. Расстояние до дачи (км) – 67
   3. Расход бензина (л на 100 км) – 8.5
   4. Цена литра бензина (руб.) – 23.7
   5. Поездка на дачу обойдется в 269 руб. 94 коп.
3. Составьте *линейную программу*, печатающую значение 1, если указанное *высказывание* является истинным, и 0 – в противном случае. Величина d является корнем только одного из уравнений $ax^2 + bx + c = 0$ и $mx + n = 0$ относительно x.
4. Составьте программу, которая преобразует введенное с клавиатуры дробное число в денежный формат. Например, число 12,348 должно быть преобразовано к виду 12 руб. 35 коп. Ниже представлен рекомендуемый вид диалога во время работы программы. Данные, вводимые пользователем:
   1. Преобразование числа в денежный формат.
   2. Введите дробное число – 23,6
   3. 23.6 руб. – это 23 руб. 60 коп.

Указания к выполнению работы

Каждое задание необходимо решить в соответствии с изученными методами обработки данных и преобразования типов данных в языке C++.

Следует реализовать каждое задание в соответствии с приведенными этапами:

9. изучить словесную постановку задачи, выделив при этом все виды данных;
10. сформулировать математическую постановку задачи;
11. выбрать метод решения задачи, если это необходимо;
12. разработать графическую схему алгоритма;
13. записать разработанный алгоритм на языке C++;
14. разработать контрольный тест к программе;
15. отладить программу;
16. представить отчет по работе.

Требования к отчету

Отчет по лабораторной работе должен соответствовать следующей структуре.

1. Титульный лист.
2. Словесная постановка задачи. В этом подразделе проводится полное описание задачи. Описывается суть задачи, анализ входящих в нее физических величин, область их допустимых значений, единицы их измерения, возможные ограничения, анализ условий при которых задача имеет решение (не имеет решения), анализ ожидаемых результатов.
3. Математическая модель. В этом подразделе вводятся математические описания физических величин и математическое описание их взаимодействий. Цель подраздела – представить решаемую задачу в математической формулировке.
4. Алгоритм решения задачи. В подразделе описывается разработка структуры алгоритма, обосновывается абстракция данных, задача разбивается на подзадачи. Схема алгоритма выполняется по ЕСПД (ГОСТ 19.003-80 и ГОСТ 19.002-80).

5. Листинг программы. Подраздел должен содержать текст программы на языке программирования C++, реализованный в среде MS Visual Studio 2010.
6. Контрольный тест. Подраздел содержит наборы исходных данных и полученные в ходе выполнения программы результаты.
7. Выводы по лабораторной работе.
8. Ответы на контрольные вопросы.

Контрольные вопросы

1. Почему в языке C++ определена строгая типизация данных, используемых в программе?
2. Как определяются границы диапазона базового типа в зависимости от выделяемой под этот тип памяти?
3. С какой целью в C++ определен тип void?
4. Какой объем памяти выделяется под переменную типа void? Какие значения может принимать переменная типа void?
5. Почему наблюдается *асимметрия* значений границ диапазонов целочисленных типов?
6. Чему будет равно значение операции *инкремента* для максимального числа в целочисленном типе? А каков результат*декремента* для минимального значения в таком же типе?
7. Почему запись целых чисел нельзя начинать с незначащих нулей?
8. Каким образом представлено число ноль в вещественных типах?
9. Почему в C++ *символьный тип* считается подмножеством целочисленного типа?
10. Каким образом можно инициализировать переменную перечисляемого типа?
11. При преобразовании целого со знаком к целому без знака всегда ли будет получено исходное числовое значение? Ответ обоснуйте.

Практическое занятие № 5
Разработка программ разветвляющейся структуры

Цель работы: научиться писать приложения на языке с++, используя операторы ветвления

Оператор if

```
int main()
{
setlocale(LC_ALL, "Russian");
double num;
cout << "Введите произвольное число: ";
cin >> num;
if (num < 10) { // Если введенное число меньше 10.
cout << "Это число меньше 10." << endl;
} else { // иначе
cout << "Это число больше либо равно 10." << endl;
}
return 0;
}
```

Если вы запустите эту программу, то при вводе числа, меньшего десяти, будет выводиться соответствующее сообщение. Если введенное число окажется большим, либо равным десяти — отобразится другое сообщение.

Оператор if служит для того, чтобы выполнить какую-либо операцию в том случае, когда условие является верным. Условная конструкция в С++ всегда записывается в круглых скобках после оператора if. Внутри фигурных скобок указывается тело условия. Если условие выполнится, то начнется выполнение всех команд, которые находятся между фигурными скобками.

Оператор переключатель switch

Оператор выбора имеет следующий синтаксис

switch *выражение_целого_типа*

{

case значение_1: оператор1;break;

...

case значение_n: оператор_n;break;

default: оператор_d;

}

Выполняется оператор следующим образом.

Вычисляется выражение, записанное как условие.

Если значение условия равно значение_i , то выполняется оператор_i и далее - выход из оператора switch.

Если значение условия не равно ни одному значение_i , то выполняется оператор_d .

Пример 1.

```
int x=2;
switch x
{
case 1: cout<<1<<endl;break;
case 2: cout<<2<<endl;break;
default: cout<<”x not equal 1 or 2”<<endl;
}
```

Ввод-вывод в С++

Обмен данными между программой и внешними устройствами осуществляется с помощью операций ввода-вывода. Типичным внешним устройством является консоль.

Другим типичным устройством является жесткий диск, на котором расположены *файлы*. Программа может создавать *файлы*, в которых хранится информация. Другая (или эта же) программа может читать информацию из *файла*.

В языке С++ нет особых операторов для ввода или вывода данных. Вместо этого имеется набор классов, стандартно поставляемых вместе с компилятором, которые и реализуют основные операции ввода-вывода.

Причиной является как слишком большое разнообразие операций ввода и вывода в разных операционных системах, особенно графических, так и возможность определения новых типов данных в языке С++. Библиотека классов для ввода-вывода решает две задачи. Во-первых, она обеспечивает эффективный ввод-вывод всех встроенных типов и простое, но тем не менее гибкое, определение операций ввода-вывода для новых типов, разрабатываемых программистом. Во-вторых, сама библиотека позволяет при необходимости развивать её и модифицировать.

Потоки. Механизм для ввода-вывода в С++ называется ***потоком***. Название произошло от того,что информация вводится и выводится в виде ***потока*** байтов – символ за символом.

Класс istream реализует *поток* ввода, класс ostream – *поток* вывода. Эти классы определены в файле заголовков iostream.h. Библиотека *потоков* ввода-вывода определяет три глобальных объекта: cout, cin и cerr. cout называется стандартным выводом, cin – стандартным вводом, cerr – стандартным *потоком* сообщений об ошибках. cout и cerr выводят на консоль и принадлежат к классу ostream, cin имеет тип istream и вводит с консоли.

Вывод осуществляется с помощью операции <<, ввод с помощью операции >>. Выражение

```
cout << "Пример вывода: " << 34;
```

напечатает на строку "Пример вывода", за которым будет выведено число 34. Выражение

```
int x;
cin >> x;
```

введет целое число в переменную x.

Манипуляторы и форматирование ввода-вывода. Часто бывает необходимо вывести строку или число в определенном формате. Для этого используются так называемые ***манипуляторы***.

Манипуляторы – это объекты особых типов, которые управляют тем, как ostream или istream обрабатывают последующие аргументы. Некоторые *манипуляторы* могут также выводить или вводить специальные символы.

endl	при выводе перейти на новую строку;
dec	выводить числа в десятичной системе (действует по умолчанию);
oct	выводить числа в восьмеричной системе;
hex	выводить числа в шестнадцатиричной системе счисления;
setw (int n)	установить ширину поля вывода в n символов (n – целое число);

Использовать *манипуляторы* просто – их надо вывести в выходной *поток*. Предположим, мы хотим вывести одно и то же число в разных системах счисления:

```
int x = 53;
cout << "Десятичный вид: " << dec
 << x << endl
 << "Восьмиричный вид: " << oct
 << x << endl
 << "Шестнадцатиричный вид: " << hex
 << x << endl
```

Аналогично используются *манипуляторы* с параметрами. Вывод числа с разным количеством цифр после запятой:

```
double x;
// вывести число в поле общей шириной
// 6 символов (3 цифры до запятой,
// десятичная точка и 2 цифры после запятой)
cout << setw(6) << setprecision(2) << fixed << x << endl;
```

Те же *манипуляторы* (за исключением endl) могут использоваться и при вводе. В этом случае они описывают представление вводимых чисел.

```
int x;
// ввести шестнадцатиричное число
cin >> hex >> x;
```

Поля вывода. Функция **width()** устанавливает минимальное число символов, использующееся в последующей операции вывода числа или строки. Так в результате следующих операций

```
cout.width(4);
cout << '(' << 12 << ')'; //получим число 12 в поле размером 4 символа.
```

Функция width() задает минимальное число символов. Если появится больше символов, они будут напечатаны все, поэтому

```
cout.width(4);
cout << '(' << "121212" << ")\n";
```

напечатает (121212)

Причина, по которой разрешено переполнение поля, а не усечение вывода,в том, чтобы избежать зависания при выводе.

3. Порядок выполнения работы

1. Ознакомиться с теоретическими сведениями.
2. Получить вариант задания у преподавателя.
3. Создание проекта. В строке меню выберите Файл> Создать> Проект. В левой области диалогового окна Новый проект последовательно разверните узлы Установленные шаблоны и Visual C++, а затем щелкните Win32. В списке установленных шаблонов в центральной области выберите шаблон Консольное приложение Win32. В поле Имя введите имя проекта. Вы можете принять расположение по умолчанию в раскрывающемся списке Расположение, ввести другое расположение или нажать кнопку Обзор, чтобы перейти к каталогу, в котором требуется сохранить проект. Во время создания проекта среда Visual Studio помещает проект в решение. По умолчанию имя решения совпадает с именем проекта. Можно изменить имя в поле *Имя решения*, но для этого примера оставьте имя по умолчанию. На странице *Обзор* окна *Мастер приложений Win32* нажмите кнопку *Далее*. На странице *Параметры приложения* выберите в поле *Тип приложения* пункт *Консольное приложение*. Нажмите кнопку *Готово*, чтобы создать проект. Добавьте код:

```
#include "stdafx.h"
#include <iostream>
#include <cmath> // Работа с математическими функциями
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
  setlocale(LC_ALL, "Russian");
  .
  .
```

```
.
cin.get();
return 0;}
```

4. Выполнить задание по варианту.
5. Продемонстрировать выполнение работы преподавателю.
6. Оформить отчет.

Контрольные вопросы

1. Что такое решение (solution) в Visual Studio .NET? Зачем они нужны?
2. Как создать решение? Как добавить туда проект?
3. Что такое список задач? Как можно добавить задачу?
4. Зачем нужно окно свойств?
5. Зачем нужно окно вывода и окно ошибок? Чем они отличаются?
6. Зачем нужно окно решения?
7. Как добавить в проект новый или существующий файл.
8. Какой синтаксис имеет цикл while?.
9. Какой синтаксис имеет цикл do-while?.
10. Какой синтаксис имеет цикл for?
11. Для чего используется оператор break?
12. Каково назначение оператора continue?

Варианты заданий.

1. Даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень — отрицательные.

2. Даны две точки $A(x_1, y_1)$ и $B(x_2,y_2)$. Составить алгоритм, определяющий, которая из точек находится ближе к началу координат.

3. Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник, и если да, то будет ли он прямоугольным.

4. Даны действительные числа x и y, не равные друг другу. Меньшее из этих двух чисел заменить половиной их суммы, а большее — их удвоенным произведением.

5. На плоскости XOY задана своими координатами точка A. Указать, где она расположена (на какой оси или в каком координатном угле).

6. Даны целые числа m, n. Если числа не равны, то заменить каждое из них одним и тем же числом, равным большему из исходных, а если равны, то заменить числа нулями.

7. Подсчитать количество отрицательных среди чисел a, b, c.

8. Подсчитать количество положительных среди чисел a, b, c.

9. Услуги телефонной сети оплачиваются по следующему правилу: за разговоры до A минут в месяц — B руб., а разговоры сверх установленной нормы оплачиваются из расчета C руб. за минуту. Написать программу, вычисляющую плату за пользование телефоном для введенного времени разговоров за месяц.

13. Грузовой автомобиль выехал из одного города в другой со скоростью v_1 км/ч. Через t ч в этом же направлении выехал легковой автомобиль со скоростью v_2 км/ч. Составить программу, определяющую, догонит ли легковой автомобиль грузовой через t_1 ч после своего выезда.

14. Перераспределить значения переменных x и y так, чтобы в x оказалось большее из этих значений, а в y - меньшее.

15. Определить правильность даты, введенной с клавиатуры (число — от 1 до 31, месяц — от 1 до 12). Если введены некорректные данные, то сообщить об этом.

16. Составить программу, определяющую результат гадания на ромашке — «любит—не любит», взяв за исходное данное количество лепестков n.

17. Написать программу — модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожароопасная ситуация», если температура в комнате превысила 60° С.

18. Рис расфасован в два пакета. Масса первого — n кг, второго — m кг. Составить программу, определяющую:

а) какой пакет тяжелее — первый или второй;

б) массу более тяжелого пакета.

19. Написать программу, которая анализирует данные о возрасте и относит человека к одной из четырех групп: дошкольник, ученик, работник, пенсионер. Возраст вводится с клавиатуры.

20. Составить программу, определяющую, пройдет ли график функции $y = ax^2 + bx + c$ через заданную точку с координатами (m, n).

21. К финалу конкурса лучшего по профессии «Специалист электронного офиса» были допущены трое: Иванов, Петров, Сидоров. Соревнования проходили в три тура. Иванов в первом туре набрал m_1 баллов, во втором — n_1, в третьем — p_1. Петров — m_2, n_2, p_2 соответственно; Сидоров — m_3, n_3 , p_3. Составить программу, определяющую, сколько баллов набрал победитель.

Практические занятия № 6,7,8
Разработка программ с использованием цикла с предусловием.
Разработка программ с использованием цикла с постусловием.
Разработка программ с использованием цикла с параметром

Цель работы: научиться писать приложения на языке с++, используя операторы цикла

Оператор цикла while

Для организации повторения, управляемого счетчиком, требуется задать:

– Имя управляющей переменной (или счетчика циклов).

– Приращение (или уменьшение), на которое изменяется управляющая переменная в каждом цикле.

– Условие проверки, не достигнуто ли конечное значение управляющей переменной (т.е. надо ли продолжать циклы).

Оператор цикла while организует цикл с предусловием, т.е. проверка условия окончания цикла происходит в начале цикла.

Рассмотрим простую программу, которая выводит на консоль числа от 1 до 10:

```
// повторение, управляемое счетчиком
#include <iostream>
main() {
	int counter = 1; // задание начального значения
	while(counter <= 10) { // условие повторения
		count << counter << endl;
		++counter; // увеличение
	}
return 0;
}
```

Вывод.

1 2 3 4 5 6 7 8 9 10

Объявление int counter = 1; определяет имя управляющей переменной (counter), объявляет переменную как целую, резервируя для нее соответствующее место в памяти, и задает ее начальное значение 1. Объявления, в которых задается начальное значение, являются фактически выполняемыми операторами. Это объявление и задание начального значения переменной counter можно было бы сделать операторами int counter; counter = 1;

Здесь само объявление не является выполняемым оператором, а присваивание осуществляется отдельным оператором. Можно использовать оба эти метода для задания начальных значений переменных. Оператор ++counter увеличивает счетчик на 1 при каждом выполнении цикла. Условие продолжения цикла в структуре while проверяет, меньше или равно значение управляющей переменной числа 10 (последнего значения, при котором условие истинно). Заметьте, что тело этой структуры while выполняется и при значении управляющей переменной, равном 10. Выполнение цикла заканчивается, когда значение управляющей переменной превысит 10 (т.е. переменная counter станет равной 11).

Программа может быть сделана более компактной, если переменной counter задать начальное значение 0 и заменить структуру while следующей:

```
while(++counter <= 10)
	cout << counter << endl;
```

Этот код экономит один оператор, поскольку инкремент выполняется непосредственно в условии структуры while до того, как это условие проверяется. Этот код также устраняет скобки, заключающие в себе тело while, поскольку теперь while содержит всего один оператор.

Оператор цикла do/while

Структура повторения do/while похожа на структуру while. В структуре while условие продолжения циклов проверяется в начале цикла, до того, как выполняется тело цикла. В структуре do/while проверка условия продолжения циклов производится после того, как тело цикла выполнено, следовательно, тело цикла будет выполнено по крайней мере один раз. Когда do/while завершается, выполнение программы продолжается с оператора, следующего за предложением while.

Таки образом оператор цикла do/while организует цикл с постусловием.

Отметим, что в структуре do/while нет необходимости использовать фигурные скобки, если тело состоит только из одного оператора. Но фигурные скобки обычно все же ставят, чтобы избежать путаницы между структурами while и do/while. Например,

while (условие)

обычно рассматривается как заголовок структуры while. Структура do/while без фигурных скобок и с единственным оператором в теле имеет вид

do

оператор while(условие);

что может привести к путанице. Последняя строка — while (условие); может ошибочно интерпретироваться как заголовок структуры while, содержащий пустой оператор. Таким образом, чтобы избежать путаницы, структура do/while даже с одним оператором часто записывается в виде

do {

оператор } while (условие);

Некоторые программисты всегда включают фигурные скобки в структуру do/while, даже если в них нет необходимости. Это помогает устранить двусмысленность, проистекающую из совпадения предложений структуры while и структуры do/while, содержащей один оператор.

Следующая программа использует структуру do/while, чтобы напечатать числа от 1 до 10. Обратите внимание, что к управляющей переменной counter в проверке окончания цикла применяется инкремент в префиксной форме. Отметьте также использование фигурных скобок, заключающих единственный оператор в теле do/while.

```
// Применение структуры повторения do/while
int main() {
int counter = 1;
do {
cout << counter << " ";
} while(++counter <= 10);
return 0;
}
```

Оператор цикла for

Структура повторения for содержит все элементы, необходимые для повторения, управляемого счетчиком, т.е. параметром. Чтобы проиллюстрировать мощь структуры for, перепишем программу.

```
// Повторение, управляемое счетчиком, со структурой for
int main() {
        // задание начального значения,условие повторения и
        // приращение – включено в заголовок структуры for
        for(int counter = 1; counter <= 10; counter++)
                cout << counter << endl;
        return 0;
```

}

Когда структура for начинает выполняться, управляющей переменной counter задается начальное значение 1. Затем проверяется условие продолжения цикла counter <= 10. Поскольку начальное значение counter равно 1, это условие удовлетворяется, так что оператор тела структуры печатает значение counter, равное 1. Затем управляющая переменная counter увеличивается на единицу в выражении counter++ и цикл опять начинается с проверки условия его продолжения. Поскольку значение counter теперь 2, предельная величина не превышена, так что программа снова выполняет тело цикла. Этот процесс продолжается, пока управляющая переменная counter не увеличится до 11 — это приведет к тому, что условие продолжения цикла нарушится и повторение прекратится. Выполнение программы продолжится с первого оператора, расположенного после структуры for (в данном случае с оператора return в конце программы).

Общая форма структуры for: for (выражение1; выражение2; выражение3) оператор, где выражение1 задает начальное значение переменной, управляющей циклом, выражение2 является условием продолжения цикла, а выражение3 изменяет управляющую переменную. В большинстве случаев структуру for можно представить эквивалентной ей структурой while следующим образом:

```
Выражение1;
while(Выражение2) {
        оператор
        Выражение3;
        }
```

«Приращение» структуры for может быть отрицательным (в этом случае в действительности происходит не приращение, а уменьшение переменной, управляющей циклом). Если условие продолжения цикла с самого начала не удовлетворяется, то операторы тела структуры for не выполняются и управление передается оператору, следующему за for. Управляющая переменная иногда печатается или используется в вычислениях в теле структуры for, но обычно это делается без изменений ее величины. Чаще управляющая переменная используется только для контроля числа повторений и никогда не упоминается в теле структуры. Хотя управляющая переменная может изменяться в теле цикла for, избегайте делать это, так как такая практика приводит к неявным, неочевидным ошибкам.

Операторы break и continue

Операторы break и continue изменяют поток управления. Когда оператор break выполняется в структурах while, for, do/while или switch, происходит немедленный выход из структуры. Программа продолжает выполнение с первого оператора после структуры. Обычное назначение оператора break — досрочно прерывать цикл или пропустить оставшуюся часть структуры switch.

Оператор continue в структурах while, for или do/while вызывает пропуск оставшейся части тела структуры и начинается выполнение следующей итерации цикла. В структурах while и do/while немедленно после выполнения оператора continue производится проверка условия продолжения цикла. В структуре for выполняется выражение приращения, а затем осуществляется проверка условия продолжения.

Оператор переключатель switch

Оператор выбора имеет следующий синтаксис

```
switch выражение_целого_типа
{
case значение_1: оператор1;break;
...
case значение_n: оператор_n;break;
default: оператор_d;
```

}

Выполняется оператор следующим образом.

Вычисляется выражение, записанное как условие.

Если значение условия равно значение_i , то выполняется оператор_i и далее - выход из оператора switch.

Если значение условия не равно ни одному значение_i , то выполняется оператор_d .

Пример 1.

```
int x=2;
switch x
{
case 1: cout<<1<<endl;break;
case 2: cout<<2<<endl;break;
default: cout<<"x not equal 1 or 2"<<endl;
}
```

Варианты заданий

1. Имеется серия измерений элементов треугольника. Группы элементов пронумерованы. В серии в произвольном порядке могут встречаться такие группы элементов треугольника:

1) основание и высота;

2) две стороны и угол между ними (угол задан в радианах);

3) три стороны.

Разработать программу, которая запрашивает номер группы элементов, вводит соответствующие элементы и вычисляет площадь треугольника. Вычисления прекратить, если в качестве номера группы введен 0.

2. Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?

3. Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, сколько амеб будет через 3, 6, 9, 12,..., 24 часа.

4. Около стены наклонно стоит палка длиной *x* м. Один ее конец находится на расстоянии *y* м от стены. Определить значение угла *a* между палкой и полом для значений *x*=к м и *y*, изменяющегося от 2 до 3 м с шагом h м.

5. У гусей и кроликов вместе 64 лапы. Сколько может быть кроликов и гусей (указать все сочетания)?

6. Сколько можно купить быков, коров и телят, платя за быка 10 т. руб., за корову — 5 т. руб., а за теленка — 0,5 т.руб., если на 100 т.руб. надо купить 100 голов скота?

8. Составить программу для проверки утверждения: «Результатами вычислений по формуле x^2 +x+ 17 при 0<=x< =15 являются простые числа». Все результаты вывести на экран.

9. Составить программу для проверки утверждения: «Результатами вычислений по формуле x^2 + x + 41 при 0<x<=40 являются простые числа». Все результаты вывести на экран.

10. Составить программу-генератор простых чисел, в основу положить формулу $2x^2$ + 29 при 0 <= x <= 28.

12. Составить программу-генератор чисел Пифагора a, b, c ($c^2=a^2+b^2$). В основу положить формулы: a = m^2 - n^2, b = 2mn, c = m^2+ n^2 (m, n - натуральные, 1 < m< к, 1<n< к, к — данное число). Результат вывести на экран в виде таблицы из пяти столбцов: m, n, a, b, c.

13. Покупатель должен заплатить в кассу 5 руб. У него имеются купюры по 1, 5, 10, 50, 100, 500, 1000 и 10000 руб. Сколько купюр разного достоинства отдаст покупатель, если он начинает платить с самых крупных купюр?

14. Ежемесячная стипендия студента составляет А руб., а расходы на проживание превышают стипендию и составляют В руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3%. Составьте программу расчета суммы денег, которую необходимо единовременно попросить у родителей, чтобы можно было прожить учебный год (10 месяцев), используя только эти деньги и стипендию.

15. Составить программу, которая печатает таблицу умножения и сложения натуральных чисел в десятичной системе счисления.

16. Составить программу, которая печатает таблицу умножения и сложения натуральных чисел в шестнадцатеричной системе счисления.

17. Найти сумму всех n-значных чисел (1 <= n < =4).

18. Найти сумму всех n-значных чисел, кратных к (1 <= n <= 4).

19. Заменить буквы цифрами так, чтобы соотношение оказалось верным (одинаковым буквам соответствуют одинаковые цифры, разным — разные):

ХРУСТ*ГРОХОТ = РРРРРРРРРРР.

20. Составить программу, которая запрашивает пароль (например, четырехзначное число) до тех пор, пока он не будет правильно введен.

21. Составить программу, которая находит наибольшее значение отношения трехзначного числа к сумме его цифр.

22. Составить алгоритм решения ребуса РАДАР = $(Р + А + Д)^4$ (различные буквы обозначают различные цифры, старшая — не 0).

23. Составить алгоритм решения ребуса МУХА + МУХА + МУХА = СЛОН (различные буквы обозначают различные цифры, старшая — не 0).

24. Составить алгоритм решения ребуса ДРУГ — ГУРД = 2727 (различные буквы обозначают различные цифры, старшая — не 0).

25. Составить алгоритм решения ребуса КОТ + КОТ = ТОК (различные буквы обозначают различные цифры, старшая — не 0).

Практическое занятие № 9
Разработка программ с использованием одномерных массивов

Цель работы: освоение средств языка С++ для изучения способов описания, ввода-вывода и обработки одномерных массивов.

Краткие теоретические сведения

Одномерный массив — массив, с одним параметром, характеризующим количество элементов одномерного массива. Фактически одномерный массив — это массив, у которого может быть только одна строка, и n-е количество столбцов. Столбцы в одномерном массиве — это элементы массива. На рисунке 1 показана структура целочисленного одномерного массива *a*. Размер этого массива — 16 ячеек.

5	-12	-12	9	10	0	-9	-12	-1	23	65	64	11	43	39	-15
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]	a[14]	a[15]

Рисунок 1— Одномерный массив

Максимальный индекс одномерного массива *a* равен 15, но размер массива 16 ячеек, потому что нумерация ячеек массива всегда начинается с 0. Индекс ячейки - это целое неотрицательное число, по которому можно обращаться к каждой ячейке массива и выполнять какие-либо действия над ней (ячейкой). Синтаксис объявления одномерного массива в С++:

тип данных имя одномерного массива [размерность одномерного массива];
//пример объявления одномерного массива, изображенного на рисунке 1:

```
int a[16];
```

где, int — целочисленный тип данных; а — имя одномерного массива; 16 — размер одномерного массива, 16 ячеек.

Всегда сразу после имени массива идут квадратные скобочки, в которых задаётся размер одномерного массива, этим массив и отличается от всех остальных переменных.

Рассмотрим на примерах способы ввода-вывода одномерного массива:

1) размер массива можно не указывать только при его инициализации, при обычном объявлении массива обязательно нужно указывать размер массива

```
#include "stdafx.h"
#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{
   cout << "obrabotka massiva" << endl;
// объявление и инициализация одномерного массива
   int array1[16] = { 5, -12, -12, 9, 10, 0, -9, -12, -1, 23, 65, 64, 11, 43, 39, -
   15 };
   cout << "indeks" << "\t\t" << "element massiva" << endl; // печать
   заголовков
    for (int counter = 0; counter < 16; counter++) //начало цикла
   {
   //вывод на экран индекса ячейки массива, а затем содержимого этой
   ячейки cout << "array1[" << counter << "]" << "\t\t" << array1[counter]
   << endl;
```

```
    }
    system("pause");
    return 0;
}
```

Результат работы представлен на рисунке 2.

```
obrabotka massiva indeks                element massiva
array1[0]              5
array1f1]              -12
array1[2]              -12
array1[3]              9
array1[4]              10
array1[5]              0
array1[6]              -9
array1[7]              -12
array1[8]              -1
array1[9]              23
array1[10]             65
array1[11]             64
array1[12]             11
array1[13]             43
array1[14]             39
array1[15]             -15
Для продолжения нажмите любую клавишу . . .
```

Рисунок 2

2) программа должна последовательно считывать десять введённых чисел с клавиатуры. Все введённые числа просуммировать, результат вывести на экран. #include "stdafx.h"

```
#include <iostream>
using namespace std;
int main(int argc, char* argv[])
{
    int array1[10]; // объявляем целочисленный массив
    cout << "Enter elementi massiva: " << endl;
    int sum = 0;
    for ( int counter = 0; counter < 10; counter++ ) // цикл для считывания чисел
        cin >> array1[counter]; // считываем вводимые с клавиатуры числа
    cout << "array1 = {";
for ( int counter = 0; counter < 10; counter++ ) // цикл для вывода элементов массива cout <<
array1[counter] << " "; // выводим элементы массива на стандартное
устройство вывода
    for ( int counter = 0; counter < 10; counter++ ) // цикл для суммирования чисел
массива
        sum += array1[counter]; // суммируем элементы массива
    cout << "}\nsum = " << sum << endl;
    system("pause");
    return 0;
}
```

Результат представлен на рисунке 3.

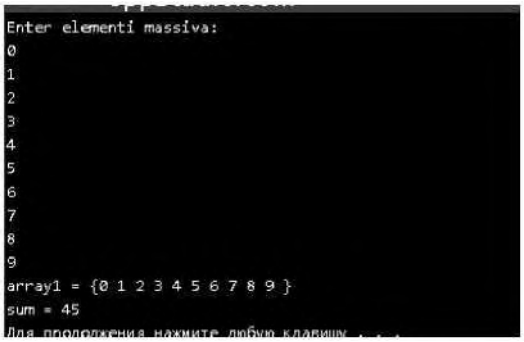


Рисунок 3.

Пример выполнения задания.

Задание: произвести следующую обработку 15 целых чисел: подсчитать сумму чисел, входящих в диапазон [-5..5] и количество нечетных чисел.

Схема программы к данному заданию с использованием одного цикла представлена на рисунке 4.

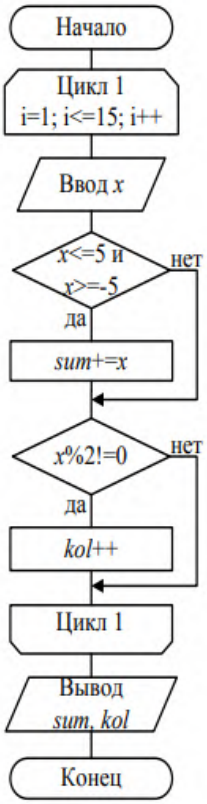


Рисунок 4

Текст программы без использования массива:

```
#include <iostream>

void main()
{
int x,sum=0,i,kol=0;
printf("Enter numbers\n");
for (i=1;i<=15;i++)
{
scanf("%d",&x);
if ((x>=-5)&&(x<=5)) sum+=x;
```

```
if (x%2!=0) kol++;
}
printf("Summa v diapazone [-5,5]=%d\n", sum);
printf("Kolichestvo nechetnih=%d", kol);
}
```

Схема программы с циклами для ввода и обработки массива (рисунок 5).

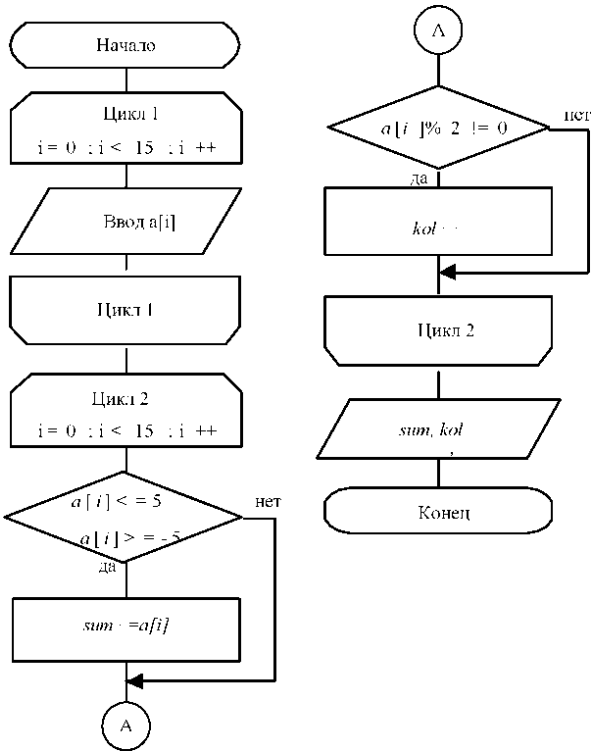


Рисунок 5

Пример программы с использованием одномерного массива

```
#include<iostream>
void main()
{
int a[15],s("E um=0,i,kol=0;
printf("Enter numbers\n");
for (i=0;i<15;i++)
scanf("%d",&a[i]);
for (i=0;i<15;i++)
{
if ((a[i]>=-5)&&(a[i]<=5)) sum+=a[i];
if (a[i]%2!=0) kol++;
}
printf("Summa v diapazone [-5,5]=%d\n", sum);
printf("Kolichestvo nechetnih=%d", kol);
}
```

Контрольные вопросы

1. Массивы в языке С++: понятие массива в языке С++, описание массива в программе, представление элементов массива в памяти, обращение к элементам массива.

Варианты заданий

1. Произвести следующую обработку 15 целых чисел: найти количество отрицательных чисел, количество нулевых и подсчитать сумму положительных чисел.
2. Произвести следующую обработку 15 целых чисел: найти количество четных чисел, а нечетные числа, входящие в диапазон [1..11] возвести в квадрат.
3. Произвести следующую обработку 15 вещественных чисел: найти количество отрицательных чисел, а числа, входящие в диапазон [0..10] возвести в квадрат.
4. Произвести следующую обработку 10 вещественных чисел: найти количество чисел, больших или равных 1,5, и подсчитать сумму отрицательных чисел.
5. Произвести следующую обработку 10 вещественных чисел: найти количество чисел, равных нулю, и найти сумму чисел, входящих в диапазон [15..15].
6. Произвести следующую обработку 15 целых чисел: найти количество отрицательных чисел и подсчитать разность положительных чисел.
7. Произвести следующую обработку 15 вещественных чисел: найти среднее арифметическое положительных чисел и подсчитать количество чисел, входящих в диапазон [-15..5].
8. Произвести следующую обработку 10 целых чисел: найти количество отрицательных чисел и подсчитать сумму положительных чисел, делящихся без остатка на 3.
9. Произвести следующую обработку 10 целых чисел: найти количество отрицательных чисел, а числа, входящие в диапазон [0..10], умножить на 10.
10. Произвести следующую обработку 10 целых чисел: найти количество отрицательных чисел, а числа, входящие в диапазон [0..10], умножить на 3.
11. Произвести следующую обработку 15 вещественных чисел: найти среднее арифметическое отрицательных чисел и подсчитать количество чисел, входящих в диапазон [0..5].
12. Произвести следующую обработку 15 вещественных чисел: найти среднее арифметическое нечетных чисел и подсчитать сумму чисел, входящих в диапазон [15..5].
13. Произвести следующую обработку 10 вещественных чисел: найти количество чисел, равных нулю, и найти синус чисел, входящих в диапазон [-15..15].
14. Произвести следующую обработку 10 целых чисел: подсчитать сумму положительных чисел и определить номера отрицательных чисел.
15. Произвести следующую обработку 15 вещественных чисел: найти количество отрицательных чисел и номера нулевых чисел.
16. Произвести следующую обработку 12 целых чисел: подсчитать количество чисел, делящихся без остатка на 5, и сумму чисел, входящих в диапазон [-5..5].
17. Произвести следующую обработку 10 вещественных чисел: подсчитать количество чисел, отличающихся от числа 3 не более чем на 0.5, и сумму отрицательных чисел.
18. Произвести следующую обработку 15 целых чисел: подсчитать количество нулевых чисел и вычислить квадраты чисел, входящих в диапазон [-5..5].
19. Произвести следующую обработку 12 целых чисел: подсчитать количество нечетных чисел и сумму отрицательных чисел.
20. Произвести следующую обработку 15 вещественных чисел: подсчитать количество чисел, отличающихся от заданного не более чем на 0.5, и сумму положительных чисел.
21. . Произвести следующую обработку 10 вещественных чисел: найти количество отрицательных чисел, находящихся в диапазоне от -5 до 5 и подсчитать сумму положительных чисел.
22. Произвести следующую обработку 15 целых чисел: найти количество чисел,

входящих в диапазон [1..11] и каждое число возвести в квадрат.

23. Произвести следующую обработку 15 вещественных чисел: найти количество чисел, равных 0, а числа, входящие в диапазон [-1..1] возвести в куб.

24. Произвести следующую обработку 10 вещественных чисел: найти количество чисел, больших или равных 1,5, и подсчитать сумму отрицательных чисел, входящих в диапазон [-1..0].

25. Произвести следующую обработку 10 вещественных чисел: найти количество чисел, меньших 15, и найти произведение чисел, входящих в диапазон [10..15].

Практическое занятие № 10

Разработка программ с использованием двумерных массивов. Указатели

Цель работы: изучение способов описания, ввода-вывода и обработки двумерных массивов, использование указателей при работе с массивами.

Краткие теоретические сведения

Двумерный массив — это обычная таблица, со строками и столбцами. Фактически двумерный массив — это одномерный массив одномерных массивов. Структура двумерного массива, с именем *a*, размером *m* на *n* показана ниже (рисунок 1).

a[0][0]	a[0][1]	a[0][2]	a[0][3]	. . .	a[0][n]
a[1][0]	a[1][1]	a[1][2]	a[1][3]	. . .	a[1][n]
a[2][0]	a[2][1]	a[2][2]	a[2][3]	. . .	a[2][n]
. . .	. . .	. . .	. . .	. . .	. . .
a[m][0]	a[m][1]	a[m][2]	a[m][3]	. . .	a[m][n]

Рисунок 1 — Двумерный массивы в С++

где, *m* — количество строк двумерного массива; *n* — количество столбцов двумерного массива;

m x *n* — количество элементов массива.

Наиболее распространенный синтаксис объявления двумерного массива:

/*тип данных*/ /*имя массива*/[/*количество строк*/][/*количество столбцов*/];

Примеры инициализации двумерного массива:

— способ 1:

int arr[5][3];

— способ 2:

int arr[5][3] = { {4, 7, 8}, {9, 66, -1}, {5, -5, 0}, {3, -3, 30}, {1, 1, 1} };

В последнем случае представление массива и обращение к элементу массива имеет вид, показанный на рисунке 2,3.

4 a[0][0]	7 [0][1]	8 [0][2]
9 a[1][0]	66 [1][1]	-1 [1][2]
5 a[2][0]	-5 [2][1]	0 [2][2]
3 a[3][0]	-3 [3][1]	30 [3][2]
1 a[4][0]	1 [4][1]	1 [4][2]

Рисунок 2 — Двумерный массив в С++

Рисунок 3 — Обращение в элементу массива

Ввод массива с клавиатуры и его вывод на экран выполняется следующим образом:

```
int m,n,ij;
 cout<<"Введите размеры матрицы:"<<endl;
 cin>>m;
 cin>>n;
 int elem;
 int ar [m][n];
  for (i=0; i<m; i++)
 {for (j=0; j<n; j++){
  cout<<«"Введите элемент:"«endl;
```

```
  cin>>elem;
   ar [i][j]=elem;
   }
 }
 for (i=0; i<m; i++)
 {for (j=0; j<n; j++){
  cout<<ar[i][j]«endl;
  }
 }
 system("pause");
 return 0;
 }
```

При работе с массивами можно использовать указатели. Указатель - переменная, значением которой является адрес ячейки памяти. То есть указатель ссылается на блок данных из области памяти, причём на самое его начало. Указатель может ссылаться на переменную или функцию. Для этого нужно знать адрес переменной или функции. Так вот, чтобы узнать адрес конкретной переменной в С++ существует унарная операция взятия адреса &. Такая операция извлекает адрес объявленных переменных, для того, чтобы его присвоить указателю.

Указатели используются для передачи по ссылке данных, что намного ускоряет процесс обработки этих данных (в том случае, если объём данных большой), так как их не надо копировать, как при передаче по значению, то есть, используя имя переменной. В основном указатели используются для организации динамического распределения памяти, например при объявлении массива, не надо будет его ограничивать в размере. Ведь программист заранее не может знать, какого размера нужен массив тому или иному пользователю, в таком случае используется динамическое выделение памяти под массив. Любой указатель необходимо объявить перед использованием, как и любую переменную:

/*тип данных*/ * /*имя указателя*/;

Работу с указателями можно представить следующим образом:

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int main(int argc, char* argv[])
{
   int var = 123; // инициализация переменной var числом 123
   int *ptrvar = &var; // указатель на переменную var (присвоили адрес переменной
указателю)
   cout << "&var = " << &var << endl;// адрес переменной var содержащийся в памяти,
извлечённый операцией взятия адреса
   cout << "ptrvar = " << ptrvar << endl;// адрес переменной var, является значением
указателя ptrvar
   cout << "var = " << var << endl; // значение в переменной var
   cout << "*ptrvar = " << *ptrvar << endl; // вывод значения содержащегося в переменной var
через указатель, операцией разименования указателя
   system("pause");
   return 0;
}
```

Результат работы представлен ниже на рисунке 4.

```
&var    = 0x22ff08
ptrvar  = 0x22ff08
var     = 123
*ptrvar = 123
Для продолжения нажмите любую клавишу . . .
```

Рисунок 4

Пример выполнения задания.

Задание. Найти максимальную сумму элементов строк матрицы 3x5. Написать программы без использования указателей и с использованием указателей.

Схема программы без использования указателей представлена на рисунке 5:

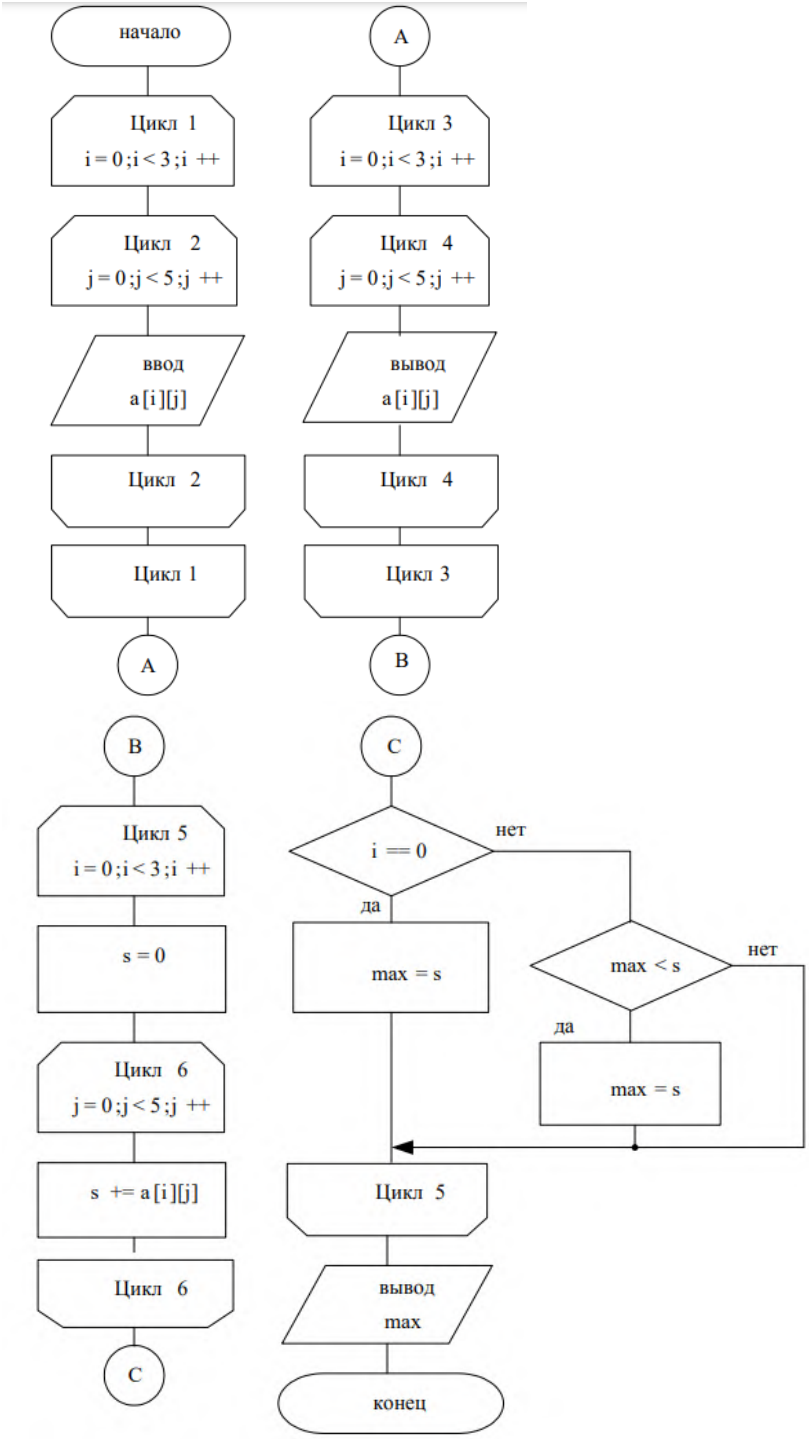
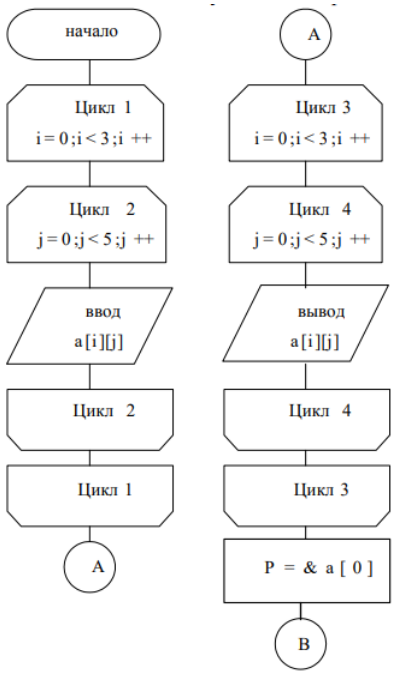


Рисунок 5

Текст программы:

```
#include <stdio.h>
void main()
{
int a[3][5], i, j, s, max;
printf (“Введите 3 строки по 5 чисел”);
for (i=0;i<3;i++)
  for (j=0;j<5;j++)
  scanf("%d",&a[i][j]);
   printf (“Матрица a :\n”);
      for (i=0; i<3; i++)
       {for (j=0; j<5; j++)
        printf (“%5d”, a[i][j]);
      printf (“\n”);
  }
for(i=0;i<3;i++)
  {s=0;
      for (j=0;j<5;j++)
       s+=a[i][j];
    if (i==0) max=s;
    else if (max<s) max=s;}
printf ("Максимальная сумма строки = %d",max);
}
```

Схема программы с использованием указателей представлена на рисунке 6



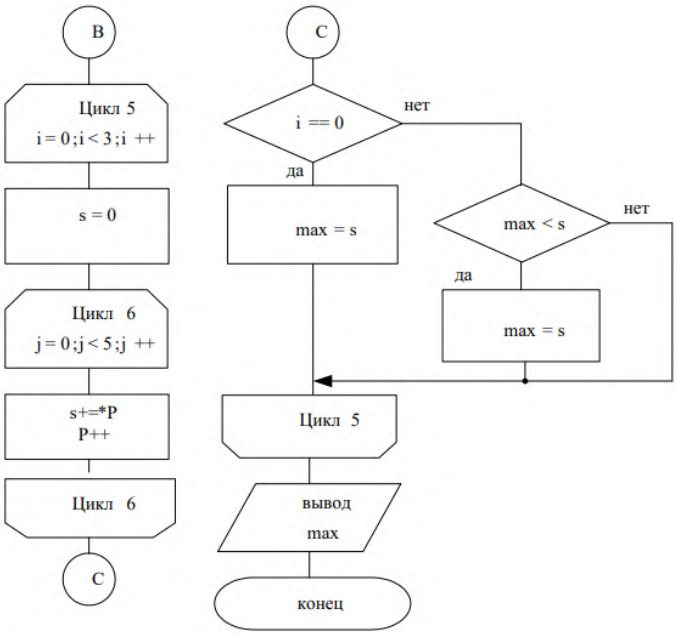


Рисунок 6

Пример программы с использованием указателей:

```
#include <stdio.h>
void main()
{
int a[3][5], *P, i, j, s, max;
printf (“Введите 3 строки по 5 чисел”);
for (i=0;i<3;i++)
 for (j=0;j<5;j++)
  scanf("%d",&a[i][j]);
printf (“Матрица a :\n”);
 for (i=0; i<3; i++)
  {for (j=0; j<5; j++)
    printf (“%5d”, a[i][j]);
    printf (“\n”);
  }
P=&a[0][0];
for(i=0;i<3;i++)
  {s=0;
    for (j=0;j<5;j++)
      {s+=*P;
       P++;
      }
    if (i==0) max=s;
    else if (max<s) max=s;
  }
printf ("Максимальная сумма строки = %d",max);
}
```

Контрольные вопросы

1. Особенности организации двумерных массивов: понятие массива в языке C++,

описание массива в программе, представление элементов массива в памяти, обращение к элементам массива.

2. Указатели в языке C++: понятие указателя, описание указателя в программе.
3. Операции над указателями.
4. Связь массивов и указателей.

Варианты заданий

1. Вычислить сумму положительных элементов каждого столбца матрицы A(txп).
2. Из матрицы *X* (mxn) построить матрицу *Y*, поменяв местами строки и столбцы.
3. Найти наименьший элемент матрицы *X* (mxn) и записать нули в ту строку и столбец, где он находится.
4. Переписать первые элементы каждой строки матрицы *A* (mxn), большие *C*, в массив *B***.** Если в строке нет элемента, большего *C*, то записать ноль в массив *B*.
5. Дана действительная матрица размера *mxn*. Найти сумму наибольших значений элементов ее строк.
6. В данной действительной матрице размера mxn поменять местами строку, содержащую элемент с наибольшим значением, со строкой, содержащей элемент с наименьшим значением. Предполагается, что такой элемент единственный.
7. В данной действительной квадратной матрице порядка *n* найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.
8. Дана действительная матрица размера *mxn*, все элементы которой различны. В каждой строке выбирается элемент с наименьшим значением, затем среди этих чисел выбирается наибольшее. Указать индексы элемента с найденным значением.
9. Дана целочисленная матрица размера mxn. Найти матрицу, получающуюся перестановкой столбцов (первого с последним, второго с предпоследним и т.д.).
10. Дана целочисленная матрица размера mxn. Найти матрицу, получающуюся перестановкой строк (первой с последней и т.д.).
11. Дана действительная матрица [*aij*], где i, j = 1..*n*. Получить действительную матрицу [*bij*], где i, j = 1..*n*, элемент *bij* которой равен сумме элементов данной матрицы, расположенных в области, определяемой индексами i, j (область заштрихована на рисунке 12):

Рисунок 12

12. Дана действительная квадратная матрица порядка *n*. Преобразовать матрицу по правилу: строку с номером *n* сделать столбцом с номером *n*, а столбец с номером *n* сделать строкой с номером *n*.
13. Просуммировать элементы матрицы X (4,5), сумма индексов которых равна заданной константе K.
14. Дана матрица M (4x5). Вычислить вектор D, компоненты которого равны сумме элементов строк матрицы.
15. Дана матрица M (6x6). Вычислить сумму элементов главной диагонали.
16. Дана матрица N (6x5). Найти столбец с минимальной суммой элементов.
17. Дана матрица M (4x5) и константа C. Вычислить матрицу D, равную произведению элементов матрицы M на константу.

18. Дана матрица M (4x6). Вычислить вектор D, компоненты которого равны сумме элементов столбцов матрицы.

19. Дана действительная квадратная матрица порядка *n*, все элементы которой различны. Найти наибольший элемент, среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.

20. Дана действительная квадратная матрица порядка n. Найти наибольшее из значений элементов, расположенных в заштрихованной части матрицы (рисунок 13):

Рисунок 13

21. Дана матрица M (2x5), определить максимальный и минимальный элементы. Поменять местами максимальный и минимальный элементы.

22. В матрице A(пхп) вычислить сумму элементов матрицы (n-2xn-2) и определить максимальный элемент в ней.

23. Дана матрица M (6x6). Вычислить произведение элементов главной диагонали с константой C.

24. Дана матрица N (6x5). Найти строку с минимальной суммой элементов, а элемент с номером nij возвести в квадрат.

25. Дана матрица вещественных чисел A (mxn). В сроке m определить максимальный элемент, а в столбце n количество элементом, меньших порога k.

Практическое занятие № 11

Разработка программ с использованием строк

Цель работы: изучение правил описания, ввода-вывода и основных функций обработки символьных данных.

Разработка программ с использованием строк **Краткие теоретические сведения**.

Строки в С++ позволяют работать с символьными данными. С помощью строк возможно осуществить чтение с клавиатуры текста, его обработка и вывод.

В С++ существует 2 типа строк:

1. Массив переменных типа char, заканчивающийся нуль-терминатором \0. Символьные строки состоят из набора символьных констант заключённых в двойные кавычки. При объявлении строкового массива необходимо учитывать наличие в конце строки нуль-терминатора, и отводить дополнительный байт под него.

char string[10]; // string - имя строковой переменной, 10 - размер массива, (в строке может поместиться 9 символов, последнее место отводится под нуль-терминатор)

2. Специальный класс string

Для его работы необходимо в начале программы подключить заголовочный файл string: *#include <string>*

Для создания строки необходимо в начале программы написать

using namespace std;

Теперь чтоб создать строку достаточно написать: *string s;*

Для записи в строку можно использовать оператор =

s="Hello";

Доступ к i-му элементу строки **s** типа string осуществляется стандартным образом s[i]. Над строками типа string определены следующие операции:

1. присваивания, например s1=s2;
2. объединения строк (s1+=s2 или s1=s1+s2) — добавляет к строке si строку s2, результат храниться в строке si, пример объединения строк:
3. сравнения строк: si=s2, si!=s2, si<s2, si>s2, si<=s2, si>=s2 — результатом будет логическое значение.

Существует множество функций для работы со строками (таблица 1).

Таблица 1- Функции для работы со строками

Функция	Объяснение
s.append(str)	добавляет в конец строки строку str
s.assign(str)	присваивает строке s значение строки str
s.clear()	отчищает строку, т.е. удаляет все элементы в ней
s.compare(str)	сравнивает строку s со строкой str и возвращает 0 в случае совпадение
s.copy(C, I, N)	копирует из строки s в строку C(может быть как строка типа string, так и строка типа char) I символов, начиная с N-го символа
bool b=s.empty()	если строка пуста, возвращает true, иначе false
s.erase(I,N)	удаляет N элементов с I-го символа
s.find(str,I)	ищет строку str начиная с I- го символа
s.insert(pos, s1)	вставляет строку **si** в строку **s**, начиная с позиции **pos**
int len=s.length()	записывает в len длину строки
s.push back(symbol)	добавляет в конец строки символ

Функция	Объяснение
s.replace(index, n,str)	берет n первых символов из str и заменяет символы строки s на них, начиная с позиции index
str=s.substr(n,	возвращает m символов начиная с позиции n
s.swap(str)	меняет содержимое s и str местами.
s.size()	возвращает число элементов в строке.

Функции для работы со строками, прототипы которых описаны в заголовочном файле string.h:

1 Сравнение строк. Для сравнения строк используются функции strcmp и strncmp. Функция

int strcmp (const char *str1, const char *str2);

лексикографически сравнивает строки strl, str2 и возвращает -1, 0 или 1, если строка str1 соответственно меньше, равна или больше строки str2.

Функция

int strncmp (const char *str1, const char *str2, size_t n);

лексикографически сравнивает не более чем n первых символов из строк str1 и str2. Функция возвращает -1, 0 или 1, если первые n символов из строки str1 соответственно меньше, равны или больше первых n символов из строки str2.

Пример:

```
// пример сравнения строк
#include #include
int main() {
char str1[] = "aa bb";
char str2[] = "aa aa";
char str3[] = "aa bb cc";
printf("%d\n", strcmp(str1, str3)); // печатает: -1
printf("%d\n", strcmp(str1, str1)); // печатает: -0
printf("%d\n", strcmp(str1, str2)); // печатает: 1
printf("%d\n", strncmp(str1, str3, 5)); // печатает: 0 return 1;
}
```

2 Копирование строк. Для копирования строк используются функции strcpy и strncpy. Функция

char *strcpy (char *str1, const char *str2);

копирует строку str2 в строку str1. Строка str2 копируется полностью, включая завершающий нулевой байт. Функция возвращает указатель на str1. Если строки перекрываются, то результат непредсказуем.

Функция

char *strncpy (char *str1, const char *str2, size_t n);

копирует n символов из строки str2 в строку str1. Если строка str2 содержит меньше чем n символов, то последний нулевой байт копируется столько раз, сколько нужно для расширения строки str2 до n символов. Функция возвращает указатель на строку str1.

Пример:

```
char str1[80];
char str2 = "Copy string.";
strcpy ( str1, str2 );
printf ( str1 ); // печатает: Copy string.
```

3 Соединение строк. Для соединения строк в одну строку используются функции strcat и strncat. Функция

char* strcat (char *str1, const char *str2);

присоединяет строку str2 к строке str1, причем завершающий нулевой байт строки str1 стирается. Функция возвращает указатель на строку str1.

Функция

char* strncat (char *str1, const char *str2, size_t n);

присоединяет n символов из строки str2 к строке str1, причем завершающий нулевой байт строки str1 стирается. Функция возвращает указатель на строку str1. если длина строки str2 меньше n, то присоединяются только символы, входящие в строку str2. После соединения строк к строке str1 всегда добавляется нулевой байт. Функция возвращает указатель на строку str1.

Пример:

```
#include <stdio.h>
#include <string.h>
int main ( )
{
char str1[80] = "String ";
char str2 = "catenation ";
char str3 = "Yes No";
strcat ( str1, str2 );
printf ("%s\n", str1 );                    // печатает: String catenation
strncat ( str1, str3, 3 );
printf ("%s\n", str1 );                    // печатает: String catenation Yes
return 1;
}
```

4 Поиск символа в строке. Для поиска символа в строке используются функции strchr, strrchr, strspn, strcspn и strpbrk. Функция char* strchr (const char *str, int c);

ищет первое вхождение символа, заданного параметром c, в строку str. В случае успеха функция возвращает указатель на первый найденный символ, а в случае неудачи - NULL.

Функция

char* strrchr (const char *str, int c);

ищет последнее вхождение символа, заданного параметром c, в строку str. В случае успеха функция возвращает указатель на последний найденный символ, а в случае неудачи - NULL.

Пример:

```
#include <stdio.h>
#include <string.h>
int main ( )
{
char str[80] = "Char search";
printf ("%s\n", strchr ( str, 'r' ));                    // печатает: r search
printf ("%s\n", strrchr ( str, 'r' )); // печатает: rch
return 1;
}
```

Функция

size_t strspn (const char *str1, const char *str2);

возвращает индекс первого символа из строки str1, который не входит в строку str2.

Функция

size_t strcspn (const char *str1, const char *str2);

возвращает индекс первого символа из строки str1, который входит в строку str2.

Пример:

```
int main ( )
{
char str[80] = "123 abc";
printf ("n = %d\n", strspn ( str, "321" ); // печатает: n = 3
printf ("n = %d\n", strcspn ( str, "cba" ); // печатает: n = 4 return 1;
}
```

Функция

char* strpbrk (const char *str1, const char *str2);

находит первый символ в строке str1, который равен одному из символов в строке str2. В случае успеха функция возвращает указатель на этот символ, а в случае неудачи - NULL.

Пример.

```
char str[80] = "123 abc";
printf ("%s\n", strpbrk ( str, "bca" )); // печатает: abc
```

5. Сравнение строк. Для сравнения строк используются функция strstr. Функция

```
char* strstr ( const char *str1, const char *str2 );
```

находит первое вхождение строки str2 (без конечного нулевого байта) в строку str1. В случае успеха функция возвращает указатель на найденную подстроку, а в случае неудачи - NULL. Если указатель str1 указывает на строку нулевой длины, то функция возвращает указатель str1.

Пример:

```
char str[80] = "123 abc 456;
printf ("%s\n", strstr ( str, "abc"); // печать: abc 456
```

6. Разбор строки на лексемы. Для разбора строки на лексемы используется функция strtok. Функция

```
char* strtok ( char *str1, const char *str2 );
```

возвращает указатель на следующую лексему (слово) в строке str1, в которой разделителями лексем являются символы из строки str2. В случае если лексемы закончились, то функция возвращает NULL. При первом вызове функции strtok параметр str1 должен указывать на строку, которая разбирается на лексемы, а при последующих вызовах этот параметр должен быть установлен в NULL. После нахождения лексемы функция strtok записывает после этой лексемы на место разделителя нулевой байт.

Пример.

```
#include <stdio.h>
#include <string.h>
int main( )
{
char str[ ] = "12 34 ab cd";
char *p;
p = strtok ( str, " " );
while (p)
{
printf ( "%s\n", p ); // печатает в столбик значения: 12 34 ab cd
p = strtok ( NULL, " " );
}
return 1;
}
```

7. Определение длины строки. Для определения длины строки используется функция strlen. Функция

```
size_t strlen ( const char *str);
```

возвращает длину строки, не учитывая последний нулевой байт. Например, char str[] = "123";

```
printf ("len = %d\n", strlen ( str )); // печатает: len = 3
```

Пример выполнения задания.

Задание. Найти слова во введенной с клавиатуры строке, вывести их на экран и подсчитать их количество.

Схема программы представлена на рисунке 1:

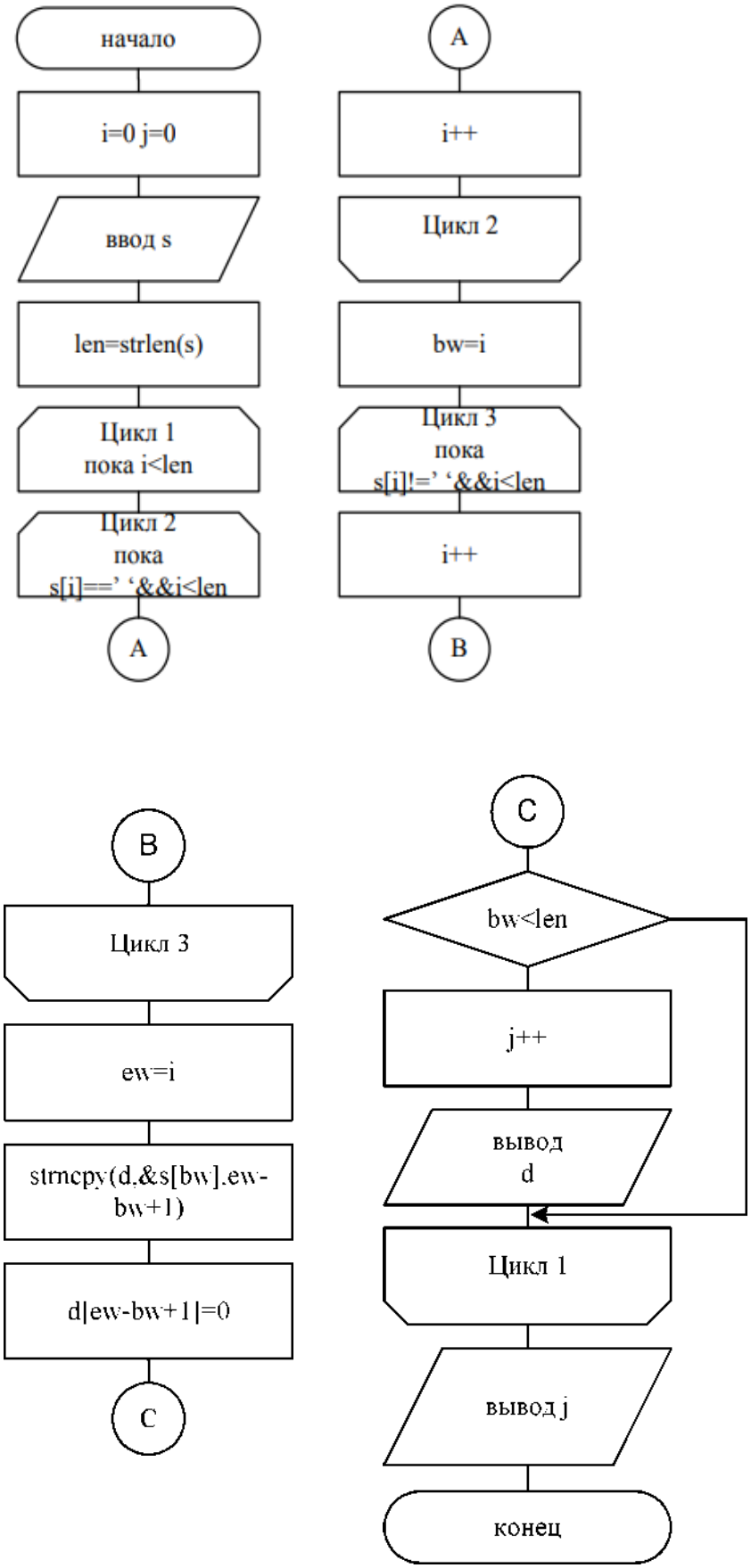

начало
i=0 j=0
ввод s
len=strlen(s)
Цикл 1
пока i<len
Цикл 2
пока
s[i]==' '&&i<len
A
A
i++
Цикл 2
bw=i
Цикл 3
пока
s[i]!=' '&&i<len
i++
B
B
Цикл 3
ew=i
strncpy(d,&s[bw],ew-bw+1)
d[ew-bw+1]=0
C
C
bw<len
j++
вывод
d
Цикл 1
вывод j
конец

Рисунок 1

Пример программы:

```
#include<stdio.h>
#include<string.h>
vid main()
{
 char s[100],d[100];
 int i=0,j=0,bw,ew,len;
 gets(s); len=strlen(s);
 while (i<len)
 {
   while((s[i]==' ')&&(i<len)) i++;
   bw=i;
   while((s[i]!=' ')&&(i<len)) i++;
   ew=i;
   strncpy(d,&s[bw],ew-bw+1);
   d[ew-bw+1]=0;
   if (bw<len)
      printf("%s\n",d);}
 }
printf("Vsego slov %d\n", j);
}
```

Контрольные вопросы

1. Строки в языке C++: понятие строки, описание строк в программе, обращение к элементам строки.
2. Три способа ввода строк в C++.
3. Три способа вывода строк в C++.
4. Способы инициализации строк (задание значений в программе).
5. Стандартные функции для обработки строк.

Варианты заданий

Дана последовательность символов S1, ..., SN. Группы символов, разделенные пробелом (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами.

1. Определить число символов в самом длинном слове строки. Слова отделяются знаком “/”.
2. В произвольном тексте выделить и отпечатать слова, начинающиеся с буквы А.
3. В произвольном тексте вставить между первым и вторым словом новое слово.
4. В произвольном тексте найти и отпечатать слова, содержащие букву Е.
5. Отпечатать второе и третье слова произвольного текста.
6. В произвольном тексте вставить между вторым и третьим словом новое слово.
7. В произвольном тексте найти и отпечатать все слова длиной 5 символов.
8. В произвольном тексте найти самое короткое слово.
9. В последовательности из 10 пятибуквенных слов найти и поменять местами пару слов, у которых первые три буквы одного совпадают с последними тремя буквами другого.
10. Упорядочить в алфавитном порядке последовательность из 10 пятибуквенных слов.
11. В строке из 50 символов отдельные слова разделены пробелом. Упорядочить

строку так, чтобы каждое следующее слово было не короче предыдущего.

12. Расположить слова строки в порядке, обратным исходному.

13. Подсчитать количество букв ‘а’ в последнем слове строки.

14. Найти количество слов, у которых первый и последний символы совпадают между собой.

15. Исключить из строки слова, расположенные между скобками (,). Сами скобки должны быть исключены.

16. В произвольном тексте найти и отпечатать слова, содержащие букву А.

17. Отпечатать первое и второе слова произвольного текста.

18. В произвольном тексте вставить после первого слова новое слово.

19. В произвольном тексте найти и отпечатать все слова длиной 4 символа.

20. В произвольном тексте найти самое длинное слово.

21. Выполнить сравнение двух строк s и d. Результат вывести в виде сообщения «идентичны» или «не идентичны».

22. Добавить в конец строки новое слово, длинною 5 симвлов, иначе выдать сообщение об ошибке.

23. Добавить в начало строки новое слово, начинающееся с буквы а, иначе, если слово начинается с другой буквы вывести сообщение о невозможности добавления.

24. Посчитать какое количество раз встречается буква n (задается при каждом выполнении алгоритма).

25. Проанализировать массив символов, состоящий из n символов. Если массив состоит из n-5 символов, добавить в конец набор символов rrrrr.

Практическое занятие № 12

Разработка программ с использованием функций

Цель работы: Освоение методов определения функций, передачи аргументов, использования библиотек функций, основ нисходящей технологии программирования.

Краткие теоретические сведения

Функции разбивают большие вычислительные задачи на маленькие подзадачи и позволяют использовать в работе то, что уже сделано другими, а не начинать каждый раз с пустого места. Соответствующие функции часто могут скрывать в себе детали проводимых в разных частях программы операций, знать которые нет необходимости, проясняя тем самым всю программу, как целое, и уменьшая трудности при внесении изменений.

Широко используются «библиотечные» функции. Каждая функция языка С имеет имя и список аргументов (формальных параметров).

Функции могут возвращать значение. Это значение может быть использовано далее в программе. Так как функция может вернуть какое-нибудь значение, то обязательно нужно указать тип данных возвращаемого значения. Если тип не указан, то по умолчанию предполагается, что функция возвращает целое значение (тип int). После имени функции принято ставить круглые скобки (это касается вызова функции, её объявления и описания). В этих скобках перечисляются параметры функции, если они есть. Если у функции нет параметров, то при объявлении и при описании функции вместо <список параметров> в приведенном далее примере кода надо поставить void - пусто.

Основная форма описания функции имеет вид:

```
        тип <имя функции>(список параметров)
{
      тело функции
}
```

Объявление (прототип) функции имеет вид:

```
        тип <имя функции>(список параметров);
```

Обратите внимание на то, что при описании функции после заголовка функции точка с запятой не ставится, а при объявлении функции точка с запятой ставится.

Вызов функции делается следующим образом:

```
        <имя функции>(параметры);
```

или

```
        <переменная>=<имя функции>(параметры);
```

При вызове функции так же ставится точка с запятой.

Почему надо объявлять функцию до использования? Дело в том, что для правильной работы кода функции машине надо знать тип возвращаемого значения, количество и типы аргументов. При вызове какой-либо функции копии значений фактических параметров записываются в стек, в соответствии с типами указанными в ее прототипе. Затем происходит переход в вызываемую функцию.

Приведем пример вызова функции

```
        /* Используем свою функцию */
void function1(void);      // Объявление функции
void main(void)       // Точка входа в программу
{
    function1();              //  Вызов функции
}
/* Описание функции  */
void function1(void)          // Заголовок функции
{      // Начало тела функции
```

```
        }      //  Конец тела функции
```

Мы объявили функцию function1(), затем её вызвали. Необходимо обратить внимание ещё на то, что тип возвращаемого значения у функции void (пусто). Это значит, что функция не будет возвращать никакого значения.

Рассмотрим пример, в котором описаны две функции, для одной из которых указан тип возвращаемого значения - int.

```
        int x;// Объявляем переменную x (глобальная переменная)
        void main(void)
{
      void function1(void);      // Объявляем функцию
      int function2();           // function2() будет
         // возвращать значение типа int
              x = 10; // Присваиваем переменной x значение
              function1(); // Вызываем функцию function1()
              x = function2();// Вызываем функцию function2()
}

/* Описание функций */
void function1(void)
{

        }
int function2(void)
{
      int y; // Объявляем локальную переменную
      y = x + 10;
      return y; // Возвращаем значение y
}
```

Теперь давайте посмотрим текст программы. Объявляем глобальную переменную x. Так как x - глобальная переменная, то она будет видна всем функция программы. В теле main() объявляем две функции, одна из которых может возвращать значение типа int. Далее присваиваем переменной x значение 10, так как x это глобальная переменная, то эта переменная будет видна функции main() т.е. функция main() может использовать эту переменную.

Далее мы вызываем функцию function1(). В следующей строке x = function2(); переменная x принимает значение которое вернет функция function2(). Посмотрите на описание функции function2(). В теле этой функции объявляем переменную y, а дальше переменной y мы присваиваем значение переменной x + 10. Так как x - глобальная переменная (она видна для функции function2), все будет работать.

Далее идет строка return y; с помощью оператора return мы возвращаем значение переменной y. Запомните, что если функция возвращает значение, то в теле этой функции обязательно должен присутствовать оператор return (он может быть и не один). Ну так вот с помощью оператора return мы возвращаем значение локальной переменной y в вызывающую функцию main().

Функции языка С могут иметь параметры. Эти параметры передаются в функцию и там обрабатываются. Ещё раз рассмотрим основную форму описания функции:

```
        тип <имя функции>(список параметров)
{
      тело функции
}
```

В списке параметров для каждого параметра должен быть указан тип.
Пример правильного списка параметров:

```
function(int x, char a, float z)
```

Пример неправильного списка параметров:

```
function(int x, a, float z)
```

Рассмотрим все это на примере. Пусть будет функция, у которой присутствует один параметр x. Функция будет возвращать квадрат значения x.

```
int square(int x)
{
    x = x * x;
    return x;
}
```

Формальные параметры - это параметры которые объявляются в заголовке функции при описании.

Фактические параметры - это параметры которые подставляются при вызове функции.

```
void myfunc(int x); // Объявление функции
void main(void)
{
   int a;
   a=5;
   myfunc(a); // a- фактический параметр
}
// Описание функции
void myfunc(int x) // x - формальный параметр
{
   x = x + 10;
}
```

В языке С функция может возвращать несколько значений. Чтобы функция могла вернуть несколько значений необходимо пользоваться указателями. Этот механизм называется - передача параметров по ссылке.

В С++ при вызове функций можно опускать параметры. В таких случаях для опущенных параметров будут использоваться значения по умолчанию. Обеспечение значений по умолчанию для параметров упрощает возможность повторного использования функций (их использования несколькими программами).

Применение библиотечных функций сокращает объем программирования, который программист должен выполнить самостоятельно. Вместо этого программа просто вызывает функции библиотеки этапа выполнения. В зависимости от компилятора библиотека этапа выполнения может состоять из тысяч функций.

Ранее было сказано, что до того, как программа сможет вызвать функцию, компилятор С++ должен узнать определение или прототип функции. Поскольку функции библиотеки этапа выполнения не определены в основной программе, необходимо указать прототип для каждой библиотечной функции, которую намерен использовать программист. Для упрощения использования библиотечных функций компилятор С++ предоставляет заголовочные файлы, содержащие корректные прототипы. Таким образом, программам необходимо просто включить требуемый заголовочный файл с помощью оператора #include, а затем вызвать необходимую функцию.

3. Порядок выполнения работы

1. Ознакомиться с теоретическими сведениями.
2. Получить вариант задания у преподавателя.
3. Выполнить задание.

4. Продемонстрировать выполнение работы преподавателю.
5. Оформить отчет.
6. Защитить лабораторную работу.

4. Требования к оформлению отчета

Отчет по лабораторной работе должен содержать следующие разделы:
титульный лист;
цель работы:
задание на лабораторную работу;
техническое описание выполненного задания;
блок-схему;
ответы на контрольные вопросы;
выводы по проделанной работе.

5. Варианты заданий

Во всех вариантах необходимо использовать пользовательские функции. Предусмотреть режим диалога.

1. Треугольник задан координатами своих вершин. Составить программу для вычисления его площади.
2. Составить программу для нахождения наибольшего общего делителя четырех натуральных чисел.
3. Составить программу для нахождения наименьшего общего кратного трех натуральных чисел.
4. Написать программу для нахождения суммы большего и меньшего из трех чисел.
5. Вычислить площадь правильного шестиугольника со стороной a, используя функцию вычисления площади треугольника.
6. На плоскости заданы своими координатами n точек. Составить программу, определяющую, между какими из пар точек самое большое расстояние. Указание. Координаты точек занести в массив.
7. Составить программу, которая в массиве A[10] находит второе по величине число (вывести на печать число, которое меньше максимального элемента массива, но больше всех других элементов).
8. Написать программу для вычисления суммы факториалов всех нечетных чисел от 1 до 9.
9. Даны две дроби A/B и C/D (A, B, C, D — натуральные числа). Составить программу для деления дроби на дробь. Результат должен быть несократимой дробью.
10. Задан массив D. Определить следующие суммы: D[1] + D[2] + D[3]; D[3] +D[4] +D[5]; D[4] + D[5] + D[6]. Пояснение. Составить подпрограмму для вычисления суммы трех последовательно расположенных элементов массива с номерами k до m.
11. На плоскости заданы своими координатами n точек. Создать массив размером n(n-1), элементами которого являются расстояния от каждой из точек до n -1 других.
12. Даны числа X, Y, Z, T — длины сторон четырехугольника. Вычислить его площадь, если угол между сторонами длиной X и Y— прямой.
13. Составить программу для вычисления суммы факториалов всех четных чисел от m до n.
14. Заменить отрицательные элементы линейного массива их модулями, не пользуясь стандартной функцией вычисления модуля. Подсчитать количество произведенных замен.
15. Дано простое число. Составить функцию, которая будет находить следующее за ним простое число.
16. Составить функцию для нахождения наименьшего нечетного натурального делителя k (k не равно 1) любого заданного натурального числа n.

17. Дано натуральное число N. Составить программу для формирования массива, элементами которого являются цифры числа N.
18. Составить программу, определяющую, в каком из данных двух чисел больше цифр.
19. Заменить данное натуральное число на число, которое получается из исходного записью его цифр в обратнov порядке.
20. Даны три квадратных матрицы A, B, C размерность 10. Вывести на печать ту из них, норма которой наименьшая. Пояснение. Нормой матрицы назовем максимум из абсолютных величин ее элементов.
21. Два натуральных числа называются «дружественными», если каждое из них равно сумме всех делителей (кроме его самого) другого числа (например, числа 220 и 284). Найти все пары «дружественных чисел», которые не больше данного числа N.
22. Два простых числа называются «близнецами», если они отличаются друг от друга на 2 (например, 41 и 43). Напечатать все пары «близнецов» из отрезка [n, 2n], где n — заданное натуральное число больше 2.

Контрольные вопросы

1. Зачем нужно использовать функции ?
2. Чем отличается прототип и определение функции?
3. Для чего нужны прототипы функций?
4. Что такое формальные и фактические параметры? Пример.

Практическое занятие № 13
Разработка программ с использованием рекурсивных функций

Цель работы: Научиться решать задачи, используя рекурсивные функции.

Краткие теоретические сведения

Суть рекурсивных методов—сведение задачи к самой себе. Вы уже знаете, что в С существует возможность рекурсивного определения функций. Эта возможность представляет собой способ программной реализации рекурсивных алгоритмов. Однако увидеть рекурсивный путь решения задач и (рекурсивный алгоритм) часто очень непросто.

Классическая задача - «Ханойская башня». На площадке (назовем ее А) находится пирамида, составленная из дисков уменьшающегося от основания к вершине размера. Эту пирамиду в том же виде требуется переместить на площадку В. При выполнении этой работы необходимо соблюдать следующие ограничения:

•перекладывать можно только по одному диску, взятому сверху пирамиды;
•класть диск можно либо только на основание площадки, либо на диск большего размера;
•в качестве вспомогательной можно использовать площадку С.

Название «Ханойская башня» связано с легендой, согласно которой в давние времена монахи одного ханойского храма взялись переместить по этим правилам башню, состоящую из 64 дисков. С завершением их работы наступит конец света. Монах и все еще работают и, надеемся, еще долго будут работать!

Нетрудно решить эту задачу для двух дисков. Обозначая перемещения диска, например, с площадки А на В так: А—>В, напишем алгоритм для этого случая
A-)C;A->B;C->B.

Функция Сложения двух натуральных чисел ($Sum(a,\ b) = a + b$) может быть рассмотрена в качестве рекурсивной функции двух переменных
$Sum(x,\ y+1) = F(x,\ y,\ Sum(x,\ y))$;

Умножение двух натуральных чисел ($Mul(a,\ b) = a \times b$) может быть рассмотрено в качестве рекурсивной функции двух переменных
$Mul(x,\ y+1) = G(x,\ y,\ Mul(x, y))$;

Рекурсивной называется функция, которая вызывает саму себя. Такая рекурсия называется прямой. При косвенной рекурсии, две или более функций вызывают друг друга. При рекурсивном вызове функции в стеке создается копия значений ее параметров, как и при вызове обычной функции, после чего управление передается первому исполняемому оператору функции. При повторном вызове этот процесс повторяется. Ясно, что для завершения вычислений каждая рекурсивная функция должна содержать хотя бы одну нерекурсивную ветвь алгоритма, заканчивающуюся оператором возврата.

При завершении функции соответствующая часть стека освобождается, и управление передается вызывающей функции, выполнение которой продолжается с точки, следующей за рекурсивным вызовом.

Приведем пример рекурсивной функции, вычисляющей факториал.

```
long fact(long n){
if (n==0 || n==1) return 1;
return (n * fact(n - 1);
}
```

Содержание работы.

1. Ознакомиться с теоретической частью методических указаний.
2. Получить вариант задания у преподавателя.
3. Решить поставленную задачу с использованием языка программирования С++.
4. Представить работающую программу преподавателю.

Варианты заданий

Решить следующие задачи, используя рекурсивную функции.

1. Найти сумму цифр заданного натурального числа.
2. Подсчитать количество цифр в заданном натуральном числе.
3. Составить программу для вычисления наибольшего общего делителя двух натуральных чисел.
4. Составить программу для нахождения числа, которое образуется из данного натурального числа при записи его цифр в обратном порядке. Например, для числа1234 получаем результат 4321.
5. Составить программу для перевода данного натурального числа в 7 -ричную систему счисления.
6. Дана символьная строка, представляющая собой запись натурального числа в 7-ичнойсистемесчисления(2<р<9).Составить программу для перевода этого числа в десятичную систему счисления.
7. Составить программу для вычисления суммы: 2! + 4! + 6! +

+ ... + «! (п< 16, п — четное).

Примечание. Тип результата значения функции — Longlnt.

8. Дано п различных натуральных чисел. Напечатать все перестановки

этих чисел.

9. Логическая функция возвращает True, если ее аргумент —

простое число.

10. Описать функцию, которая удаляет из строки все лишние пробелы. Пробелы считаются лишними, если их подряд идет более двух, если они стоят в конце строки после последней точки, если стоят после открывающегося парного знака препинания.

Содержание отчета.

1. Цель работы и задание.
2. Пункты, соответствующие порядку выполнения работы.
3. Выводы по работе.

Практическое занятие № 14
Разработка программ работы с файлами

Цель работы: научиться выполнять ввод-вывод из файла с помощью стандартной библиотеки

Краткие теоретические сведения

Частью стандартной библиотеки C++ является библиотека iostream – объектно-ориентированная иерархия классов, где используется и множественное, и виртуальное наследование. В ней реализована поддержка для файлового ввода/вывода данных встроенных типов. Кроме того, разработчики классов могут расширять эту библиотеку для чтения и записи новых типов данных.

Для использования библиотеки iostream в программе необходимо включить заголовочный файл

```
#include <iostream>
```

Операции ввода/вывода выполняются с помощью классов istream (потоковый ввод) и ostream (потоковый вывод). Третий класс, iostream, является производным от них и поддерживает двунаправленный ввод/вывод. Для удобства в библиотеке определены три стандартных объекта-потока:

- cin – объект класса istream, соответствующий стандартному вводу. В общем случае он позволяет читать данные с терминала пользователя;
- cout – объект класса ostream, соответствующий стандартному выводу. В общем случае он позволяет выводить данные на терминал пользователя;
- cerr – объект класса ostream, соответствующий стандартному выводу для ошибок. В этот поток мы направляем сообщения об ошибках программы.

Помимо чтения с терминала и записи на него, библиотека iostream поддерживает чтение и запись в файлы. Для этого предназначены следующие классы:

- ifstream, производный от istream, связывает ввод программы с файлом;
- ofstream, производный от ostream, связывает вывод программы с файлом;
- fstream, производный от iostream, связывает как ввод, так и вывод программы с файлом.

Чтобы использовать часть библиотеки iostream, связанную с файловым вводом/выводом, необходимо включить в программу заголовочный файл

```
#include <fstream>
```

Если файл будет использоваться только для вывода, мы определяем объект класса ofstream. Например:

```
ofstream outfile( "copy.out", ios::base::out );
```

Передаваемые конструктору аргументы задают имя открываемого файла и режим открытия. Файл типа ofstream может быть открыт либо – по умолчанию – в режиме вывода (ios_base::out), либо в режиме дозаписи (ios_base::app). Такое определение файла outfile2 эквивалентно приведенному выше:

```
// по умолчанию открывается в режиме вывода
ofstream outfile2( "copy.out" );
```

Если в режиме вывода открывается существующий файл, то все хранившиеся в нем данные пропадают. Если же мы хотим не заменить, а добавить данные, то следует открывать файл в режиме дозаписи: тогда новые данные помещаются в конец. Если указанный файл не существует, то он создается в любом режиме.

Прежде чем пытаться прочитать из файла или записать в него, нужно проверить, что файл был успешно открыт:

```
if ( ! outfile ) { // открыть файл не удалось
  cerr <<"не могу открыть "copy.out" для записи\n";
  exit( -1 );
}
```

Класс ofstream является производным от ostream. Все определенные в ostream операции применимы и к ofstream.

Следующая программа читает из стандартного ввода символы и копирует их в стандартный вывод:

```
#include <fstream>
int main()
{
   // открыть файл copy.out для вывода
   ofstream outFile( "copy.out" );
   if ( ! outFile ) {
      cerr << "Не могу открыть 'copy.out' для вывода\n";
      return -1;
   }
   char ch;
   while ( cin.get( ch ) )
   outFile.put( ch );
}
```

Чтобы открыть файл только для чтения, применяется объект класса ifstream, производного от istream. Следующая программа читает указанный пользователем файл и копирует его содержимое на стандартный вывод:

```
#include <fstream>
#include <string>
int main()
{
   cout << "filename: ";
   string file_name;
   cin>> file_name;
   // открыть файл для ввода
   ifstream inFile( file_name.c_str() );
   if ( !inFile ) {
       cerr < <"не могу открыть входной файл: "
           <<file_name << " -- аварийный останов!\n";
       return -1;
   }
   char ch;
   while ( inFile.get( ch ))
          cout.put( ch );
}
```

Объекты классов ofstream и ifstream разрешено определять и без указания имени файла. Позже к этому объекту можно присоединить файл с помощью функции-члена open():

```
ifstream curFile;
// ...
curFile.open( filename.c_str() );
if ( ! curFile )  // открытие успешно?
  // ...
```

Чтобы закрыть файл, вызываем функцию-член close():

```
.
.
.
.
```

```
    curFile.close();
  }
}
```

Объект класса fstream может также открывать файл одновременно для ввода и вывода. Например, приведенная инструкция открывает файл word.out для ввода и дозаписи:

```
fstream io( "word.out", ios_base::in|ios_base::app );
```

Для задания нескольких режимов используется оператор побитового ИЛИ.

Порядок выполнения работы

1. Ознакомиться с теоретическими сведениями.
2. Получить вариант задания у преподавателя.
3. Выполнить задание.
4. Продемонстрировать выполнение работы преподавателю.
5. Оформить отчет.
6. Защитить лабораторную работу.

Требования к оформлению отчета

Отчет по лабораторной работе должен содержать следующие разделы:
титульный лист;
цель работы:
задание на лабораторную работу;
техническое описание выполненного задания;
блок-схему;
выводы по проделанной работе.

Варианты заданий

1. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Получить в файле g те компоненты файла f, которые являются четными.
2. Записать в файл 10 действительных чисел. Вычислить произведение компонентов файла и вывести на печать.
3. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Получить в файле g все компоненты файла f, которые делятся на 2.
4. Записать в файл 10 целых чисел, полученных с помощью генератора случайных чисел. Подсчитать количество пар противоположных чисел среди компонентов этого файла.
5. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Из файла f получить файл g, исключив повторные вхождения чисел.
6. Записать в файл 10 произвольных натуральных чисел. Переписать в другой файл те элементы, которые кратны К.
7. Заполнить файл 10 действительными числами, полученными с помощью датчика случайных чисел. Найти сумму минимального и максимального элементов этого файла.
8. Записать в файл 15 натуральных чисел: a_1, a_2,..., a_n (числа получить с помощью датчика случайных чисел). Сформировать новый файл, элементами которого являются числа a_1, a_1*a_2, $a_1*a_2*a_3$, ….
9. Записать в файл f 10 натуральных чисел. Получить в другом файле все компоненты файла f, кроме тех, которые кратны К.
10. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Найти количество удвоенных нечетных чисел среди компонентов файла.

11. Заполнить файл f натуральными числами, полученными с помощью генератора случайных чисел. Найти количество квадратов нечетных чисел среди компонентов.
12. Записать в файл 10 действительных чисел. Найти наибольшее из значений модулей компонентов с нечетными номерами.
13. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Из файла f получить файл g, исключив повторные вхождения чисел. Порядок следования чисел сохранить.
14. Записать в файл 10 действительных чисел. Найти разность первого и последнего компонентов файла.
15. Записать в файл f 10 целых чисел, полученных с помощью генератора случайных чисел. Заполнить файл g числами, которые являются произведениями соседних компонентов файла f.
16. Багаж пассажира характеризуется количеством вещей и их общим весом. Дан файл Bagazh, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже каждого пассажира представляют собой запись с двумя полями: одно поле целого типа (количество вещей) и одно — действительного (вес в килограммах). Найти багаж, средний вес одной вещи в котором отличается не более чем на t кг от общего среднего веса одной вещи.
17. В условиях предыдущей задачи найти число пассажиров, имеющих более двух вещей, и число пассажиров, количество вещей которых превосходит среднее число вещей.
18. В условиях задачи 17 выяснить, имеется ли пассажир, багаж rоторого состоит из одной вещи весом менее t кг.
19. Дан файл Bibl, содержащий сведения о книгах. Сведения о каждой из книг — это фамилия автора, название и год издания. Найти названия книг данного автора, изданных начиная с 2000 г.
20. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Получить название игрушек, цена которых не превышает 1400 руб. и которые подходят детям 5 лет.
21. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет).
22. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Определить стоимость самого дорогого конструктора.
23. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Напечатать название наиболее дорогих игрушек (цена которых отличается от цены самой дорогой игрушки не более чем на 500 руб.);
24. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Получить названия игрушек, которые подходят детям как четырех, так и десяти лет.
25. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Получить сведения о

том, можно ли подобрать игрушку, любую, кроме мяча, подходящую ребенку трех лет.

26. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Получить название самой дешевой игрушки.
27. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Получить название самой дорогой игрушки для детей до четырех лет.
28. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Получить названия игрушек для детей четырех-пяти лет.
29. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Получить название самой дорогой игрушки, подходящей детям двух-трех лет.
30. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Определить стоимость самой дорогой куклы.
31. Дан файл Assort, содержащий сведения об игрушках: указываются название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Определить суммарную стоимость кукол для детей шести лет.

Практическое занятие № 15
Основы работы с интегрированной средой
Краткие теоретические сведения

Microsoft Visual Studio .NET 20xx. В Visual Studio .NET (далее VS .NET) каждый проект является частью того, что Microsoft называет решением (solution). Любой код, созданный в VS .NET IDE, относится к некоторому решению. Решение можно рассматривать как хранилище всей информации, необходимой для компиляции программы и ее перевода в форму, пригодную для исполнения. Таким образом, решение состоит из одного или нескольких проектов; различных вспомогательных файлов (графических изображений, ресурсных файлов, метаданных, то есть данных, описывающих другие данные, и т. д.); документации в формате XML. Решение позволяет легко выбрать файлы, задействованные в решении конкретной проблемы.

Новое решение создается командой File > New > Project. Далее требуется выбрать тип проекта, который будет первым в решении, имя решения, и каталог где оно будет находиться. Все новые проекты добавляются в решение лишь с одним отличием, при создании проекта в поле «решение» требуется выбрать «В текущее» (Add to Solution) вместо «в новое решение» (Create new solution). При помощи команды View в главном меню всегда можно вызвать нужное окно на передний план (и передать ему фокус). Все окна IDE свободно перетаскиваются мышью.

Основные окна IDE.

Редактор текста. Редактор обладает полным набором стандартных возможностей, поддерживаемых в редакторах такого рода (вырезание, вставка, поиск/замена и т. д.). Для работы с ними можно использовать стандартные комбинации клавиш Windows (Ctrl+X — вырезать, Ctrl+V — вставить и т. д.). Если вы предпочитаете работать с командами меню, к вашим услугам меню Edit и контекстное меню окна программы. Полный список сочетаний клавиш вызывается из меню Edit; кроме того, он приведен в разделе «Editing, shortcut keys» справочной системы. Например, комбинация Ctrl+I включает режим поиска с приращением.

В распоряжении разработчика имеется средство IntelliSense, выдающее информацию о методах заданного объекта или параметрах, передаваемых при вызове функции. Обычно IntelliSense вызывается автоматически, но его можно вызвать нажатием «Ctrl + Пробел».

Настройка большинства глобальных параметров редактора выполняется в диалоговом окне — выполните команду Tools > Options и выберите в списке строку Text Editor. Например, чтобы выбрать размер позиций табуляции, щелкните в строке Text Editor и выберите нужное значение для всех языков или только для C++. Здесь же выбирается режим создания отступов: None (отступы отсутствуют), Block (курсор выравнивается по началу предыдущей строки) или Smart (автоматическое создание отступов в теле цикла, как того требует хороший стиль программирования). Кстати говоря, устанавливать размер позиций табуляции и форматировать отступы можно в готовом тексте, для чего используются комбинации клавиш Ctrl+K, Ctrl+F (сочетания клавиш требуется нажать подряд, без длительной паузы) или команда Edit > Advanced > Format Selection.

Также изменять отступы выделенного блока можно используя Tab для увеличения отступа и Shift + Tab для уменьшения.

Редактор поддерживает и такую возможность, как свертка фрагментов программы и отображение на их месте заголовков (folding). Обратите внимание на значки «-» рядом с некоторыми строками. Если щелкнуть на таком значке, в листинге будет скрыта соответствующая область (region), а после первой строки кода из блока появится многоточие. Если задержать указатель мыши над многоточием, на экране будет показан свернутый код. Для управления сверткой используется подменю Edit > Outlining. Информация о редакторе IDE находится в разделе справки «Code and Text Editor».

Список задач. В Visual Studio поддерживается список задач (task list). Идея состоит в том, что в программу включаются комментарии с описанием действий, которые предполагается выполнить в будущем; тип задачи определяется специальным ключевым словом, следующим после знака комментария. В настоящее время определены три встроенные категории задач — TODO, HACK и UNDONE. Комментарии с задачами выводятся в окне, вызываемом командой View > Other Windows > Task List (или комбинацией клавиш Ctrl+/, Ctrl + T).

Окно решения. В окне решения (Solution Explorer) выводится список файлов, входящих в решение. По умолчанию имя решения совпадает с именем первого созданного в нем проекта. Используя Solution Explorer, можно добавлять в проект различные файлы. Например, текст или исходный код. Для этого требуется щелкнуть правой кнопкой мыши по папке, куда необходимо добавить файл, в контекстном меню выбрать Add > New Item (Добавить > Новый элемент), чтобы добавить новый файл, или Add > Existent Item (Добавить > Существующий элемент), чтобы добавить существующий файл.

Окно свойств. Функции окна свойств в VS .NET уже не ограничиваются простым заданием свойств элементов управления. Содержимое окна зависит от того, что в настоящий момент выделено в IDE. Имя и тип выделенного элемента указаны в списке, находящемся в верхней части окна. Чтобы изменить значение свойства, щелкните в правой ячейке и начинайте вводить символы. В окне свойств действуют стандартные комбинации клавиш, используемые при редактировании в системе Windows.

Окно вывода и окно ошибок. В окне вывода (вызываемом командой View > Output или комбинацией клавиш Alt + 2) отображается текущая информация состояния. При построении решения в этом окне компилятор выводит сообщения как об успешном завершении, так и о возникших ошибках. В окне ошибок выводятся все ошибки или предупреждения возникшие во время компиляции программы. Оно вызывается сочетанием клавиш Ctrl + /, Ctrl + E.

Порядок выполнения работы

1. Ознакомиться с теоретическими сведениями.
2. Получить вариант задания у преподавателя.
3. Выполнить задание.
4. Продемонстрировать выполнение работы преподавателю.
5. Оформить отчет.
6. Защитить лабораторную работу.

Требования к оформлению отчета

Отчет по лабораторной работе должен содержать следующие разделы:
титульный лист;
цель работы:
задание на лабораторную работу;
техническое описание выполненного задания;
выводы по проделанной работе.

Задание на работу

1. Создать решение, содержащее консольное приложение.
2. Добавить проекту файл с исходным кодом следующего содержания:

```
#include “stdafx.h”
int sum( int a, int b){
   return a + b;
}
```

3. В функцию main добавить следующий код:

```
printf(“Sum: %d”,sum(3, 5));
```

4. Запустить созданный проект (клавиша F5). Объяснить, что выведено в окно вывода.
5. Добавить заголовочный файл с прототипом функции sum:

int sum(int a, int b)

6. Добавить в файл с описанием функции main подключение вновь созданного заголовочного файла #include “имя_файла.h”.
7. Заново запустить программу. Объяснить, что произошло.
8. Добавить в одном из файлов с исходным кодом новые задачи. Добавьте новые задачи через список задач. В чем разница?

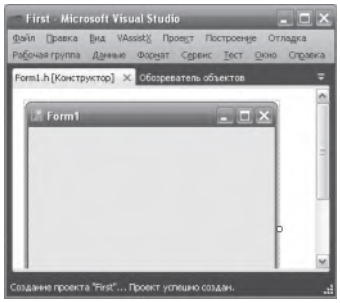
Контрольные вопросы

1. Что такое решение (solution) в Visual Studio .NET? Зачем они нужны?
2. Как создать решение? Как добавить туда проект?
3. Что такое IntelliSense? Как он вызывается?
4. Что такое список задач? Как можно добавить задачу?
5. Как осуществляется «сворачивание кода»? Какие области сворачиваются?
6. Зачем нужно окно свойств?
7. Зачем нужно окно вывода и окно ошибок? Чем они отличаются?
8. Зачем нужно окно решения?
9. Как настраиваются параметры форматирования текста в окне редактора?
10. Какие существуют команды форматирования? Как можно менять отступы блоков текста?
11. Как добавить в проект новый или существующий файл.

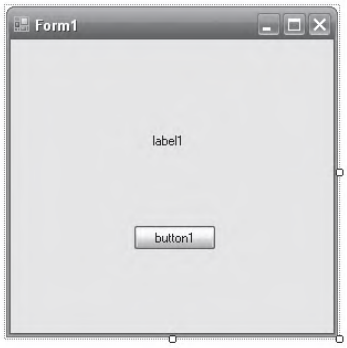
Практическое занятие № 16
Создание интерфейса пользователя в среде разработки программ

Цель работы: научиться создавать простейшие программы с экранной формой и элементами управления.

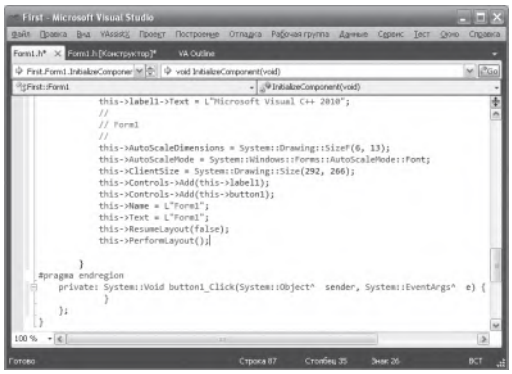
Краткие теоретические сведения



Для создания проекта выбираем File->New Project-> Windows Forms Application Visual C++ вводим имя проекта. Экранная форма — Form1, в которой можно расположить элементы управления. Например поля для ввода текста textBox, командные кнопки button, строчки текста label, которые не могут быть отредактированы пользователем. Визуальное программирование предполагает возможность перетаскивания элементов с помощью мыши из панели элементов Toolbox, где расположены всевозможные элементы управления, в форму. Панель Toolbox предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм. Она дает также доступ почти ко всему, что можно перетащить на визуальный конструктор, используемый для создания форм, XML, схем, классов и т. д.



Добавьте на форму метку label и кнопку button в форму, дважды щелкая на этих элементах на панели Toolbox или перетаскивая мышкой. У каждого объекта есть свойства. Свойств много, их можно увидеть, если щелкнуть правой кнопкой мыши в пределах формы и выбрать в контекстном меню команду Properties, при этом появится панель свойств Properties. Для объекта label1 выберем свойство Text и напишем напротив этого поля «Лабораторная работа № 9» (вместо текста label1). Для объекта button1 также в свойстве Text напишем «Ввод». Объекты не только имеют свойства, но и обрабатываются событиями. Событием, например, является клик на кнопке, загрузка (Load) формы в оперативную память и пр. Управляют событиями с помощью процедуры обработки события. Для получения пустого обработчика события щелчок на командной кнопке в свойствах кнопки button1 кликаем на значке молнии Events (события) и в списке всех возможных событий кнопки button1 выбираем двойным кликом событие Click. После этого мы попадаем на вкладку программного кода Form1.h.



На вкладке Form1.h мы увидим, что управляющая среда Visual C++ 2010 сгенерировала строки программного кода. Например, для свойства Text кнопки button управляющая среда назначила *this->button1->Text = L"Ввод";*

Теперь напишем процедуру обработки события Click на кнопке button1. У нас есть обработчик события button1_Click: *private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) { }*. В фигурных скобках обработчика события напишем: *MessageBox::Show("Это моя программа!");*

Здесь вызывается метод Show объекта MessageBox с текстом «Это моя программа!» Оператор разрешения области действия (::) указывает системе найти метод Show среди методов объекта Messagebox. Теперь нажмем F5 и проверим работоспособность программы.

При работе с формой часто ввод данных организуют через элемент управления textBox . Добавляем в форму текстовое поле textBox. Изменим некоторые свойства элементов управления. Получим пустой обработчик загрузки формы (дважды кликаем по экранной форме). Задаем свойствам формы (к форме обращаемся посредством ссылки this), кнопке button1 и текстовому полю textBox1, метке label1 следующие значения:

```
this->Text = "Извлечение квадратного корня";
button1->Text = "Извлечь корень";
textBox1->Clear(); // Очистка текстового поля
label1->String::Empty;
```

Далее программируем событие button1.

```
private: System:: Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
Single X;
bool s1 = Single::TryParse(textBox1->Text,
System::Globalization::NumberStyles::Number,
System::Globalization::NumberFormatInfo::CurrentInfo, X);
if (s1 == false)
{
label1->Text = "Введите чило";
label1->ForeColor = Color::Red; // - цвет текста
return; }
Single Y = (Single)Math::Sqrt(X);
label1->ForeColor = Color::Black;
label1->Text = String::Format("Корень из {0} равен {1:F5}", X, Y);
}
```

При обработке события click проверяется, введено ли число в текстовом поле. Проверка осуществляется с помощью функции TryParse. Первым параметром метода TryParse является анализируемое поле textBox1->Text. Второй параметр — это разрешаемый для преобразования стиль числа. Третий параметр указывает, на какой основе формируется допустимый формат, CurrentInfo – это на основе текущего языка и региональных параметров. Четвертый параметр возвращает результат преобразования. Функция TryParse возвращает булеву переменную true или false, успешно ли выполнено преобразование. Если пользователь ввел число, то будет выполняться оператор извлечения квадратного корня Math::Sqrt(X). Математические функции Visual Studio 2010 являются методами класса Math. Функция Math::Sqrt(X)возвращает значение типа double (двойной точности с плавающей запятой), которое мы приводим с помощью неявного преобразования (Single) к переменной одинарной точности. Затем формируем строку label1->Text с использованием метода String::Format. Использованный формат ≪Корень из {0} равен {1:F5}≫ означает: взять нулевой выводимый элемент, то есть переменную X, и записать эту переменную вместо фигурных скобок; после чего взять первый

выводимый элемент, то есть Y, и записать его вместо вторых фигурных скобок в формате с фиксированной точкой и пятью десятичными знаками после запятой.

Рассмотрим пример работы с CheckBox (Флажком).

```
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
this->Text = "Флажок CheckBox";
checkBox1->Text = "Полужирный"; checkBox1->Focus();
label1->Text = "Выбери стиль шрифта";
label1->TextAlign = ContentAlignment::MiddleCenter;
label1->Font = gcnew System::Drawing::Font("Courier New",14.0F);
}
private: System::Void checkBox1_CheckedChanged(System::Object^ sender,
System::EventArgs^ e)
{ // Изменение состояния флажка на противоположное
if (checkBox1->Checked == true) label1->Font = gcnew
System::Drawing::Font("Courier New", 14.0F,FontStyle::Bold);
if (checkBox1->Checked == false) label1->Font = gcnew
System::Drawing::Font("Courier New", 14.0F,FontStyle::Regular);
}
```

При обработке события загрузки формы задаем начальные значения некоторых свойств объектов Form1 (посредством ссылки this), label1 и checkBox1. Изменение состояния флажка соответствует событию CheckedChanged. если флажок установлен (то есть содержит «галочку») Checked = true, то для метки label1 устанавливается тот же шрифт Courier New, 14 пунктов, но Bold, то есть полужирный. Если флажок не установлен, то есть checkBox1.Checked = false, то шрифт устанавливается Regular, то есть обычный.

Элемент управления ComboBox служит для отображения вариантов выбора в раскрывающемся списке. Продемонстрируем работу этого элемента управления Из панели Toolbox перетащим в форму два текстовых поля TextBox, кнопку Button, метку Label и комбинированный список ComboBox.

```
private: System:: Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
this->Text = "Калькулятор"; label1->Text = "Равно: ";
button1->Text = "Выбор операции";
comboBox1->Text = "Выбор операции";
array<String^>^ Операции = {"Сложение", "Вычитание",
"Умножение", "Деление", "Очистить"};
comboBox1->Items->AddRange(Операции);
comboBox1->TabIndex = 2;
textBox1->Clear(); textBox1->TabIndex = 0;
textBox2->Clear(); textBox2->TabIndex = 1;
}
private: System::Void comboBox1_SelectedIndexChanged(System::Object^ sender,
System::EventArgs^ e)
{
label1->Text = "Равно: ";
Single x, y, z= 0;
```

```
bool s1 = Single::TryParse(textBox1->Text,
System::Globalization::NumberStyles::Number,
System::Globalization::NumberFormatInfo::CurrentInfo, x);
bool s2 = Single::TryParse(textBox2-
>Text,System::Globalization::NumberStyles::Number,
System::Globalization::NumberFormatInfo::CurrentInfo, y);
if (s1 == false || s2 == false)
{
MessageBox::Show("Следует вводить числа!", "Ошибка",
MessageBoxButtons::OK, MessageBoxIcon::Error);
return;
}
switch (comboBox1->SelectedIndex)
{
case 0: z = x + y; break;
case 1: z = x - y; break;
case 2: z = x * y; break;
case 3: z = x / y; break;
case 4: textBox1->Clear(); textBox2->Clear(); label1->Text = "Равно: ";
 return;
}
label1->Text = String::Format("Равно {0:F5}", Z);
}
private: System::Void button1_Click(System::Object^ sender,System::EventArgs^ e)
{
comboBox1->DroppedDown = true;
}
};
}
```

При обработке события загрузки формы присваиваем начальные значения некоторым свойствам, задаем коллекцию элементов комбинированного списка, задаем табличные индексы TabIndex для текстовых полей и комбинированного списка. Табличный индекс определяет порядок обхода элементов. При обработке события comboBox1_SelectedIndexChanged с помощью функции TryParse проверяем, можно ли текстовые поля преобразовать в число. По умолчанию при инсталляции русифицированной версии Windows разделителем целой и дробной частей числа является запятая. Оператор switch осуществляет множественный выбор арифметической
операции в зависимости от индекса выбранного элемента списка SelectedIndex. Оператор switch передает управление той или иной метке case. Последний оператор в процедуре обработки события изменения индекса выбранного элемента осуществляет формирование строки с помощью метода String::Format для вывода ее на метку label1. Формат {0:F5} означает, что значение переменной z следует выводить по фиксированному формату с пятью знаками после запятой. Последняя процедура обработки события обеспечивает раскрытие комбинированного списка через нажатие на кнопку.

Рассмотрим пример, который позволяет пользователю вводить в текстовое поле
цифры, а также разделитель целой и дробной части числа.

```
String^  razd;
private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
{
this->Text = "Введите число";
razd = Globalization::NumberFormatInfo::CurrentInfo->NumberDecimalSeparator;
}
private: System::Void textBox1_KeyPress(System::Object^ sender,
System::Windows::Forms::KeyPressEventArgs^ e)
{
bool razd_find = false;
if (Char::IsDigit(e->KeyChar) == true) return;
if (e->KeyChar == (char)Keys::Back) return;
if (textBox1->Text->IndexOf(razd) != -1)
razd_find = true;
if (razd_find == true) { e->Handled = true; return; }
if (e->KeyChar.ToString() == razd) return;
e->Handled = true;
}
};
```

Выясняем, что установлено в данной системе в качестве разделителя целой дробной части — точка или запятая. Этот разделитель записываем в строковую переменную razd, которая видна из всех процедур данной программы, поскольку объявлена вне всех процедур. Далее в процедуре обработки события KeyPress разрешаем ввод десятичных цифр и нажатие клавиши Backspace путем обхода с помощью return последнего оператора процедуры e->Handled = true, запрещающего ввод символа в текстовое поле. В данной задаче разрешить ввод разделителя мы можем только один раз, но при этом надо помнить, что пользователь может его удалить и ввести в другом месте числовой строки. Каждый раз при очередном нажатии клавиши, разрешив ввод десятичных цифр, в текстовом поле ищем искомый разделитель. Если он найден, то запрещаем ввод любых нецифровых символов, включая злосчастный разделитель. А если не найден, то разрешаем его ввод.

Содержание работы.

1. Ознакомиться с теоретической частью методических указаний.
2. Получить вариант задания у преподавателя.
3. Решить поставленную задачу с использованием языка программирования C++.
4. Представить работающую программу преподавателю.

Варианты заданий

Разработать программу с использованием экранной формы и элементов управления по варианту.

1. Вычислить периметр и площадь прямоугольного треугольника по длинам a и b двух катетов.

2. Заданы координаты трех вершин треугольника. Найти его периметр и площадь.
3. Вычислить длину окружности и площадь круга одного и того же заданного радиуса *R*.
4. Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
5. Вычислить расстояние между двумя точками с данными координатами.
6. Даны два действительных числа *хну.* Вычислить их сумму, разность, произведение и частное.
7. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
8. Дана сторона равностороннего треугольника. Найти площадь этого треугольника, его высоту, радиусы вписанной и описанной окружностей.
9. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
10. Найти площадь кольца, внутренний радиус которого равен r, а внешний — *R (R> r).*
11. Треугольник задан величинами своих углов и радиусом описанной окружности. Найти стороны треугольника.
12. Найти площадь равнобедренной трапеции с основаниями *a* и *b* и углом а при большем основании *a.*
13. Вычислить корни квадратного уравнения *ax2+ Ьx + c* = 0 с заданными коэффициентами *a, b* и *c* (предполагается, что *a±0* и что дискриминант уравнения неотрицателен).
14. Найти площадь треугольника, две стороны которого равны *a* и b, а угол между этими сторонами *y.*
15. Написать программу, которая выводит на экран первые четыре степени числа л.
16. Найти сумму членов арифметической прогрессии, если известны ее первый член, знаменатель и число членов прогрессии.
17. Найти (в радианах в градусах) все углы треугольника со сторонами *a, b, c.*
18. Составить программу перевода радианной меры угла в градусы, минуты и секунды.
19. Составить программу для вычисления пути, пройденного лодкой, если ее скорость в стоячей воде *v* км/ч, скорость течения реки *v1* км/ч, время движения по озеру t1 ч, а против течения реки — *t2* ч.
20. Текущее показание электронных часов: *m* ч (0 < *m* < 23) *n* мин (0 < *n* < 59) *к* с (0 < *к* < 59). Какое время будут показывать часы через *p ч q* мин *r* с?
21. Вычислить высоты треугольника со сторонами *a, b, c.*
22. Полторы кошки за полтора часа съедают полторы мышки. Сколько мышек съедят А кошек за К часов?
23. Составить программу вычисления объема цилиндра и конуса, которые имеют одинаковую высоту и одинаковый радиус основания.
24. Дана величина *A,* выражающая объем информации в байтах. Перевести *A* в более крупные единицы измерения информации.
25. Окружность вписана в квадрат заданной площади. Найти площадь квадрата, вписанного в эту окружность. Во сколько раз площадь вписанного квадрата меньше площади заданного?

Содержание отчета

1. Цель работы и задание.
2. Пункты, соответствующие порядку выполнения работы.
3. Выводы по работе.

Практическое занятие № 17
Организация классов и принцип инкапсуляции

Цель работы: ознакомление с основными концепциями объектно-ориентированного программирования; изучение классов языка C++, способов их описания и использования, получение представления о перегрузке операторов и функций; получение навыков применения объектов в прикладных программах.

Краткие теоретические сведения

Объектно-ориентированный подход впитал в себя лучшие идеи структурированного и комбинирует их с новыми мощными концепциями, позволяющими увидеть задачу программирования в новом свете. Объектно-ориентированное программирование позволяет легко разложить сложную задачу на подзадачи, взаимодействующие друг с другом. Затем можно преобразовать эти подзадачи в единицы, называемые объектами. Все объектно-ориентированные языки имеют три общие концепции: инкапсуляцию, полиморфизм и наследование.

Инкапсуляция представляет собой механизм, который связывает вместе код и данные и который хранит их от внешнего воздействия и от неправильного использования. Более того, именно инкапсуляция позволяет создавать объекты. Попросту говоря, объект представляет собой логическое целое, включающее в себя данные и код для работы с этими данными. Мы можем определить часть кода и данных как собственность объекта, которая недоступна извне. На этом пути объект обеспечивает существенную защиту против случайной модификации или некорректного использования таких своих частных (*private*) членов. Во всех случаях объект представляет собой переменную, тип которой определяется пользователем. На первый взгляд может показаться странным представлять себе объект, который соединяет в себе вместе код и данные, как переменную. Тем не менее, в объектном программировании дело обстоит именно так. Когда создается объект, неявным образом создается новый тип переменной.

Объектно-ориентированные языки программирования поддерживают полиморфизм, который можно охарактеризовать следующей фразой: «один интерфейс – множество методов». Т.е. полиморфизм представляет собой механизм, который позволяет использовать один и тот же интерфейс при реализации целого набора различных действий. Выбор того, какое именно действие будет совершено, определяется конкретной ситуацией. Например, рассмотрим пример программы, которая определяет три различных типа списков. Один из них используется для целых чисел, другой – для символов, третий – для значений с плавающей запятой. Благодаря полиморфизму, можно создать три набора функций, имеющих одинаковое имя *push()* (поместить) и *pop()* (извлечь) – по одной на каждый тип данных. Общая концепция (интерфейс) заключается в том, чтобы вставлять и извлекать данные в список и из списка. Функции определяют специфические способы (методы), с помощью которых эти операции выполняются для каждого типа данных. Когда информация вставляется в список, автоматически вызывается та версия функции *push(),* которая соответствует типу обрабатываемых данных. Задача выбора специфического действия, т.е. метода, в зависимости от конкретной ситуации возлагается на компилятор.

Наследование представляет собой процесс, благодаря которому один объект может наследовать, приобретать свойства другого объекта. Это свойство поддерживает концепцию классификации, чем и обуславливается его важность. Например, красное яблоко представляет собой часть класса яблоко, который, в свою очередь, представляет собой часть класса фрукт, входящий в больший класс продукты питания. Без использования классификации каждый объект должен был бы определять все свои характеристики явным образом. На основе классификации объект нуждается только в определении таких качеств, которые отличают его от других объектов этого класса. Благодаря механизму наследования объект может характеризоваться в рамках классификации общего и частного.

Механизм объектов реализуется в языке программирования C++ с помощью классов. Общий вид описания класса следующий:

```
class <имя класса> {
    private:
        <частные данные и функции>;
protected:
        <защищенные данные и функции>;
    public:
        <публичные данные и функции>;
} <список объектов>;
```

Класс может содержать как частные (*private*) и защищенные (*protected*), так и публичные (*public*) данные. По умолчанию все члены класса являются частными, т.е. доступ к таким данным и функциям имеют только члены данного класса. Публичные элементы доступны для обращения из других мест программы. Защищенные члены аналогичны частным, отличия касаются только механизмов их наследования, которые будут рассматриваться в следующей лабораторной работе.

Как только определен класс, можно создать объект этого типа, используя имя класса. Фактически имя класса становится спецификатором нового типа данных. Рассмотрим, например, класс, описывающий окружность:

```
class circle {
        int x; // координаты центра
        int y;
        int r; // значение радиуса
    public:
        float get_length(void);
        float get_square(void);
        void set_params(int c_x, int c_y, int c_r);
        friend void func(void);
}
circle one, *two;
// создаем объект окружность one и указатель на объект two
```

Сравните: *int a, *b;*

Внутри объявления класса используются прототипы функций. Для написания реального кода функции-члена компилятору необходимо указать, к какому именно классу относится данная функция:

```
float circle::get_length(void)
{
    return 2*3.1415*r
}

float circle:: get_square(void)
{
    return 3.1415*r*r;
}

void circle:: set_params(int c_x, int c_y, int c_r)
{  x = c_x, y = c_y, r = c_r;
}
```

Последовательность символов :: называется оператором области видимости. Он показывает принадлежность функции конкретному классу. В C++ несколько различных классов могут использовать одинаковые имена функций. Компилятор понимает, какая из них принадлежит какому классу благодаря оператору области видимости и имени класса.

Функция, не являющаяся членом класса, может иметь доступ к его частным членам в случае, если она объявлена дружественной (*friend*) этому классу (см. пример выше). Одна из причин, почему C++ допускает существование таких функций, связана с той ситуацией, когда два класса для определенной цели должны использовать одну и ту же функцию.

Перед использовании объекта может потребоваться инициализировать некоторые его данные. Поскольку требования инициализации являются весьма распространенными, то C++ позволяет производить инициализацию объектов во время их создания. Такая

автоматическая инициализация выполняется с помощью функции, называемой конструктором класса. Функция конструктор, являющаяся членом класса и имеющая имя, совпадающее с именем класса, представляет собой специальный тип функции. Конструктор объекта вызывается автоматически при создании объекта. При инициализации глобальных или статических объектов конструктор вызывается только один раз, для локальных объектов конструктор вызывается каждый раз, когда встречается объявление объекта. Следует иметь в виду, что в C++ конструкторы не могут возвращать значений. Теперь в классе circle функция set_params() может быть заменена конструктором:

```
class circle {
        int x; // координаты центра
        int y;
        int r; // значение радиуса
    public:
        circle(int c_x, int c_y, int c_r);
        ~circle();
        float get_length(void);
        float get_square(void);
        friend void func(void);
}
circle::circle(int c_x, int c_y, int c_r)
  {
        x = c_x; y = c_y; r = c_r;
  }
```

Для передачи аргументов конструктору необходимо задать его значение при объявлении объекта. C++ поддерживает два способа решения этой задачи. В первом вызывается непосредственно с передачей ему значений переменных:

```
circle a = circle(1, 2, 3);
```

Во втором способе аргументы следуют непосредственно за объектом в круглых скобках:

```
circle a(1, 2, 3);
```

Дополнением конструктора является деструктор. Во многих случаях перед уничтожением объекта необходимо выполнить определенные действия. Локальные объекты создаются при входе в блок кода и уничтожаются при выходе из него. Глобальные объекты уничтожаются при завершении работы программы. Имеется много причин, чтобы существовал деструктор. Например, может потребоваться освободить память, которая была ранее зарезервирована. В C++ за дезактивацию отвечает деструктор. Он должен иметь то же имя, что и конструктор только к нему добавляется значок ~: *~circle()*.

Если рассматривать часть программы, которая не входит в состав класса, то для доступа к его членам необходимо использовать следующую конструкцию:

<имя объекта>.<имя члена класса>, если мы работаем непосредственно с объектом или

<имя указателя>->><имя члена класса>, если мы работаем с указателем на объект. Например:

```
circle one, *two;
float a, b;
a = one.get_length();
b = two->get_square();
```

Напомним, что созданные объекты в C++ являются обычными переменными, поэтому они могут передаваться в функции в качестве аргументов, а также служить

типом возвращаемого функцией значения. Помимо этого, возможно создавать массивы объектов и получать указатели на объекты:

```
circle function(circle a, circle* b)
void main(void) {
circle one, *two, three[10], *four, *five, six;
two = &one; four = three; five = &three[2];
six = function(one, five);
}
```

Для реализации полиморфизма в C++ существует механизм, называемый перегрузкой функций. Смысл его заключается в том, что две и более функции могут иметь одинаковое имя, если они отличаются набором параметров в интерфейсе. Компилятор определяет, какую функцию можно использовать в конкретной ситуации, благодаря типу аргумента. По существу перегрузка функций позволяет создавать единое имя для операции, а компилятор устанавливает, какую именно функцию следует использовать в конкретной ситуации для выполнения данной операции. Например, опишем функции для ввода символов, целых и вещественных чисел, которые могут выдавать подсказку пользователю:

```
#include<iostream.h>
void promt(char *str, char *ch)
    {
        cout<<str;
        cin>>*ch;
    }
void promt(char *str, int *i)
    {
        cout<<str;
        cin>>*i;
    }
void promt(char *str; float *f);
    {
        cout<<str;
        cin>>*f;
    }
void main(void)
    {
        char a; int b; float c;
        promt("Enter a char: ", &a);
        promt("Enter an integer: ", &b);
        promt("Enter a float: ", &c);
        cout<<a<<" "<<b<<" "<<c;
    }
```

Другим способом реализации полиморфизма в языке C++ служит перегрузка операторов. Например, можно перегрузить операторы +, = и т.д. для выполнения специфичных действий. В общем случае можно перегружать операторы C++, определяя, что они означают применительно к определенному классу. Можно перегрузить оператор + по отношению к объектам типа <множество> таким образом, что он производит объединение двух произвольных множеств. Другой класс может использовать + совершенно другим образом. Например, для добавления элементов в список. Для того, чтобы перегрузить оператор, необходимо определить, что именно означает оператор по отношению к тому классу, к которому он применяется. Для этого определяется функция-

оператор, задающая действие оператора. Общая форма записи функции оператора для случая, когда она является членом класса, имеет вид:

```
<тип> <имя класса>::operator#(<список аргументов>)
    {
    <действия, выполняемые оператором>;
    }
```

Здесь перегружаемый оператор подставляется вместо символа #, а тип задает тип значений возвращаемых оператором. Для того, чтобы упростить использование перегруженного оператора в сложных выражениях, в качестве возвращаемого значения часто выбирают тот же самый тип, что и класс, для которого перегружен оператор. Например, перегрузим операторы + и = для класса *circle*.

```
class circle {
        int x; // координаты центра
        int y;
        int r; // значение радиуса
    public:
        circle(int c_x, int c_y, int c_r);
        ~circle();
        circle operator+(circle);
        circle operator=(circle);
}
circle::circle(int c_x, int c_y, int c_r)
  {
        x = c_x; y = c_y; r = c_r;
  }
circle circle::operator+(circle a)
// смысл оператора определяет сами – среднее //арифметическое двух окружностей
    {
        circle temp;
        temp.x = (x+a.x)/2;
        temp.y = (y + a.y)/2;
        temp.r = (r + a.r)/2;
        return temp;
    }
circle circle::operator=(circle a)
    {
        x = a.x; y = a.y; r = a.r;
        return *this;
    }
void main(void)
    {
        circle one(1, 1, 1), two(2, 2, 2), three(3, 3, 3);
        one = two + three;
    }
```

Необходимо отметить два момента. Во-первых, обе функции-оператора имеют по одному параметру, хотя и перегружают бинарные операторы + и =. Во-вторых, мы использовали ключевое слово языка C++ *this*. Всякий раз, когда вызывается функция-член класса, в нее неявно передается указатель на объект, вызывающий данную функцию. Чтобы получить доступ к этому указателю, и используется *this*.

Теперь вернемся к рассматриваемому примеру. Во всех случаях именно объект, стоящий слева от знака операции вызывает функцию-оператор. К нему можно обратиться через *this*. Объект, стоящий справа от знака операции, передается функции как аргумент. Поэтому строка *x = a.x* эквивалентна *this->x = a.x*, а *this* в операторе присваивания также используется для возвращения значения.

При перегрузке унарных операций функции-операторы не будут иметь параметров, а при перегрузке бинарных операторов, функции-операторы имеют по одному аргументу.

Порядок выполнения работы

1. Получить задание у преподавателя.
2. Разработать алгоритм решения задачи и написать программу, реализующую задание.
3. Проверить правильность ее работы.
4. Составить отчет и защитить работу.

Варианты заданий

1. Разработать класс, позволяющий выполнять действия над матрицами произвольных размерностей (сложение, вычитание, умножение, транспонирование, нахождение определителя). Интерфейс класса должен включать перегруженные операторы. Предусмотреть возможность загрузки данных из файла.

2. Разработать класс, позволяющий выполнять действия над векторами произвольной длины (сложение, вычитание, скалярное и векторное умножение, нахождение модуля и направляющих косинусов). Интерфейс класса должен включать перегруженные операторы. Предусмотреть возможность загрузки данных из файла.

3. Разработать класс, позволяющий оперировать с комплексными числами (сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, перевод из одной формы записи в другую). Интерфейс класса должен включать перегруженные операторы. Предусмотреть возможность загрузки данных из файла.

4. Разработать класс, позволяющий оперировать с числами произвольной длины (сложение, вычитание, умножение). Интерфейс класса должен включать перегруженные операторы. Длинными называются числа, для операций с которыми нельзя использовать стандартные типы данных. Например, 1234567890987654321. Предусмотреть возможность загрузки данных из файла.

5. Разработать класс, позволяющий оперировать с множествами целых чисел (находить объединение, пересечение, инверсию и мощность множеств). Интерфейс класса должен включать перегруженные операторы. Предусмотреть возможность загрузки данных из файла.

6. Разработать класс, позволяющий хранить и обрабатывать информацию о студентах группы (добавление информации, сортировку по различным критериям, поиск по различным критериям, вывод информации об успеваемости, выбор всех отличников, хорошистов, троечников, выбор всех студентов, имеющих средний балл не ниже заданного). Интерфейс класса должен включать перегруженные операторы. Предусмотреть возможность загрузки данных из файла.

7. Разработать класс для хранения текста произвольного объема. Предусмотреть ввод и поиск информации, добавление и удаление данных в произвольное место текста, объединение нескольких текстов в один и разбиение текста на части. Интерфейс класса должен включать перегруженные операторы. Предусмотреть возможность загрузки данных из файла.

Контрольные вопросы

1. В чем заключается отличие объектно-ориентированного программирования от структурированного? Перечислите достоинства и недостатки объектно-ориентированного подхода.

2. В чем заключается смысл инкапсуляции?

3. В чем заключается смысл полиморфизма?

4. В чем заключается смысл наследования?

5. Классы в С++. Их объявление, описание и использование.

6. Что такое конструкторы и деструкторы? Для чего они используются?

7. Каким образом можно передавать объекты в функции? Может ли функция возвращать объекты?

8. Что такое перегрузка функций? Для чего она используется?

9. Что такое перегрузка операторов? Для чего она используется?

10. Что означает ключевое слово *this* в тексте программы на С++?

11. Объясните работу перегруженного оператора присваивания в примере, предложенном в теоретических положениях. Почему он возвращает значение?

Практическое занятие № 18
Программная реализация принципов наследования и полиморфизма

Цель работы: Ознакомление с механизмом наследования классов и возможностями, которые он предоставляет; получение навыков использования наследования в прикладных программах.

Краткие теоретические сведения

Наследование и полиморфизм являются одними из основных элементов объектно-ориентированного подхода в программировании. С помощью наследования можно создать общий класс, определяющий общие черты совокупности объектов. Этот класс может наследоваться другими более специфическими классами, каждый из которых может добавить свои элементы в описании объекта. Есть еще одна причина, по которой наследованию придается большая важность: с его помощью поддерживается полиморфизм во время выполнения программы. В С++ полиморфизм поддерживается как на этапе выполнения программы, так и на этапе ее компиляции. В качестве примера полиморфизма на этапе компиляции можно указать перегрузку операторов и функций. Но, к сожалению, при ее использовании нельзя решить всех задач, возникающих в объектно-ориентированном программировании. Поэтому в С++ используется также полиморфизм времени исполнения программы, для чего используются производные классы и виртуальные функции.

Класс, который наследуется другим классом, называется базовым классом. Иногда его также называют родительским классом. Класс, выполняющий наследование, называется производным классом или потомком.

Когда один класс наследует другой, то все публичные члены базового класса доступны в производном классе. В противоположность этому частные члены базового класса не доступны внутри производного класса. Если необходимо оставить член частным и вместе с тем позволить использовать его производным классам, то для этих целей и используется ключевое слово *protected*. Защищенный член подобен частному за исключением механизма наследования. При наследовании защищенного члена производный класс также имеет к нему доступ. Таким образом, указав спецификатор доступа *protected*, можно позволить использовать член внутри иерархии классов и запретить доступ к нему извне этой иерархии. Общая форма наследования классов имеет следующий вид:

```
class <имя производного класса>: <доступ> <имя базового класса> {
……
<описание производного класса>;
……
};
```

Здесь доступ определяет, каким способом наследуется базовый класс. Спецификатор доступа может принимать три значения: *private*, *public* и *protected*. В случае, если спецификатор доступа опущен, то по умолчанию подразумевается на его месте спецификатор *public*. При этом все публичные и защищенные члены базового класса становятся соответственно публичными и защищенными членами производного класса. Если спецификатор доступа имеет значение *private*, то все публичные и защищенные члены базового класса становятся частными членами производного класса. И, наконец, если спецификатор доступа принимает значение *protected*, то все публичные и защищенные члены базового класса становятся защищенными членами производного класса. Например:

```
class X {
        protected:
                int i, j;
        public:
                void get_ij() {cin>>i>>j;}
```

```
                void put_ij() {cout<<i<<" "<<j<<endl;}
};
// В классе Y i и j класса X становятся частными членами
class Y: private X {
        protected:
                int k;
        public:
                int get_k() {return k;}
                void make_k() {k = i * j;}
};
/* Класс Z имеет доступ к k класса Y, т.к. он защищенный, но не имеет доступа к i и
j, т.к. они частные */
class Z : public Y {
public:
                void f();
};
void Z::f()
        {
        k = 5; // корректно
                i = j = 10; // не корректно, т.к. нет доступа
        }
```

При использовании производных классов важно представлять себе, каким образом и когда выполняются конструкторы и деструкторы базового и производного классов. Конструкторы вызываются в том же самом порядке, в каком классы следуют друг за другом в иерархии классов. Поскольку базовый класс ничего не знает про свои производные классы, то его инициализация может быть отделена от инициализации производных классов и производится до их создания, так что конструктор базового класса вызывается перед вызовом конструктора производного класса. В противоположность этому деструктор производного класса вызывается перед деструктором базового класса.

Когда базовый класс имеет конструктор с аргументами, производные классы должны явным образом обрабатывать эту ситуацию путем передачи базовому классу необходимых аргументов. Для этого используется расширенная форма конструкторов производных классов, в которые передаются аргументы конструкторам базовых классов:

<порожденный конструктор>(<список аргументов>): <базовый конструктор> (<список аргументов>)

```
        {
                ……
        }
```

Список аргументов, ассоциированный с базовым классом, может состоять из констант, глобальных переменных или параметров конструктора производного класса. Поскольку инициализация объекта происходит во время выполнения программы, можно использовать в качестве аргумента любой идентификатор, определенный в области видимости класса.

```
class X {
        protected:
                int a, b;
        public:
                X(int i, int j) {a = i; b = j;}
};
class Y : public X {
        private:
```

```
                int c;
        public:
                Z(int l, int m, int n): X(l, m) { c = n;}
}
void main(void)
        {
                Z ob(0,1,2);
        }
```

Также один класс может наследовать атрибуты двух и более классов одновременно. Для этого используется список базовых классов, в котором каждый из базовых классов отделен друг от друга запятой. Например:

```
class X {
        protected:
                int a;
public:
        void make_a(int p) { a = p; }
};
class Y {
        protected:
                int b;
public:
        void make_b(int q) { a = q; }
}
class Z: public X, public Y {
public:
        int make_ab() { return a*b; }
}
```

Поскольку класс *Z* наследует оба класса *X* и *Y*, то он имеет доступ к публичным и защищенным данным обоих классов. При множественном наследовании конструкторы базовых классов вызываются слева направо по порядку их следования в описании производного класса. Деструкторы же вызываются в обратном порядке – справа налево.

Имеется одна особенность использования указателей на классы при наследовании. В общем случае указатель одного типа не может указывать на объект другого типа. Из этого правила есть исключение, которое относится только к производным классам. В C++ указатель на базовый класс может указывать на объект производного класса, полученный из этого базового класса. Пусть имеем некий базовый класс *B_class* и его производный класс *D_class*. В C++ любой указатель типа *B_class** может также указывать на объект типа *D_class*. Например, если имеются следующие объявления переменных:

```
B_class *p;
B_class b_ob;
D_class d_ob;
```

то следующие присвоения абсолютно законны:

```
p = &b_ob;
p = &d_ob;
```

Используя указатель *p*, можно получить доступ ко всем членам *d_ob*, которые наследованы от *b_ob*. Однако специфические члены *d_ob* не могут быть получены с использованием указателя *p* (по крайней мере до тех пор, пока не будет осуществлено преобразование типов). Это является следствием того, что указатель «знает» только о членах базового типа и не знает ничего о специфических членах производных типов. Если необходимо получить доступ к элементам производного класса с помощью указателя,

имеющего тип указателя на базовый класс, необходимо воспользоваться приведением типов. Например: *((D_class*)p)->function();*

Хотя указатель, имеющий тип указателя на базовый класс, может использоваться в качестве указателя на производный объект, обратное не имеет место. Это означает, что указатель на производный класс не может использоваться для доступа к объектам базового типа.

Полиморфизм времени исполнения обеспечивается не только за счет использования производных классов, но и применением виртуальных функций. Виртуальная функция – это функция, объявленная с ключевым словом *virtua*l в базовом классе и переопределенная в одном или нескольких производных классах. Виртуальные функции являются особыми функциями, потому что при вызове объекта производного класса с помощью указателя или ссылки на него C++ генерирует код, который определяет во время исполнения программы, какую функцию вызвать, основываясь на типе объекта. Для разных объектов вызываются разные версии одной и той же виртуальной функции. Класс, содержащий одну или более виртуальных функций, называется полиморфным классом. Если функция объявлена как виртуальная, то она остается таковой вне зависимости от количества уровней в иерархии классов, через которые она прошла. Например, рассмотрим классы, содержащие информацию о геометрических фигурах:

```
class figure {
        protected:
                double x, y;
        public:
                void set(double a, double b) { x = a; y = b; }
                virtual double area() { cout<<”No area”; return 0; }
}
class triangle: public figure {
        public:
                double area() { return 0.5*x*y;}
}
class square: public figure {
        public:
                double area() { return x*y; }
}
class circle: public figure {
        protected:
                double r;
        public:
                circle(double a) { r = a;}
                double area() { return 3.14*r*r};
}
void main(void)
        {       double res;
                figure *p; // создание указателя базового типа
                triangle t;     square s;       circle c(10); // объекты производных типов
                p = &t;         p->set(1, 1);
                res = p->area(); // находим площадь треугольника
                p = &s;         p->set(2, 2);
                res = p->area(); // находим площадь четырехугольника
                p = &c;         p->set(20, 30);
                res = p->area(); // находим площадь окружности
        }
```

Ключевым моментом в использовании виртуальных функций для обеспечения полиморфизма времени исполнения служит то, что используется указатель именно на базовый класс. Полиморфизм времени исполнения достигается только при вызове виртуальной функции с использованием указателя или ссылки на базовый класс. Ничего не мешает использовать виртуальные функции, как и любые другие нормальные функции, однако достичь полиморфизма времени исполнения на этом пути не удастся. Также стоит отметить, что виртуальным может быть и деструктор класса, хотя конструктор виртуальным быть не может.

Когда виртуальная функция не переопределена в производном классе, то при вызове ее в объекте производного класса вызывается версия из базового класса. Однако во многих случаях невозможно ввести содержательное определение виртуальной функции в базовом классе. Например, при объявлении класса *figure* в предыдущем примере реализация функции *area()* не несет никакого смысла. Могут быть также такие виртуальные функции, которые обязательно должны быть переопределены в производных классах, без чего эти классы не будут иметь никакого значения. В таких случаях необходим метод, гарантирующий, что производные классы действительно определят все необходимые функции. Язык C++ предлагает в качестве решения этой проблемы чисто виртуальные функции. Это такие функции, которые объявлены в базовом классе, но не имеют в нем определения. Т.к. они не имеют определений, т.е. тел в этом базовом классе, то всякий производный класс обязан иметь свою собственную версию реализации. Для объявления чистой виртуальной функции используется следующая общая форма: virtual <тип возвращаемого значения> <имя функции> (<список параметров>) = 0;

Например, для определения чисто виртуальной функции area() в классе figure необходимо написать следующее: *virtual double area() = 0;*

Если какой-либо класс имеет хотя бы одну чисто виртуальную функцию, то такой класс называется абстрактным. Важной особенностью абстрактных классов является то, что нельзя создать ни одного объекта данного класса. Вместо этого абстрактный класс служит в качестве базового для других производных классов. Причина, по которой абстрактный класс не может быть использован для объявления объектов, заключается в том, что одна или несколько его функций-членов не имеют определения. Тем не менее, даже если базовый класс является абстрактным, все равно можно объявлять указатели или ссылки на него, с помощью которых затем поддерживается полиморфизм времени исполнения.

Порядок выполнения работы

1. Ознакомиться с теоретическими сведениями.
2. Получить вариант задания у преподавателя.
3. Выполнить задание.
4. Продемонстрировать выполнение работы преподавателю.
5. Оформить отчет.
6. Защитить лабораторную работу.

Варианты заданий

1. Разработать иерархию классов, которые позволяют оперировать с геометрическими фигурами: точка, линия, фигура, окружность, треугольник, четырехугольник, многоугольник, пирамида и призма. Реализация должна допускать создание объектов с различными параметрами, вычисление различных геометрических характеристик фигур, сопоставление одноименных фигур друг с другом. Интерфейс должен содержать виртуальные функции.

2. Разработать иерархию классов, которые позволяют создавать картотеку с информацией по предметам различного типа (книги, мебель, бытовая техника, компьютеры, продукты питания и т.д.). Реализация должна допускать добавление информации в картотеку, поиск информации, отбор по различным критериям, сортировку и

хранение наиболее часто просматриваемых карточек. Интерфейс должен содержать виртуальные функции.

3. Разработать иерархию классов, которые позволяют обрабатывать информацию о работниках различного уровня на предприятии (уборщики, рабочие, мастера и инженеры различных подразделений, экономисты, дирекция и т.д.). Реализация должна допускать прием на работу и увольнение сотрудников, их перевод из одного подразделения в другое, начисление ежемесячной зарплаты, учет отпусков, поиск и сортировку информации по различным критериям. Интерфейс должен содержать виртуальные функции.

4. Разработать иерархию классов, позволяющих моделировать работу системного блока персонального компьютера. Реализация должна позволять добавлять различные устройства в ПК, эмулировать их взаимодействие между собой, производить апгрейд аппаратуры. Интерфейс должен содержать виртуальные функции.

5. Разработать иерархию классов, описывающих существующее на рынке программное обеспечение. Реализация должна допускать классификацию ПО по различным критериям, добавление и удаление ПО в базу данных и из нее, выработку аргументированных (по каким критериям осуществлен выбор) рекомендаций пользователю по покупке программ для решения определенных задач. Интерфейс должен содержать виртуальные функции.

Контрольные вопросы

1. Что такое полиморфизм? Как он поддерживается в С++?
2. Что такое наследование классов? Для чего оно применяется?
3. В чем заключаются особенности описания конструкторов и деструкторов при наследовании? В каком порядке они вызываются при создании и удалении объектов производного типа?
4. В чем заключаются особенности использования указателей на базовые и производные классы?
5. Что такое виртуальные функции? Для чего они применяются?
6. Какие Вы знаете стандартные классы языка С++ ?

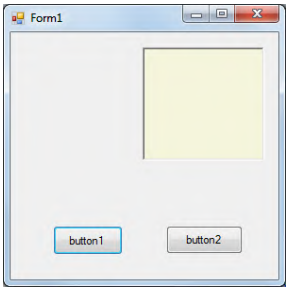
Практическое занятие № 19
Работа с графикой. Рисование графических примитивов
Краткие теоретические сведения

Необходимо спроектировать приложение, интерфейс которого включает форму, элемент вывода графики PictureBox, кнопки button1 и button2.

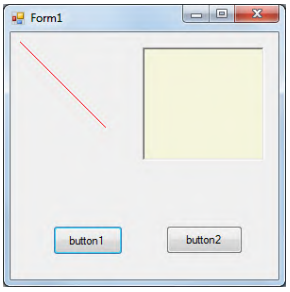
При клике по кнопке button1 на форме рисуется отрезок прямой красного цвета.

При клике по кнопке button2 в элементе PictureBox рисуется отрезок прямой синего цвета.

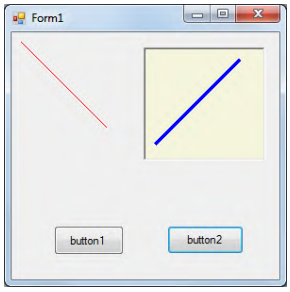
Разработаем интерфейс



а) после запуска



а) после нажатия button1



а) после нажатия button1

Порядок создания элементов управления

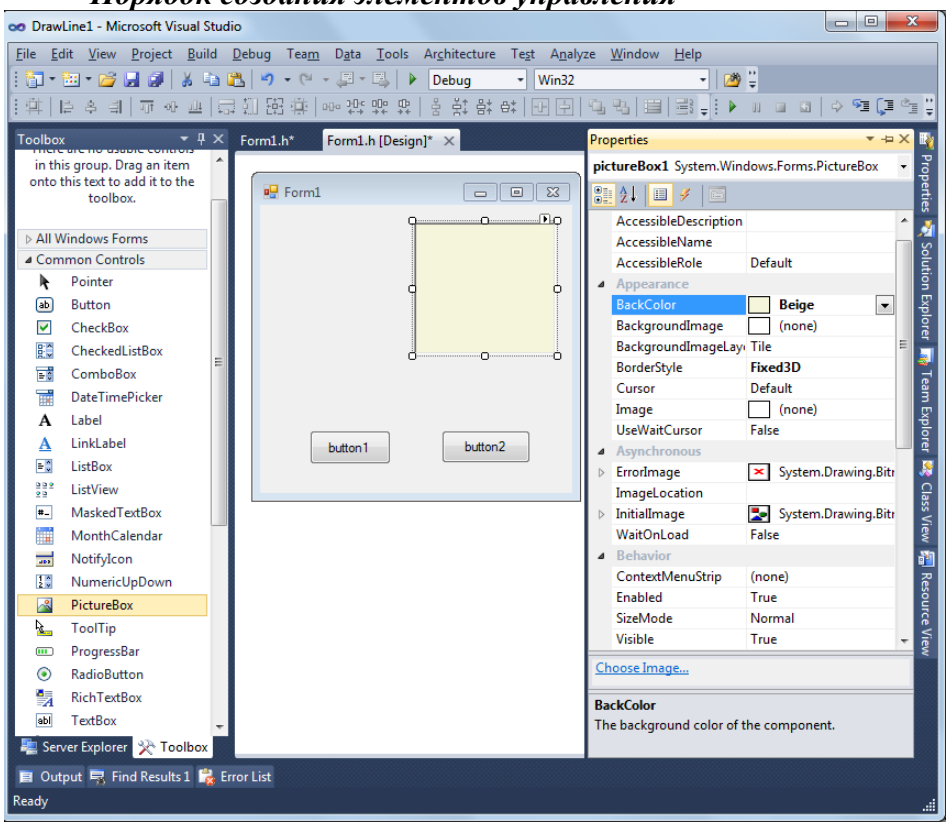


Рис. Окно дизайнера формы

Класс формы

```
#pragma once
namespace DrawLine1 {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;

	public ref class Form1 : public System::Windows::Forms::Form
	{
	public:
		Form1(void)
		{
			InitializeComponent();
			//TODO: Add the constructor code here
		}

	protected:
		~Form1()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::Button^  button1;
	protected:
	private: System::Windows::Forms::Button^  button2;
	private: System::Windows::Forms::PictureBox^  pictureBox1;

	private:
		System::ComponentModel::Container ^components;
```

```
#pragma region Windows Form Designer generated code
            void InitializeComponent(void)
            {
                  this->button1 = (gcnew System::Windows::Forms::Button());
                  this->button2 = (gcnew System::Windows::Forms::Button());
                  this->pictureBox1 = (gcnew
System::Windows::Forms::PictureBox());
                  (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->pictureBox1))->BeginInit();
                  this->SuspendLayout();
                  // button1
                  this->button1->Location = System::Drawing::Point(45, 205);
                  this->button1->Name = L"button1";
                  this->button1->Size = System::Drawing::Size(74, 31);
                  this->button1->TabIndex = 0;
                  this->button1->Text = L"button1";
                  this->button1->UseVisualStyleBackColor = true;
                  this->button1->Click += gcnew System::EventHandler(this,
&Form1::button1_Click);
                  // button2
                  this->button2->Location = System::Drawing::Point(165, 205);
                  this->button2->Name = L"button2";
                  this->button2->Size = System::Drawing::Size(81, 30);
                  this->button2->TabIndex = 1;
                  this->button2->Text = L"button2";
                  this->button2->UseVisualStyleBackColor = true;
                  this->button2->Click += gcnew System::EventHandler(this,
&Form1::button2_Click);
                  // pictureBox1
                  this->pictureBox1->BackColor = System::Drawing::Color::Beige;
                  this->pictureBox1->BorderStyle =
System::Windows::Forms::BorderStyle::Fixed3D;
                  this->pictureBox1->Location = System::Drawing::Point(140, 16);
                  this->pictureBox1->Name = L"pictureBox1";
                  this->pictureBox1->Size = System::Drawing::Size(129, 120);
                  this->pictureBox1->TabIndex = 2;
                  this->pictureBox1->TabStop = false;
                  // Form1
                  this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                  this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
                  this->ClientSize = System::Drawing::Size(284, 262);
                  this->Controls->Add(this->pictureBox1);
                  this->Controls->Add(this->button2);
                  this->Controls->Add(this->button1);
                  this->Name = L"Form1";
                  this->Text = L"Form1";
                  (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->pictureBox1))->EndInit();
                  this->ResumeLayout(false);

            }
#pragma endregion
      private: System::Void button1_Click(System::Object^  sender,
System::EventArgs^  e) {
      Color ^col=gcnew Color;
      Pen ^pen= gcnew Pen(col->Red);
      Graphics ^im  =this->CreateGraphics();
      im->DrawLine(pen,10,10,100,100);
      }
      private: System::Void button2_Click(System::Object^  sender,
System::EventArgs^  e) {
      Color ^col=gcnew Color;
      Pen ^pen= gcnew Pen(col->Blue,4);
      Graphics ^im  =pictureBox1->CreateGraphics();
      im->DrawLine(pen,10,100,100,10);
      }
```

```
};
}
```

Порядок выполнения работы

1. Получить задание у преподавателя.
2. Разработать алгоритм решения задачи и написать программу, реализующую задание.
3. Проверить правильность ее работы.
2. Составить отчет и защитить работу.

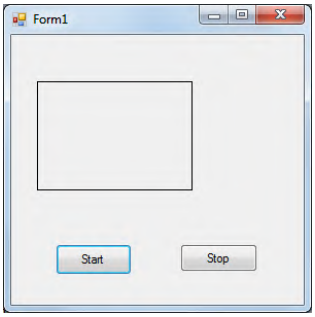
Варианты заданий

1. Спроектировать приложение для рисования отрезка прямой линии на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линии.
2. Спроектировать приложение для рисования эллипса на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линии.
3. Спроектировать приложение для рисования окружности на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линии.
4. Спроектировать приложение для рисования сектора эллипса на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линии.
5. Спроектировать приложение для рисования хорды эллипса на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линии.
6. Спроектировать приложение для рисования под произвольным углом эллипса на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линии, значение угла поворота.
7. Спроектировать приложение для рисования графика функции на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линии графика,таблица значений функции.
8. Спроектировать приложение для рисования изометрии пирамиды на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линий,таблица значений координат вершин пирамиды.
9. Спроектировать приложение для рисования изометрии параллелепипеда на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линий,таблица значений координат вершин параллелепипеда.
10. Спроектировать приложение для рисования изометрии параллелограмма в декартовой системе координат на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода. В поле ввода задаются параметры линий, таблица значений координат вершин параллелограмма.

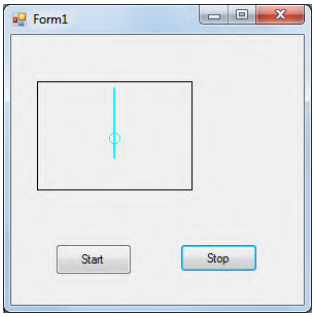
Практическое занятие № 20
Работа с графикой. Рисование анимированных объектов
Краткие теоретические сведения

Спроектируем приложение, интерфейс которого включает форму, кнопку Start, кнопку Stop. При инициализации формы на ней рисуется прямоугольный контур таймера. При клике по кнопке Start в на форме рисуется секундная стрелка синим цветом каждую нечетную секунду и красным цветом – каждую четную.

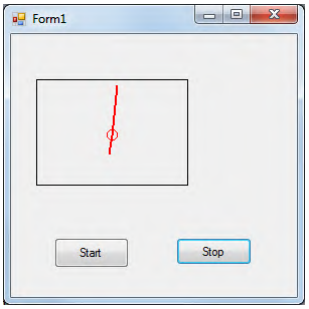
Разработка интерфейса



а) после запуска

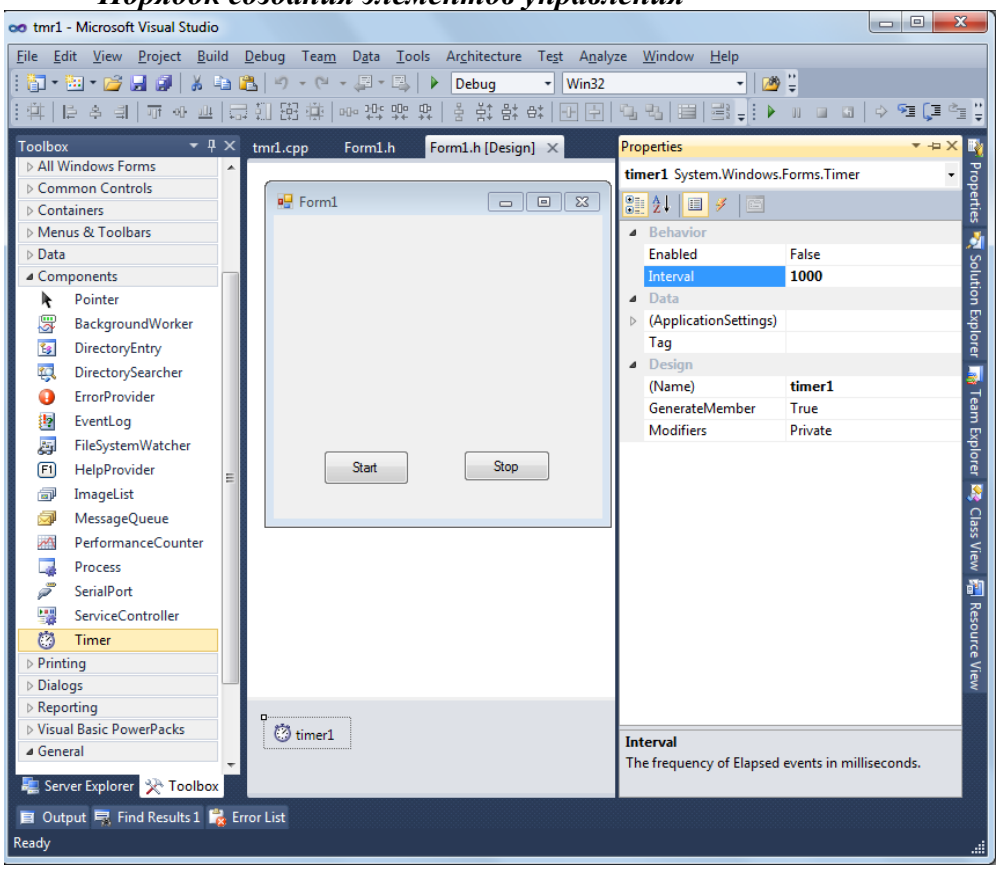


а) после нажатия start



а) после истечениядной секунды

Порядок создания элементов управления



Класс формы

```
#pragma once
#include <math.h>
#include <stdlib.h>
using namespace std;
namespace tmr1 {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;

	public ref class Form1 : public System::Windows::Forms::Form
	{
	public:
	Form1(void)
	{
		InitializeComponent();
		//TODO: Add the constructor code here
		im=this->CreateGraphics();
		col=gcnew Color;
		pen =gcnew Pen(col->Red);
		// Build the rectangles from points and size
		Drawing::Point point1 = Drawing::Point(25,45);
		Drawing::Size size = Drawing::Size(150, 105);
		rect1 = Drawing::Rectangle(point1, size);
```

```
        }
        protected:
        ~Form1()
        {
                if (components)
                {
                delete components;
                }
        }
        private: System::Windows::Forms::Button^  button1;
        protected:
        private: System::Windows::Forms::Button^  button2;
        private: System::Windows::Forms::Timer^  timer1;
        private: System::ComponentModel::IContainer^  components;
        private:
         Color ^col;
         Graphics ^im;
         Pen ^ pen ;
         Drawing::Rectangle rect1;
#pragma region Windows Form Designer generated code
        void InitializeComponent(void)
        {
                this->components = (gcnew System::ComponentModel::Container());
                this->button1 = (gcnew System::Windows::Forms::Button());
                this->button2 = (gcnew System::Windows::Forms::Button());
                this->timer1 = (gcnew System::Windows::Forms::Timer(this->components));
                this->SuspendLayout();
                this->Paint +=
                gcnew System::Windows::Forms::PaintEventHandler(this,
                &Form1::Form1_Paint);
                // button1
                this->button1->Location = System::Drawing::Point(43, 203);
                this->button1->Name = L"button1";
                this->button1->Size = System::Drawing::Size(74, 30);
                this->button1->TabIndex = 0;
                this->button1->Text = L"Start";
                this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnewSystem::EventHandler(this,&Form1::button1_Click);
                // button2
                this->button2->Location = System::Drawing::Point(164, 203);
                this->button2->Name = L"button2";
                this->button2->Size = System::Drawing::Size(75, 27);
                this->button2->TabIndex = 1;
                this->button2->Text = L"Stop";
                this->button2->UseVisualStyleBackColor = true;
                this->button2->Click += gcnew System::EventHandler(this, &Form1::button2_Click);
                // timer1
                this->timer1->Interval = 1000;
                this->timer1->Tick += gcnew System::EventHandler(this, &Form1::timer1_Tick);
                // Form1
                this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
```

```
		this->ClientSize = System::Drawing::Size(284, 262);
		this->Controls->Add(this->button2);
		this->Controls->Add(this->button1);
		this->Name = L"Form1";
		this->Text = L"Form1";
		this->ResumeLayout(false);

	}
#pragma endregion
//----------My code--------------
	private:
	System::Void Form1_Paint(System::Object^ sender,
System::Windows::Forms::PaintEventArgs^ e)
		{
		// Draw a rectangle
		e->Graphics->DrawRectangle(Pens::Black, rect1);
		}

	private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
		if(timer1->Enabled);else timer1->Enabled=true;
		 im->DrawRectangle(Pens::Black, rect1);}
	private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e) {
		 if(!timer1->Enabled);else timer1->Enabled=false; }
	private: System::Void timer1_Tick(System::Object^  sender, System::EventArgs^  e) {
		static float x=100,y=100,fi=0,hfi=3.14159/30,fil,pi=3.14159 ,
		x1=100,y1=100,xl,yl,x1l,y1l,cx=100,
		cy=100,l1=20,l=50,ts=0;
		x1l=x1;y1l=y1;xl=x;yl=y;
	static int t;
	t++;
	y=100-l*sin(fi+pi/2);
	x=100-l*cos(fi+pi/2);
	x1=cx+l1*cos(fi+pi/2);
	y1=cy+l1*sin(fi+pi/2);
	fi=fi+hfi;
	pen->Color=this->BackColor;
	pen->Width=2;
	im->DrawLine(pen,int(xl),int(yl),100,100);
	im->DrawLine(pen,int(x1l),int(y1l),int(cx),int(cy));
	if(t%2)pen->Color=col->Cyan;else pen->Color=col->Red;
	im->DrawLine(pen,int(x1),int(y1),int(cx),int(cy));
	im->DrawLine(pen,int(x),int(y),100,100);
	im->DrawEllipse((pen->Width=1,pen),int(cx-5),int(cy-5),10,10);
		 }
	};
}
```

Порядок выполнения работы

1. Получить задание у преподавателя.
2. Разработать алгоритм решения задачи и написать программу, реализующую задание.

3. Проверить правильность ее работы.
4. Составить отчет и защитить работу.

Варианты заданий

1. Спроектировать приложение для рисования изометрии отрезка прямой линии в трехмерной системе координат на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются также параметры линии.
2. Спроектировать приложение для рисования для рисования изометрии эллипса в трехмерной системе координат на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются также параметры линии.
3. Спроектировать приложение для рисования для рисования изометрии окружности в трехмерной системе координат на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются также параметры линии.
4. Спроектировать приложение для рисования для рисования изометрии сектора эллипса в трехмерной системе координат на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются также параметры линии.
5. Спроектировать приложение для рисования для рисования изометрии и хорды эллипса в трехмерной системе координат на форме. На форме предусмотреть кнопку для ввода данных,кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются также параметры линии.
6. Спроектировать приложение для рисования изометрии графика функции на форме. На форме предусмотреть кнопку для ввода данных, кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот плоскости графика функции осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются параметры линии графика,таблица значений функции.
7. Спроектировать приложение для рисования изометрии пирамиды на форме. На форме предусмотреть кнопку для ввода данных, кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот изображения объекта осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются координаты вершин пирамиды.
8. Спроектировать приложение для рисования изометрии параллелепипеда на форме. На форме предусмотреть кнопку для ввода данных, кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот изображения объекта осуществляется вокруг оси, параметры которой задаются в поле ввода. В поле ввода задаются параметры линий,таблица значений координат вершин параллелепипеда.
9. Спроектировать приложение для рисования изометрии параллелограмма в декартовой системе координат на форме. На форме предусмотреть кнопку для ввода данных, кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот изображения объекта осуществляется вокруг оси, параметры которой задаются в поле ввода В поле ввода задаются параметры линий, таблица значений координат вершин параллелограмма.

10. Спроектировать приложение для рисования изометрии ограниченного параболоида вращения в декартовой системе координат на форме. На форме предусмотреть кнопку для ввода данных, кнопу для рисования, поле ввода, кнопку для однократного поворота объекта. Поворот изображения объекта осуществляется вокруг оси, параметры которой задаются в поле ввода В поле ввода задаются параметры линий, таблица значений параметров параболоида.

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тульский государственный университет»
Технический колледж им. С.И. Мосина

Методические указания
по выполнению курсовых работ

по междисциплинарному курсу «Программные и программно-аппаратные средства защиты информации»

для специальности

10.02.05 Обеспечение информационной безопасности автоматизированных систем

Тула 2024

УТВЕРЖДЕНЫ
Цикловой комиссией информационных технологий
Протокол от «18» 09 2024 г. № 2
Председатель цикловой комиссии И.В. Миляева

СОДЕРЖАНИЕ

ВВЕДЕНИЕ ……4
1 Структура курсовой работы 5
2 Оформление курсовой работы..... 6
3 Оформление текста доклада 12
4 Мультимедийное сопровождение защиты курсовой работы 13
5 Регламент защиты курсовой работы..... 14
6 Критерии оценки докладчика 15
6.1 Выступление..... 15
6.2 Оформление текста курсовой работы..... 16
6.3 Оформление презентации 16
6.4 Ответы на вопросы комиссии..... 17
7 Темы курсовых работ 18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... 20
Основные источники:..... 20
Дополнительные печатные источники:..... 20

ВВЕДЕНИЕ

Курсовая работа – один из основных видов контроля обучаемых, целью которого является проверка полученного уровня знаний по дисциплине, и способность применения их при решении выбранной или поставленной задачи. Оформление отчётных материалов в письменном виде и в форме публичного выступления по результатам исследования.

Курсовая работа является самостоятельной письменной работой студента, под руководством преподавателя, касающаяся какой-то одной достаточно узкой темы или решения конкретной задачи путём применения известных методик. [1]

Основной задачей при подготовке и защите курсовой работы является создание ситуаций максимально приближенной к реальной жизни или подготовки студента к последующей защите выпускной квалификационной работы [2]. Структурно задача делиться на два этапа:

- Самостоятельная подготовка, во время которой студенты самостоятельно определяют круг первоочередных задач, составляют план работы, докладывают о выполнении плана преподавателю, принимают решение о готовности к выступлению.
- Публичная защита или доклад о результатах исследования, ознакомление аудитории с использованными методами решения, для закрепления у обучаемых навыков выступления перед группами лиц.

1 Структура курсовой работы

Курсовая работа выполняется в строгом соответствии с ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления». Должен включать оглавление, введение, одну или несколько глав (2-3), заключение и список использованных источников. [1,3,5]

Во введении должны быть: актуальность темы; цель работы; задачи, которые нужно решить, чтобы достигнуть указанной цели; краткая характеристика структуры (введение, до трёх глав, заключение и библиография); краткая характеристика использованной литературы.

Объем введения - 1-1,5 страницы.

Главы могут делиться на пункты и подпункты в зависимости от этапов решения непосредственной задачи, рекомендуется заканчивать выводами.

В заключении должны быть ответы, на поставленные во введении задачи и дан общий вывод. Объем заключения - 1-1,5 страницы.

Общий объём составляет 10- 24 страницы.

Список использованных источников должен включать не менее 5 (пяти) позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы.

У курсовой работы могут быть приложения - рисунки, схемы, диаграммы, слайды презентации и прочее.

2 Оформление курсовой работы

Текст печатается на белой бумаге формата А4 в книжной ориентации. Используется шрифт: обычный - Times New Roman размером 14 пунктов, интервал 1,5, отступ для абзаца 1,25 см. Цвет шрифта черный. Выбор шрифта и интервала не случаен: Times New Roman – один из наиболее удобных и легких для чтения шрифтов, а полуторный интервал оптимален для восприятия текста. Текст необходимо размещать только на одной стороне листа. Поля оформляются следующим образом: верхнее, нижнее — 20мм, правое — 10 мм, левое поле необходимо для переплета, поэтому оно шире — 30 мм. Нумерация учитывает все страницы, но на титульном листе и на содержании номера страниц не проставляются. На всех остальных листах номер обозначается внизу посередине арабскими цифрами. Если в основном тексте используются формулы, они должны набираться в редакторе формул Microsoft Equation в размере, соответствующем остальному тексту. На рисунке 1 представлен образец настройки параметров страницы.

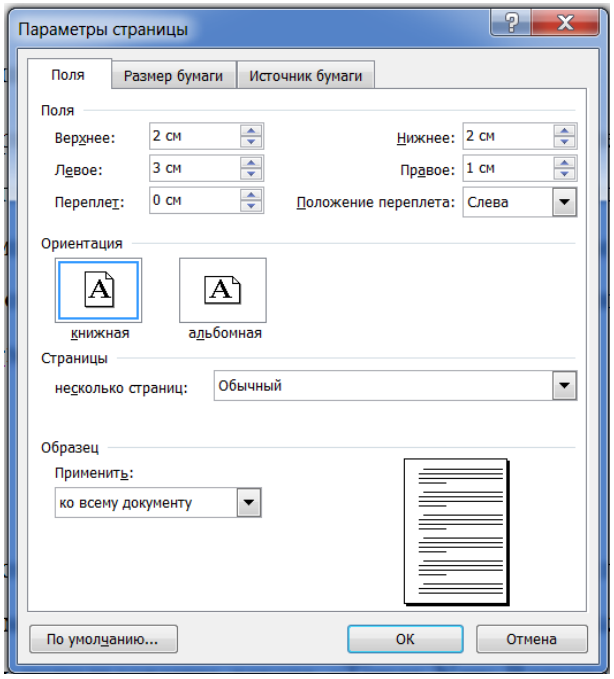


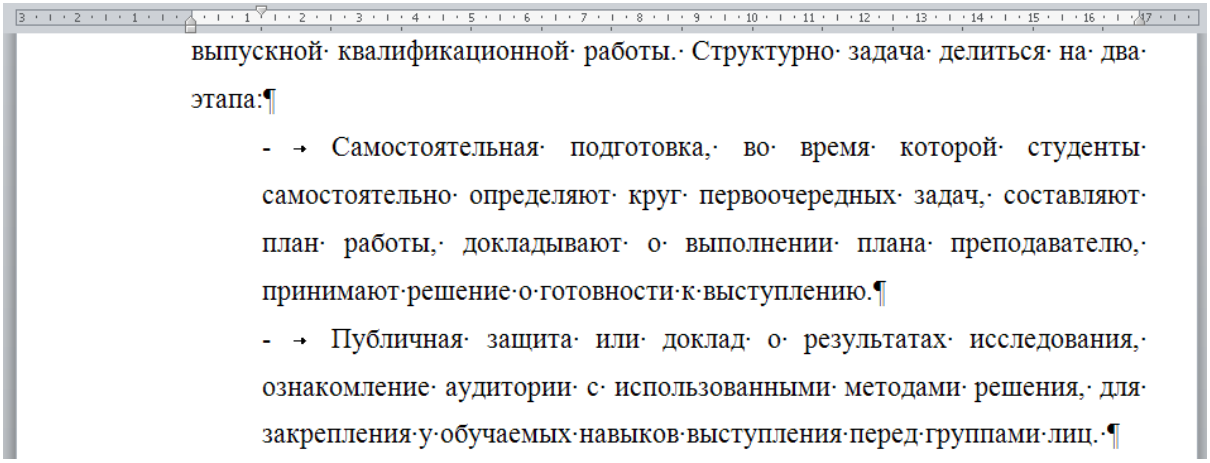
Рисунок 1 - Образец настройки параметров страницы

Допускается использование текста «Times New Roman» с меньшим размером кегля, то есть 8-13 пунктов, при оформлении текста таблиц, пояснительных надписей на рисунках, схемах, диаграммах.

Каждая из частей курсовой работы начинается с новой страницы. Заголовки без нумерации пишутся заглавными буквами и размещаются по центру строки. Заголовки с нумерацией пишется строчными буквами с заглавной, размещается «по ширине страницы» и с отступом красной строки. Между заголовком и последующим текстом оставляется пустая строка.

Главы могут делиться на пункты. Точка после номера не ставится. Номер пункта включает номер соответствующей главы, отделяемый от собственного номера точкой, например: «1.3». Заголовки не должны иметь переносов и подчеркиваний, но допускается выделять их «жирностью» или курсивом. Между заголовком (названием главы) и подзаголовком (названием пункта) оставляется две строки.

Текст размещается с центрированием «по ширине страницы». Абзацы выделяются красной строкой с отступом не менее 1,25-1,27 см. внутри пунктов могут быть перечисления, перед каждой позицией ставиться дефис или при необходимости ссылки в тексте документа на одно из перечислений, строчную букву, после которой ставиться скобка. Пример внешнего вида набора текста показан на рисунке 2.


выпускной· квалификационной· работы.· Структурно· задача· делиться· на· два· этапа:¶

- → Самостоятельная· подготовка,· во· время· которой· студенты· самостоятельно· определяют· круг· первоочередных· задач,· составляют· план· работы,· докладывают· о· выполнении· плана· преподавателю,· принимают·решение·о·готовности·к·выступлению.¶
- → Публичная· защита· или· доклад· о· результатах· исследования,· ознакомление· аудитории· с· использованными· методами· решения,· для· закрепления·у·обучаемых·навыков·выступления·перед·группами·лиц.·¶

Рисунок 2 - Пример внешнего вида набора текста

Нумерация рисунков, таблиц и формул обозначается арабскими цифрами может быть сквозной или в пределах главы. Рисунки (схемы, диаграммы) сопровождаются пояснительными подписями. При этом подпись размещается по центру страницы, сокращение слова рисунок (Рис.) не допускается. Название рисунка следует через дефис. Точка в конце названия не ставиться, если название состоит из 2 и более предложений, то они разделяются точками. Рисунки помещаются после первого упоминания в тексте, или на следующей странице. На все рисунки должны быть ссылки в тексте. Между рисунком и текстом оставляется пустая строка. Например: образец оформления рисунка представлен на рисунке 3.

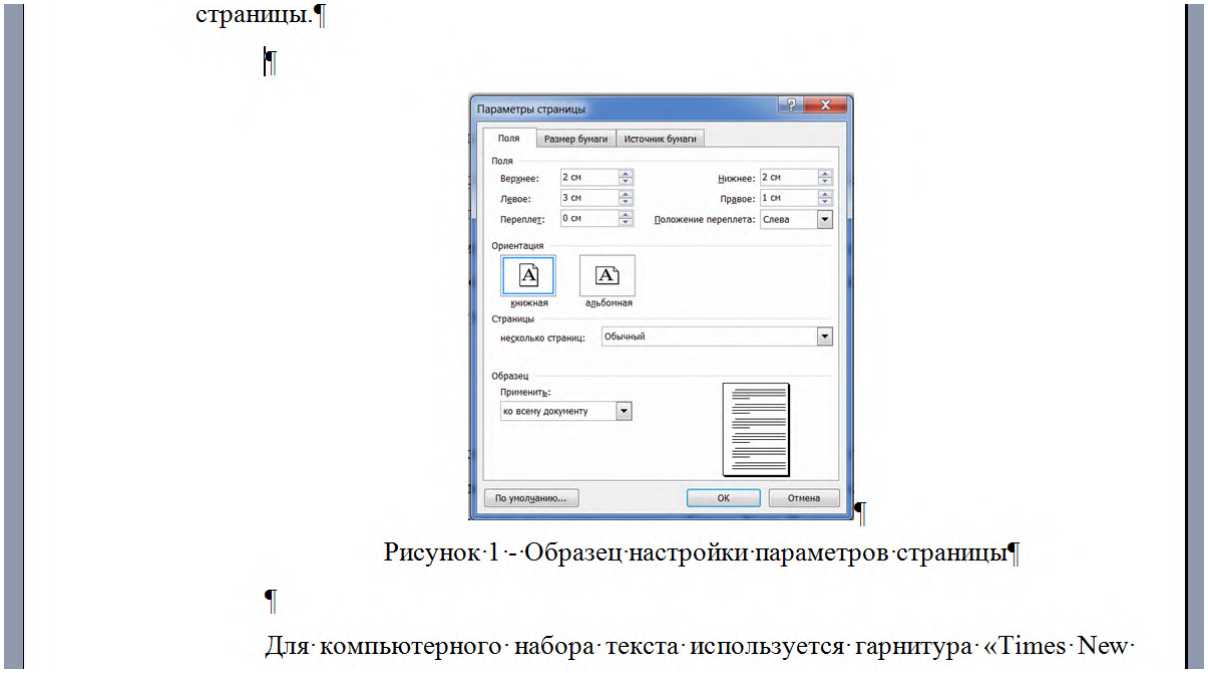


Рисунок 3 – Образец оформления рисунка

Статистический материал рекомендуется оформлять в виде таблицы. Таблицу помещают после первого упоминания в тексте. Над левым верхним углом таблице помещается надпись "Таблица" с указанием ее порядкового номера. Таблицы нумеруются последовательно арабскими цифрами или в пределах главы. Затем следует заголовок таблицы. При ссылке на таблицу указывается ее номер, например: (таблица 1 или таблица 2.3). Таблицы

помещаются после первого упоминания в тексте, или на следующей странице, сокращение слова таблица (Табл.) не допускается. Образец оформления таблицы представлен на рисунке 4.

Результаты·распределения·участников·«Конкурса»·по·рабочим·группам·представлены·в·таблице·5.¶

¶

Таблица·5·-·Распределение·участников·«Конкурса»·по·рабочим·группам¶

№¶ п/п	инженер·-·начальник Отдела·Защиты·Информации	старший·техник·по·программно-аппаратной·защите	техник·по·инженерно-технической·защите	техник
1	Акользина·Ирина·Сергеевна	Королев·Александр·Евгеньевич	Есипов·Евгений·Игоревич	Антошина·Оксана·Александровна
2	Белоголовский·Денис·Сергеевич	Кудинова·Надежда·Андреевна	Арсеньев·Владислав·Александрович	Черковский·Никита·Янович
3	Большаков·Максим·Андреевич	Власова·Юлия·Семёновна	Аминов·Ян·Владиславович	Волкова·Анастасия·Викторовна
4	Власкин·Павел·Сергеевич	Тузов·Михаил·Юрьевич	Фадина·Ольга·Леонидовна	Глибин·Александр·Викторович

¶

Рисунок 4 – Образец оформления таблицы

Математические формулы и зависимости размещаются непосредственно в тексте, нумеруются последовательно арабскими цифрами или в пределах главы. При ссылке на формулу указывается ее номер, например: (формула 1 или формула 2.3).

Обозначения символов используемых в формуле приводиться непосредственно под формулой. При этом используется правила для оформления примечаний, символы располагаются последовательно, текст набирается размером 12 pt, междустрочный интервал равен единице. Формулы следующие одна за другой и не разделённые текстом, разделяют запятой. Образец оформления формулы представлен на рисунке 5.

Для оценки глубины распространения оседающего аэрозоля, образованного линейным источником, используем формулу

$$\Gamma_1 \approx 3{,}5 \cdot 10^{-7} \beta \exp\left\{0{,}2 \ln\left[\frac{2 \cdot 10^6 k_{S2} \cdot G_{ов}}{\beta \cdot l}\right] - \ln \Delta_{доп}\right\}^{0,5}, \qquad (2.3)$$

где: H – высота выливания ОВ, м; $\bar{u}(H)$ – средняя интегральная скорость ветра в слое от по-

Рисунок 5 – Образец оформления формул

Материал, дополняющий текст работы, размещается в приложениях. Приложениями могут быть таблицы, схемы, диаграммы, чертежи, расчеты и т.д. Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ.

Каждое приложение следует начинать с новой страницы. Вверху первой страницы каждого приложения посередине рабочей строки прописными буквами печатают слово «ПРИЛОЖЕНИЕ» и его обозначение. Приложение должно иметь заголовок, который записывают по центру рабочей строки с прописной буквы отдельной строкой.

На все точные числовые данные, прямые цитаты и определения, требуются ссылки на список использованных источников. Обозначаются в тексте реферата в квадратных скобках с указанием номера источника по списку литературы (рисунок 6)

большого фактического материала, глубокого анализа и фундаментальных выводов. [1]

Основной задачей при подготовке реферата является создание ситуаций максимально приближенной, для подготовки студента к последующей защите

Рисунок 6 - Обозначение ссылки на список использованных источников

Список использованных источников для курсовой работы должен включать не менее 5 позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы. Источники указываются в той же последовательности, в которой они располагаются по тексту.

Образец заполнения списка использованных источников представлен на рисунке 7.

Список использованных источников

1. **ГОСТ 7.32-2001** «Отчет о научно-исследовательской работе. Структура и правила оформления» [Текст]. – Введ. 2001-05-22. – М. : Госстандарт России : Изд-во стандартов, 2001.

2. **Желнова О.А.** Основные компетентности подготовки специалистов в системе среднего профессионального образования [Электронный ресурс] : науч. журн. / ISSN 1812-7339, 2008. – Режим доступа : http://www.rae.ru/fs/?section=content&op=show_article&article_id=7781137

3. Порядок представления и правила оформления рукописей статей [Электронный ресурс] : науч. журн. / «Известия ТулГУ», 2012. – Режим доступа : https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CC4QFjAD&url=http%3A%2F%2Fpublishing.tsu.tula.ru%2Fdocs%2Fru

Рисунок 7 - Образец заполнения списка использованных источников

Обратите внимание, что при указании интернет ресурса, обязательно указывается и его название и электронный адрес.

3 Оформление текста доклада

Для и изложения и публичной защиты результатов исследования по теме курсовой работы рекомендуется подготовить устный доклад продолжительностью 5-6 минут. В докладе следует отразить тему и цель исследования, основные задачи, использованные методы и способы их решения, желательно указать источники и авторов обладающих авторитетом в исследуемой области. Раскрыть и обосновать суть работы.

В виду того, что текст доклада является личным вспомогательным материалом докладчика, материал может быть представлен в любом виде (текста, схемы, диаграммы, таблицы и т.д.). Текст доклада формируется из расчета 3 листа формата А4, 16 pt, междустрочный интервал 1, на 5-6 минут доклада, спокойного размеренного монолога.

Текст доклада рекомендуется начинать со следующей формулировки:

Вашему вниманию представляется доклад по теме …..

Задача докладчика, в своём выступлении отразить суть доклада и быть готовым отвечать на возможные вопросы по раскрытой тематике.

По окончании выступления, обучаемый должен подвести вывод, в котором передаётся суть основной мысли доклада.

4 Мультимедийное сопровождение защиты курсовой работы

В виду того, что курсовая работа является отчётным материалом по решению специализированной задачи, при изложении и публичной защите результатов исследования поощряется использование средств мультимедийного сопровождения, в виде слайдов, схем, видеороликов и презентаций. Для более полной наглядности отображения сопутствующих материалов, поощряется использование помощников.

Слайды презентации должны отвечать общим требованиям оформления, но с небольшими поправками, обусловленными незначительным искажением текста и графических объектов при отображении с помощью LCD-проектора.

Текст для оформления слайдов должен отвечать следующим требованиям, используется шрифт «Arial», для заголовков высота 24 pt, для текста слайда 20 pt. Текст располагается с междустрочным интервалам равным единице, распределённым по ширине слайда. Слайды за исключением первого (титульного) нумеруются сквозной нумерацией в правом верхнем углу. Фон и цвет текста подбирается таким образом, что бы исключить затруднение в распознавании текста или графических объектов.

Использование анимации приветствуется в случаях отображения некоторых функциональных особенностей изделий, выделения текста, какой либо области слайда или показе видеофайлов. В других случаях использование анимации нежелательно.

На доклад протяжённостью 5-6 минут, составляется 5-8 слайдов включая титульный, структурно-логическую схему, этапы решения задачи и выводы по работе.

5 Регламент защиты курсовой работы

Перед заслушиванием, преподаватель даёт одну-две минуты докладчику и его помощнику для подготовки к выступлению, запуску и проверке презентации на предмет её готовности к показу, положению трибуны, наличию указки других средств необходимых для проведения доклада.

По мере готовности преподаватель представляет слово докладчику. При этом один из помощников выступающего является секретарём, задача которого заключается в фиксировании времени начала и конца доклада в листе протокола защиты курсовой работы, а так же конспектировании вопросов и ответов по окончании доклада. Второй помощник выступает в качестве оператора и меняет раздаточный материал или слайды презентации.

По окончании доклада, преподаватель предоставляет 1-2 минуты выступающему для отдыха и подготовки к ответу на вопросы, затем открывает дискуссию по теме доклада. Комиссия задаёт по тексту курсовой работы, по слайдам. Как правило одним из основных вопросов является практическая ценность и значимость работы.

Подводя итог защиты, преподаватель обязан дать оценку курсовой работы, и озвучить все недостатки по работе.

6 Критерии оценки докладчика

Критерии оценивания докладчика складываются из требований к оформлению текста курсовой работы, слайдов презентации, качеству представления материала на выступлении, ответов на вопросы по докладу [2].

При этом, тема, цели и задачи доклада должны соответствовать теме выступления. Акцентирование внимания на основных идеях соответствующих целям и задачам. Язык изложения материала понятен аудитории и вызывает интерес у аудитории. Очевидная актуальность, точность и полезность содержания, обосновывается и подтверждается достоверной информацией об исторических справках и текущих событиях.

Использование в презентации графических иллюстраций, статистики, диаграмм и графиков, экспертных оценок, примеров, сравнений, цитат.

Тематическая последовательность, логика и переходы выступления от вступления к основной части, от одной основной идеи (части) к другой, от одного слайда к другому.

В заключении повторение основных целей и задач работы, подведение итогов.

6.1 Выступление

Оценка «отлично» подразумевает свободное знание текста доклада и ориентацию в сопутствующем мультимедийном сопровождении, внятное и чёткое изложение доклада.

Оценка «хорошо» подразумевает знание текста доклада и ориентацию в сопутствующем мультимедийном сопровождении, изложение материала производится чтением доклада.

Оценка «удовлетворительно» подразумевает слабое знание текста доклада, отсутствие мультимедийного сопровождения, изложение материала производится чтением доклада с ошибками, не правильной постановкой ударений.

Оценка «неудовлетворительно» подразумевает не знание материала и отсутствие мультимедийного сопровождения, путанное и нечёткое изложение доклада.

6.2 Оформление текста курсовой работы

Оценка «отлично» подразумевает полное раскрытие темы исследования, приведение примеров из истории или современности, практическую направленность работы, строгое соответствие оформления текста по ГОСТ «Отчёт о НИР».

Оценка «хорошо» подразумевает раскрытие темы исследования, практическую направленность работы, незначительные ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

Оценка «удовлетворительно» подразумевает незначительные отклонения от темы исследования, практическую направленность работы, ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

Оценка «неудовлетворительно» подразумевает выполнение работы не по теме исследования, грубейшие ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

6.3 Оформление презентации

Оценка «отлично» подразумевает наличие презентации соответствующей рекомендациям указанным в пункте 4.

Оценка «хорошо» подразумевает наличие презентации с незначительными отклонениями от рекомендаций указанным в пункте 4.

Оценка «удовлетворительно» подразумевает наличие презентации выполненной не по рекомендациям указанным в пункте 4.

Оценка «неудовлетворительно» отсутствует мультимедийное сопровождение и раздаточный материал.

6.4 Ответы на вопросы комиссии

Оценка «отлично» подразумевает свободное знание материала, быстрые и точные ответы на поставленные вопросы, приведение примеров из истории или современности.

Оценка «хорошо» подразумевает знание материала, правильные ответы на поставленные вопросы.

Оценка «удовлетворительно» подразумевает слабое знание материала, ответы на поставленные вопросы слабые и неуверенные.

Оценка «неудовлетворительно» подразумевает не знание материала отсутствие ответов на вопросы аудитории.

7 Темы курсовых работ

1. Мандатное управление доступом
2. Проверка целостности файлов
3. Программа для аутентификации с помощью usb-устройства
4. Аутентификация с использованием ключа eToken
5. Разграничение доступа к принтерам
6. Разграничение доступа к устройствам. Флеш-накопители
7. Реализация асимметричного шифрования
8. Реализация симметричного шифрования
9. Программа для учета установленного ПО
10. Программа для учета установленного АО
11. Средства обеспечения информационной безопасности банков данных.
12. Разработка программы автоматизированного анализа результатов опросного метода оценки показателей обеспечения информационной безопасности деятельности организации, полученных методом сбора информации анкет (опроса).
13. Использование ЭЦП для обеспечения защиты информации при использовании системы электронного документооборота.
14. Обеспечение защиты конфиденциальной информации в распределённых системах разграничения доступа.
15. Анализ существующих методик оценки экономического ущерба от разглашения (утраты) конфиденциальной информации.
16. Информационная система мониторинга и координации деятельности сотрудников информационно-технического отдела.
17. Инструментальные средства анализа рисков информационной безопасности.
18. Сравнительный и оценочный анализ международных стандартов в области информационной безопасности и управления рисками.

19. Оценочный анализ методов и средств тестирования системы защиты вычислительных сетей (аудита информационной безопасности).
20. Оценка эффективности существующих программных и программно-аппаратных средств защиты информации с применением специализированных инструментов и методов (индивидуальное задание)
21. Обзор и анализ современных программно-аппаратных средств защиты информации (индивидуальное задание)
22. Выбор оптимального средства защиты информации исходя из методических рекомендаций ФСТЭК и имеющихся исходных данных (индивидуальное задание)
23. Применение программно-аппаратных средств защиты информации от различных типов угроз на предприятии (индивидуальное задание)
24. Проблема защиты информации в облачных хранилищах данных и ЦОДах
25. Защита сред виртуализации
26. Средства обеспечения информационной безопасности банков данных.
27. Обеспечения защиты информации при использовании системы электронного документооборота.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основные источники:

1 Программно-аппаратные средства обеспечения информационной безопасности : учебное пособие / А. В. Душкин, О. М. Барсуков, Е. В. Кравцов, К. В. Славнов ; под редакцией А. В. Душкина. — Москва : Горячая линия-Телеком, 2018. — 248 с. — ISBN 978-5-9912-0470-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/111053

2 Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для среднего профессионального образования / О. В. Казарин, А. С. Забабурин. — Москва : Издательство Юрайт, 2020. — 312 с. — (Профессиональное образование). — ISBN 978-5-534-13221-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/449548

3 Бабаш, А.В. Криптографические методы защиты информации : учебник / Бабаш А.В., Баранова Е.К. — Москва : КноРус, 2020. — 189 с. — ISBN 978-5-406-00169-1.—— Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://book.ru/book/933943

4 Васильева, И. Н. Криптографические методы защиты информации : учебник и практикум для вузов / И. Н. Васильева. — Москва : Издательство Юрайт, 2020. — 349 с. — (Высшее образование). — ISBN 978-5-534-02883-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/450998

5 Басалова, Г. В. Основы криптографии : учебное пособие / Г. В. Басалова. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 282 с. — ISBN 978-5-4497-0340-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/89455.html

Дополнительные печатные источники:

1 Джонс, К. Д. Инструментальные средства обеспечения безопасности : учебное пособие / К. Д. Джонс, М. Шема, Б. С. Джонсон. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 913 с. — ISBN 978-5-4497-0871-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/102011.html

2 Никифоров, С. Н. Методы защиты информации. Пароли, скрытие, шифрование : учебное пособие / С. Н. Никифоров. — 3-е изд., стер. — Санкт-Петербург : Лань, 2020. — 124 с. — ISBN 978-5-8114-6352-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/146885

3 Авдошин, С. М. Технологии и продукты Microsoft в обеспечении информационной безопасности : учебное пособие / С. М. Авдошин, А. А. Савельева, В. А. Сердюк. — 3-е изд. — Москва : Интернет-Университет

Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 431 с. — ISBN 978-5-4497-0935-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/102070.html

4 Гельбух, С. С. Сети ЭВМ и телекоммуникации. Архитектура и организация : учебное пособие / С. С. Гельбух. — Санкт-Петербург : Лань, 2019. — 208 с. — ISBN 978-5-8114-3474-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/118646

5 Технологии защиты информации в компьютерных сетях : учебное пособие / Н. А. Руденков, А. В. Пролетарский, Е. В. Смирнова, А. М. Суровов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 368 с. — ISBN 978-5-4497-0931-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/102069.html

6 Казарин, О. В. Надежность и безопасность программного обеспечения : учебное пособие для вузов / О. В. Казарин, И. Б. Шубинский. — Москва : Издательство Юрайт, 2020. — 342 с. — (Высшее образование). — ISBN 978-5-534-05142-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/454453

7 Лапонина, О. Р. Основы сетевой безопасности: криптографические алгоритмы и протоколы взаимодействия : учебное пособие / О. Р. Лапонина ; под редакцией В. А. Сухомлина. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 605 с. — ISBN 978-5-4497-0684-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/97571.html

8 Лось, А. Б. Криптографические методы защиты информации для изучающих компьютерную безопасность : учебник для вузов / А. Б. Лось, А. Ю. Нестеренко, М. И. Рожков. — 2-е изд., испр. — Москва : Издательство Юрайт, 2020. — 473 с. — (Высшее образование). — ISBN 978-5-534-12474-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/450277

9 Запечников, С. В. Криптографические методы защиты информации : учебник для вузов / С. В. Запечников, О. В. Казарин, А. А. Тарасов. — Москва : Издательство Юрайт, 2020. — 309 с. — (Высшее образование). — ISBN 978-5-534-02574-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/450538

10 Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации».

11 Федеральный закон от 27 июля 2006 г. № 152-ФЗ «О персональных данных».

12 Федеральный закон от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании».

13 Федеральный закон от 4 мая 2011 г. № 99-ФЗ «О лицензировании отдельных видов деятельности».

14 Федеральный закон от 30 декабря 2001 г. № 195-ФЗ «Кодекс Российской Федерации об административных правонарушениях».

15 Указ Президента Российской Федерации от 16 августа 2004 г. № 1085 «Вопросы Федеральной службы по техническому и экспортному контролю».

16 Указ Президента Российской Федерации от 6 марта 1997 г. № 188 «Об утверждении перечня сведений конфиденциального характера».

17 Указ Президента Российской Федерации от 17 марта 2008 г. № 351 «О мерах по обеспечению информационной безопасности Российской Федерации при использовании информационно-телекоммуникационных сетей международного информационного обмена».

18 Положение о сертификации средств защиты информации. Утверждено постановлением Правительства Российской Федерации от 26 июня 1995 г. № 608.

19 Состав и содержание организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных. Утверждены приказом ФСТЭК России от 18 февраля 2013 г. № 21.

20 Меры защиты информации в государственных информационных системах. Утверждены ФСТЭК России 11 февраля 2014 г.

21 Административный регламент ФСТЭК России по предоставлению государственной услуги по лицензированию деятельности по технической защите конфиденциальной информации. Утвержден приказом ФСТЭК России от 12 июля 2012 г. № 83.

22 Административный регламент ФСТЭК России по предоставлению государственной услуги по лицензированию деятельности по разработке и производству средств защиты конфиденциальной информации. Утвержден приказом ФСТЭК России от 12 июля 2012 г. № 84.

23 Специальные требования и рекомендации по технической защите конфиденциальной информации (СТР-К). Утверждены приказом Гостехкомиссии России от 30 августа 2002 г. № 282.

24 Требования о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах. Утверждены приказом ФСТЭК России от 11 февраля 2013 г. № 17.

25 Требования о защите информации, содержащейся в информационных системах общего пользования. Утверждены приказами ФСБ России и ФСТЭК России от 31 августа 2010 г. № 416/489.

26 Требования к системам обнаружения вторжений. Утверждены приказом ФСТЭК России от 6 декабря 2011 г. № 638.

27 Руководящий документ. Геоинформационные системы. Защита информации от несанкционированного доступа. Требования по защите информации. Утвержден ФСТЭК России, 2008.

28 Руководящий документ. Защита от несанкционированного доступа к информации. Часть 2. Программное обеспечение базовых систем ввода-вывода персональных электронно-вычислительных машин. Классификация по уровню

контроля отсутствия недекларированных возможностей. Утвержден ФСТЭК России 10 октября 2007 г.

29 Приказ ФАПСИ при Президенте Российской Федерации от 13 июня 2001 г. № 152 «Об утверждении инструкции об организации и обеспечении безопасности хранения, обработки и передачи по каналам связи с использованием средств криптографической защиты информации с ограниченным доступом, не содержащей сведений, составляющих государственную тайну».

30 Приказ ФСБ России от 9 февраля 2005 г. № 66 «Об утверждении Положения о разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации».

31 ГОСТ Р ИСО/МЭК 13335-1-2006 Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий

32 ГОСТ Р ИСО/МЭК ТО 13335-3-2007 Информационная технология. Методы и средства обеспечения безопасности. Часть 3. Методы менеджмента безопасности информационных технологий

33 ГОСТ Р ИСО/МЭК ТО 13335-4-2007 Информационная технология. Методы и средства обеспечения безопасности. Часть 4. Выбор защитных мер

34 ГОСТ Р ИСО/МЭК ТО 13335-5-2006 Информационная технология. Методы и средства обеспечения безопасности. Часть 5. Руководство по менеджменту безопасности сети

35 ГОСТ Р ИСО/МЭК 17799-2005 Информационная технология. Практические правила управления информационной безопасностью

36 ГОСТ Р ИСО/МЭК 15408-1-2008 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель

37 ГОСТ Р ИСО/МЭК 15408-2-2008 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2. Функциональные требования безопасности

38 ГОСТ Р ИСО/МЭК 15408-3-2008 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Требования доверия к безопасности

39 ГОСТ Р 34.10-2001. "Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи"

40 ГОСТ Р 34-11-94. "Информационная технология. Криптографическая защита информации. Функция хэширования"

41 ГОСТ Р 50922-2006 Защита информации. Основные термины и определения. Ростехрегулирование, 2006.

42 ГОСТ Р 52069.0-2013 Защита информации. Система стандартов. Основные положения. Росстандарт, 2013.

43 ГОСТ Р 51583-2014 Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения. Росстандарт, 2014.

44 ГОСТ Р 51624-2000 Защита информации. Автоматизированные системы в защищенном исполнении. Общие требования. Госстандарт России, 2000.

45 ГОСТ Р 51275-2006 Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения. Ростехрегулирование, 2006.

46 ГОСТ Р 52447-2005 Защита информации. Техника защиты информации.
Номенклатура показателей качества. Ростехрегулирование, 2005.

47 ГОСТ Р 50543-93 Конструкции базовые несущие. Средства вычислительной техники. Требования по обеспечению защиты информации и электромагнитной совместимости методом экранирования. Госстандарт России, 1993.

48 ГОСТ Р 56103-2014 Защита информации. Автоматизированные системы в защищенном исполнении. Организация и содержание работ по защите от преднамеренных силовых электромагнитных воздействий. Общие положения. Росстандарт, 2014.

49 ГОСТ Р 56115-2014 Защита информации. Автоматизированные системы в защищенном исполнении. Средства защиты от преднамеренных силовых электромагнитных воздействий. Общие требования. Росстандарт, 2014.

50 ГОСТ Р ИСО/МЭК 15408-1-2012 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. Росстандарт, 2012.

51 ГОСТ Р ИСО/МЭК 15408-2-2013 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2. Функциональные требования безопасности (прямое применение ISO/IEC 15408-2:2008). Росстандарт, 2013.

52 Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных. Утверждена ФСТЭК России 14 февраля 2008 г.

53 Сборник временных методик оценки защищенности конфиденциальной информации от утечки по техническим каналам. УтвержденГостехкомиссией России, 2002.

54 ГОСТ Р 50922-2006 Защита информации. Основные термины и определения. Ростехрегулирование, 2006.

55 ГОСТ Р 51275-2006 Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения. Ростехрегулирование, 2006.

56 Сборник временных методик оценки защищенности конфиденциальной информации от утечки по техническим каналам. УтвержденГостехкомиссией России, 2002.

57 Требования о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах. Утверждены приказом ФСТЭК России от 11 февраля 2013 г. № 17.

58 Меры защиты информации в государственных информационных системах. Утверждены ФСТЭК России 11 февраля 2014 г.

59 Методические рекомендации по технической защите информации, составляющей коммерческую тайну. Утверждены ФСТЭК России 25 декабря 2006 г.

Периодические издания:

1 Системный администратор : [журнал]. - Москва, 2020

2 Программирование : научный журнал / учредители : ФГБОУ ВО МГУ им. М.В.Ломоносова, РАН, Отделение информатики, вычислительной техники и автоматизации РАН. - Москва : Наука, 2020 - . - ISSN 0132-3474. - Текст : электронный // НЭБ eLibrary [сайт]. — URL: https://www.elibrary.ru/title_about_new.asp?id=7966

3 Информационно-управляющие системы : научный журнал / учредитель : ООО «Информационно[управляющие системы». - Санкт-Петербург : Изд-во Санкт-Петербургского государственного университета аэрокосмического приборостроения, 2020 - . - ISSN 1684-8853. - Текст : электронный // НЭБ eLibrary [сайт]. — URL: https://www.elibrary.ru/title_about.asp?id=25785

Интернет-ресурсы:

1. ЭБС Юрайт. - Интернет- ссылка https://urait.ru/
2. ЭБС BOOK.ru. - Интернет- ссылка https://www.book.ru/
3. ЭБС Лань. - Интернет-ссылка https://e.lanbook.com/
4. ЭБС IPRBooks. - Интернет- ссылка http://www.iprbookshop.ru/
5. НЭБ eLibrary. - Интернет-ссылка https://www.elibrary.ru/

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тульский государственный университет»
Технический колледж им. С.И. Мосина

Методические указания
по выполнению курсовых работ

по междисциплинарному курсу «Инженерно-технические средств физической защиты объектов информатизации»

для специальности

10.02.05 Обеспечение информационной безопасности автоматизированных систем

Тула, 2024

УТВЕРЖДЕНЫ

Цикловой комиссий информационных технологий

Протокол от «12» 09 2024 г. № 2

Председатель цикловой комиссии И.В. Миляева

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4
1 Структура курсовой работы ... 5
2 Оформление курсовой работы ... 6
3 Оформление текста доклада ... 12
4 Мультимедийное сопровождение защиты курсовой работы ... 13
5 Регламент защиты курсовой работы ... 14
6 Критерии оценки докладчика ... 15
6.1 Выступление ... 15
6.2 Оформление текста курсовой работы ... 16
6.3 Оформление презентации ... 16
6.4 Ответы на вопросы комиссии ... 17
7 Темы курсовых работ ... 18
ПРИЛОЖЕНИЕ А ... 20

ВВЕДЕНИЕ

Курсовая работа – один из основных видов контроля обучаемых, целью которого является проверка полученного уровня знаний по дисциплине, и способность применения их при решении выбранной или поставленной задачи. Оформление отчётных материалов в письменном виде и в форме публичного выступления по результатам исследования.

Курсовая работа является самостоятельной письменной работой студента, под руководством преподавателя, касающаяся какой-то одной достаточно узкой темы или решения конкретной задачи путём применения известных методик. [1]

Основной задачей при подготовке и защите курсовой работы является создание ситуаций максимально приближенной к реальной жизни или подготовки студента к последующей защите выпускной квалификационной работы [2]. Структурно задача делиться на два этапа:

- Самостоятельная подготовка, во время которой студенты самостоятельно определяют круг первоочередных задач, составляют план работы, докладывают о выполнении плана преподавателю, принимают решение о готовности к выступлению.
- Публичная защита или доклад о результатах исследования, ознакомление аудитории с использованными методами решения, для закрепления у обучаемых навыков выступления перед группами лиц.

1 Структура курсовой работы

Курсовая работа выполняется в строгом соответствии с ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления». Должен включать оглавление, введение, одну или несколько глав (2-3), заключение и список использованных источников. [1,3,5]

Во введении должны быть: актуальность темы; цель работы; задачи, которые нужно решить, чтобы достигнуть указанной цели; краткая характеристика структуры (введение, до трёх глав, заключение и библиография); краткая характеристика использованной литературы.

Объем введения - 1-1,5 страницы.

Главы могут делиться на пункты и подпункты в зависимости от этапов решения непосредственной задачи, рекомендуется заканчивать выводами.

В заключении должны быть ответы, на поставленные во введении задачи и дан общий вывод. Объем заключения - 1-1,5 страницы.

Общий объём составляет 10- 24 страницы.

Список использованных источников должен включать не менее 5 (пяти) позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы.

У курсовой работы могут быть приложения - рисунки, схемы, диаграммы, слайды презентации и прочее.

2 Оформление курсовой работы

Текст печатается на белой бумаге формата А4 в книжной ориентации. Используется шрифт: обычный - Times New Roman размером 14 пунктов, интервал 1,5, отступ для абзаца 1,25 см. Цвет шрифта черный. Выбор шрифта и интервала не случаен: Times New Roman – один из наиболее удобных и легких для чтения шрифтов, а полуторный интервал оптимален для восприятия текста. Текст необходимо размещать только на одной стороне листа. Поля оформляются следующим образом: верхнее, нижнее — 20мм, правое — 10 мм, левое поле необходимо для переплета, поэтому оно шире — 30 мм. Нумерация учитывает все страницы, но на титульном листе и на содержании номера страниц не проставляются. На всех остальных листах номер обозначается внизу посередине арабскими цифрами. Если в основном тексте используются формулы, они должны набираться в редакторе формул Microsoft Equation в размере, соответствующем остальному тексту. На рисунке 1 представлен образец настройки параметров страницы.

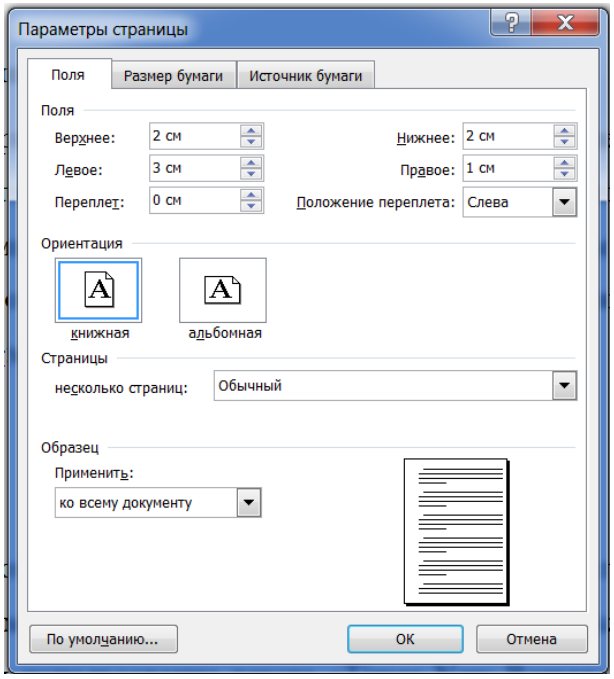


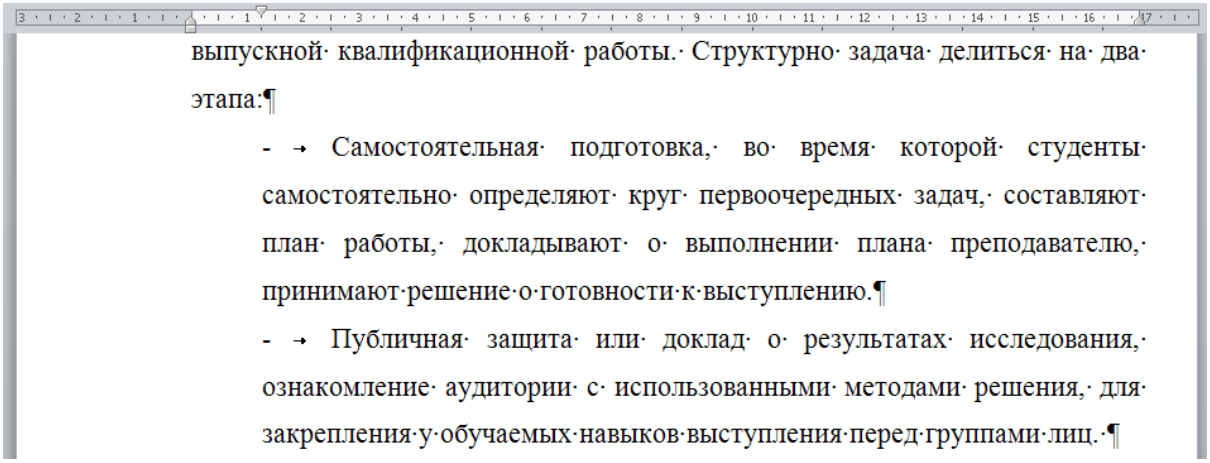
Рисунок 1 - Образец настройки параметров страницы

Допускается использование текста «Times New Roman» с меньшим размером кегля, то есть 8-13 пунктов, при оформлении текста таблиц, пояснительных надписей на рисунках, схемах, диаграммах.

Каждая из частей курсовой работы начинается с новой страницы. Заголовки без нумерации пишутся заглавными буквами и размещаются по центру строки. Заголовки с нумерацией пишется строчными буквами с заглавной, размещается «по ширине страницы» и с отступом красной строки. Между заголовком и последующим текстом оставляется пустая строка.

Главы могут делиться на пункты. Точка после номера не ставится. Номер пункта включает номер соответствующей главы, отделяемый от собственного номера точкой, например: «1.3». Заголовки не должны иметь переносов и подчеркиваний, но допускается выделять их «жирностью» или курсивом. Между заголовком (названием главы) и подзаголовком (названием пункта) оставляется две строки.

Текст размещается с центрированием «по ширине страницы». Абзацы выделяются красной строкой с отступом не менее 1,25-1,27 см. внутри пунктов могут быть перечисления, перед каждой позицией ставиться дефис или при необходимости ссылки в тексте документа на одно из перечислений, строчную букву, после которой ставиться скобка. Пример внешнего вида набора текста показан на рисунке 2.


выпускной· квалификационной· работы.· Структурно· задача· делиться· на· два· этапа:¶

- → Самостоятельная· подготовка,· во· время· которой· студенты· самостоятельно· определяют· круг· первоочередных· задач,· составляют· план· работы,· докладывают· о· выполнении· плана· преподавателю,· принимают·решение·о·готовности·к·выступлению.¶
- → Публичная· защита· или· доклад· о· результатах· исследования,· ознакомление· аудитории· с· использованными· методами· решения,· для· закрепления·у·обучаемых·навыков·выступления·перед·группами·лиц.·¶

Рисунок 2 - Пример внешнего вида набора текста

Нумерация рисунков, таблиц и формул обозначается арабскими цифрами может быть сквозной или в пределах главы. Рисунки (схемы, диаграммы) сопровождаются пояснительными подписями. При этом подпись размещается по центру страницы, сокращение слова рисунок (Рис.) не допускается. Название рисунка следует через дефис. Точка в конце названия не ставиться, если название состоит из 2 и более предложений, то они разделяются точками. Рисунки помещаются после первого упоминания в тексте, или на следующей странице. На все рисунки должны быть ссылки в тексте. Между рисунком и текстом оставляется пустая строка. Например: образец оформления рисунка представлен на рисунке 3.

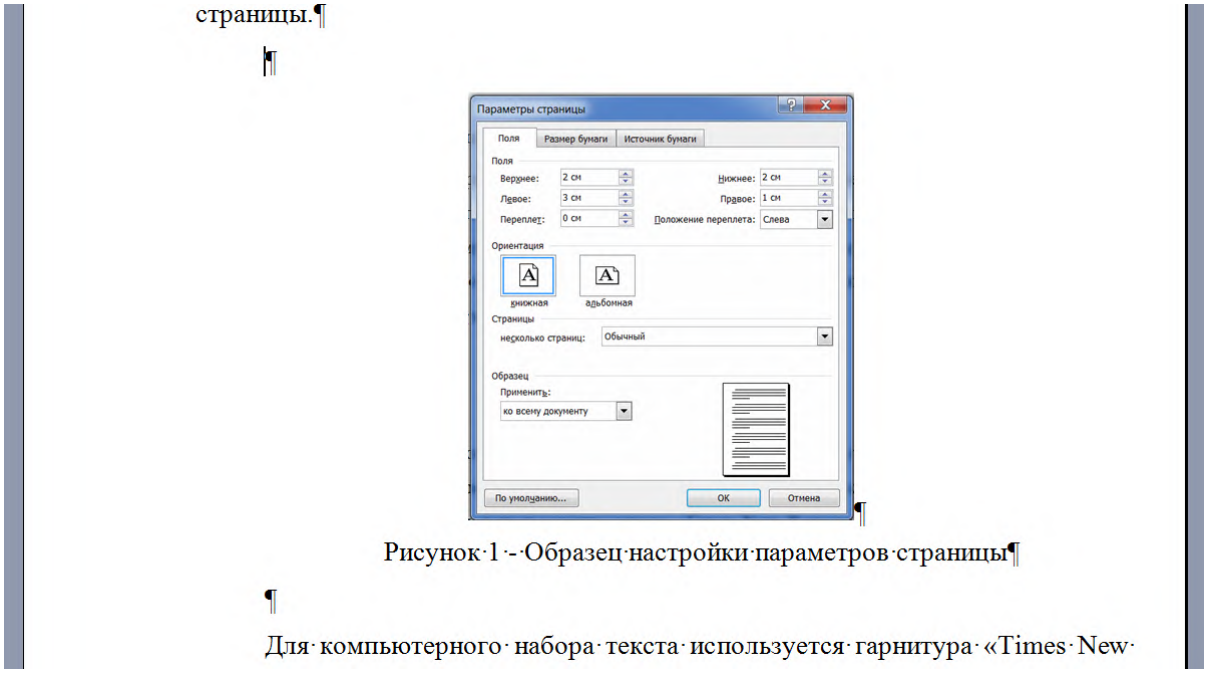


Рисунок 3 – Образец оформления рисунка

Статистический материал рекомендуется оформлять в виде таблицы. Таблицу помещают после первого упоминания в тексте. Над левым верхним углом таблице помещается надпись "Таблица" с указанием ее порядкового номера. Таблицы нумеруются последовательно арабскими цифрами или в пределах главы. Затем следует заголовок таблицы. При ссылке на таблицу указывается ее номер, например: (таблица 1 или таблица 2.3). Таблицы

помещаются после первого упоминания в тексте, или на следующей странице, сокращение слова таблица (Табл.) не допускается. Образец оформления таблицы представлен на рисунке 4.

Результаты распределения участников «Конкурса» по рабочим группам представлены в таблице 5.

Таблица 5 – Распределение участников «Конкурса» по рабочим группам

№ п/п	инженер – начальник Отдела Защиты Информации	старший техник по программно-аппаратной защите	техник по инженерно-технической защите	техник
1	Акользина Ирина Сергеевна	Королев Александр Евгеньевич	Есипов Евгений Игоревич	Антошина Оксана Александровна
2	Белоголовский Денис Сергеевич	Кудинова Надежда Андреевна	Арсеньев Владислав Александрович	Черковский Никита Янович
3	Большаков Максим Андреевич	Власова Юлия Семёновна	Аминов Ян Владиславович	Волкова Анастасия Викторовна
4	Власкин Павел Сергеевич	Тузов Михаил Юрьевич	Фадина Ольга Леонидовна	Глибин Александр Викторович

Рисунок 4 – Образец оформления таблицы

Математические формулы и зависимости размещаются непосредственно в тексте, нумеруются последовательно арабскими цифрами или в пределах главы. При ссылке на формулу указывается ее номер, например: (формула 1 или формула 2.3).

Обозначения символов используемых в формуле приводиться непосредственно под формулой. При этом используется правила для оформления примечаний, символы располагаются последовательно, текст набирается размером 12 pt, междустрочный интервал равен единице. Формулы следующие одна за другой и не разделённые текстом, разделяют запятой. Образец оформления формулы представлен на рисунке 5.

Для оценки глубины распространения оседающего аэрозоля, образованного линейным источником, используем формулу

$$\Gamma_1 \approx 3{,}5 \cdot 10^{-7} \beta \exp\left\{0{,}2 \ln\left[\frac{2 \cdot 10^6 k_{S2} \cdot G_{ов}}{\beta \cdot l}\right] - \ln \Delta_{дст}\right\}^{0{,}5}, \qquad (2.3)$$

где: H – высота выливания ОВ, м; $\bar{u}(H)$ – средняя интегральная скорость ветра в слое от по-

Рисунок 5 – Образец оформления формул

Материал, дополняющий текст работы, размещается в приложениях. Приложениями могут быть таблицы, схемы, диаграммы, чертежи, расчеты и т.д. Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ.

Пример - ПРИЛОЖЕНИЕ А

Каждое приложение следует начинать с новой страницы. Вверху первой страницы каждого приложения посередине рабочей строки прописными буквами печатают слово «ПРИЛОЖЕНИЕ» и его обозначение. Приложение должно иметь заголовок, который записывают по центру рабочей строки с прописной буквы отдельной строкой.

На все точные числовые данные, прямые цитаты и определения, требуются ссылки на список использованных источников. Обозначаются в тексте реферата в квадратных скобках с указанием номера источника по списку литературы (рисунок 6)

большого фактического материала, глубокого анализа и фундаментальных выводов. [1]

Основной задачей при подготовке реферата является создание ситуаций максимально приближенной, для подготовки студента к последующей защите

Рисунок 6 - Обозначение ссылки на список использованных источников

Список использованных источников для курсовой работы должен включать не менее 5 позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы. Источники указываются в той же последовательности, в которой они располагаются по тексту.

Образец заполнения списка использованных источников представлен на рисунке 7.

Список использованных источников

1. **ГОСТ 7.32-2001** «Отчет о научно-исследовательской работе. Структура и правила оформления» [Текст]. – Введ. 2001-05-22. – М. : Госстандарт России : Изд-во стандартов, 2001.

2. **Желнова О.А.** Основные компетентности подготовки специалистов в системе среднего профессионального образования [Электронный ресурс] : науч. журн. / ISSN 1812-7339, 2008. – Режим доступа : http://www.rae.ru/fs/?section=content&op=show_article&article_id=7781137

3. Порядок представления и правила оформления рукописей статей [Электронный ресурс] : науч. журн. / «Известия ТулГУ», 2012. – Режим доступа : https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CC4QFjAD&url=http%3A%2F%2Fpublishing.tsu.tula.ru%2Fdocs%2Fru

Рисунок 7 - Образец заполнения списка использованных источников

Обратите внимание, что при указании интернет ресурса, обязательно указывается и его название и электронный адрес.

3 Оформление текста доклада

Для и изложения и публичной защиты результатов исследования по теме курсовой работы рекомендуется подготовить устный доклад продолжительностью 5-6 минут. В докладе следует отразить тему и цель исследования, основные задачи, использованные методы и способы их решения, желательно указать источники и авторов обладающих авторитетом в исследуемой области. Раскрыть и обосновать суть работы.

В виду того, что текст доклада является личным вспомогательным материалом докладчика, материал может быть представлен в любом виде (текста, схемы, диаграммы, таблицы и т.д.). Текст доклада формируется из расчета 3 листа формата А4, 16 pt, междустрочный интервал 1, на 5-6 минут доклада, спокойного размеренного монолога.

Текст доклада рекомендуется начинать со следующей формулировки:

Вашему вниманию представляется доклад по теме …..

Задача докладчика, в своём выступлении отразить суть доклада и быть готовым отвечать на возможные вопросы по раскрытой тематике.

По окончании выступления, обучаемый должен подвести вывод, в котором передаётся суть основной мысли доклада.

4 Мультимедийное сопровождение защиты курсовой работы

В виду того, что курсовая работа является отчётным материалом по решению специализированной задачи, при изложении и публичной защите результатов исследования поощряется использование средств мультимедийного сопровождения, в виде слайдов, схем, видеороликов и презентаций. Для более полной наглядности отображения сопутствующих материалов, поощряется использование помощников.

Слайды презентации должны отвечать общим требованиям оформления, но с небольшими поправками, обусловленными незначительным искажением текста и графических объектов при отображении с помощью LCD-проектора.

Текст для оформления слайдов должен отвечать следующим требованиям, используется шрифт «Arial», для заголовков высота 24 pt, для текста слайда 20 pt. Текст располагается с междустрочным интервалам равным единице, распределённым по ширине слайда. Слайды за исключением первого (титульного) нумеруются сквозной нумерацией в правом верхнем углу. Фон и цвет текста подбирается таким образом, что бы исключить затруднение в распознавании текста или графических объектов.

Использование анимации приветствуется в случаях отображения некоторых функциональных особенностей изделий, выделения текста, какой либо области слайда или показе видеофайлов. В других случаях использование анимации нежелательно.

На доклад протяжённостью 5-6 минут, составляется 5-8 слайдов включая титульный, структурно-логическую схему, решения задачи и выводы по работе.

5 Регламент защиты курсовой работы

Перед заслушиванием, преподаватель даёт одну-две минуты докладчику и его помощнику для подготовки к выступлению, запуску и проверке презентации на предмет её готовности к показу, положению трибуны, наличию указки других средств необходимых для проведения доклада.

По мере готовности преподаватель представляет слово докладчику. При этом один из помощников выступающего является секретарём, задача которого заключается в фиксировании времени начала и конца доклада в листе протокола защиты курсовой работы, а так же конспектировании вопросов и ответов по окончании доклада. Второй помощник выступает в качестве оператора и меняет раздаточный материал или слайды презентации.

По окончании доклада, преподаватель предоставляет 1-2 минуты выступающему для отдыха и подготовки к ответу на вопросы, затем открывает дискуссию по теме доклада. Комиссия задаёт по тексту курсовой работы, по слайдам. Как правило одним из основных вопросов является практическая ценность и значимость работы.

Подводя итог защиты, преподаватель обязан дать оценку курсовой работы, и озвучить все недостатки по работе.

6 Критерии оценки докладчика

Критерии оценивания докладчика складываются из требований к оформлению текста курсовой работы, слайдов презентации, качеству представления материала на выступлении, ответов на вопросы по докладу [2].

При этом, тема, цели и задачи доклада должны соответствовать теме выступления. Акцентирование внимания на основных идеях соответствующих целям и задачам. Язык изложения материала понятен аудитории и вызывает интерес у аудитории. Очевидная актуальность, точность и полезность содержания, обосновывается и подтверждается достоверной информацией об исторических справках и текущих событиях.

Использование в презентации графических иллюстраций, статистики, диаграмм и графиков, экспертных оценок, примеров, сравнений, цитат.

Тематическая последовательность, логика и переходы выступления от вступления к основной части, от одной основной идеи (части) к другой, от одного слайда к другому.

В заключении повторение основных целей и задач работы, подведение итогов.

6.1 Выступление

Оценка «отлично» подразумевает свободное знание текста доклада и ориентацию в сопутствующем мультимедийном сопровождении, внятное и чёткое изложение доклада.

Оценка «хорошо» подразумевает знание текста доклада и ориентацию в сопутствующем мультимедийном сопровождении, изложение материала производится чтением доклада.

Оценка «удовлетворительно» подразумевает слабое знание текста доклада, отсутствие мультимедийного сопровождения, изложение материала производится чтением доклада с ошибками, не правильной постановкой ударений.

Оценка «неудовлетворительно» подразумевает не знание материала и отсутствие мультимедийного сопровождения, путанное и нечёткое изложение доклада.

6.2 Оформление текста курсовой работы

Оценка «отлично» подразумевает полное раскрытие темы исследования, приведение примеров из истории или современности, практическую направленность работы, строгое соответствие оформления текста по ГОСТ «Отчёт о НИР».

Оценка «хорошо» подразумевает раскрытие темы исследования, практическую направленность работы, незначительные ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

Оценка «удовлетворительно» подразумевает незначительные отклонения от темы исследования, практическую направленность работы, ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

Оценка «неудовлетворительно» подразумевает выполнение работы не по теме исследования, грубейшие ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

6.3 Оформление презентации

Оценка «отлично» подразумевает наличие презентации соответствующей рекомендациям указанным в пункте 4.

Оценка «хорошо» подразумевает наличие презентации с незначительными отклонениями от рекомендаций указанным в пункте 4.

Оценка «удовлетворительно» подразумевает наличие презентации выполненной не по рекомендациям указанным в пункте 4.

Оценка «неудовлетворительно» отсутствует мультимедийное сопровождение и раздаточный материал.

6.4 Ответы на вопросы комиссии

Оценка «отлично» подразумевает свободное знание материала, быстрые и точные ответы на поставленные вопросы, приведение примеров из истории или современности.

Оценка «хорошо» подразумевает знание материала, правильные ответы на поставленные вопросы.

Оценка «удовлетворительно» подразумевает слабое знание материала, ответы на поставленные вопросы слабые и неуверенные.

Оценка «неудовлетворительно» подразумевает не знание материала отсутствие ответов на вопросы аудитории.

7 Темы курсовых работ

1. Разработка системы технической защиты информации типового объекта
2. Организация инженерно–технической защиты объекта
3. Комплексная защита информация
4. Создание комплексной системы защиты информации объекта
5. Угрозы информации и информационным системам
6. Методы и средства защиты информации от несанкционированного доступа
7. Демаскирующие признаки объектов защиты информации
8. Теория и методология защиты информации в офисе
9. Обоснование необходимости защиты информации на объекте
10. Защита информации от несанкционированного доступа
11. Разработка защиты кабинета руководителя
12. Построение инженерно-технической защиты объекта
13. Инженерно-техническая защита макро-биосистем
14. Расчет основных показателей качества системы охранной сигнализации объекта информатизации.
15. Выбор варианта структуры построения системы сбора и обработки информации объекта информатизации.
16. Построение системы обеспечения безопасности объекта информатизации с заданными показателями качества.
17. Инструментальные средства анализа рисков информационной безопасности.
18. Сертификация и аудит объекта информатизации: организационные аспекты
19. Методика проведения аудита объекта информатизации. Этапы проведения аудита
20. Варианты аудита безопасности объекта информатизации.
21. Методы тестирования системы защиты объекта информатизации
22. Средства анализа защищенности объекта информатизации

Список использованных источников

1. ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления» [Текст]. . – Введ. 2001-05-22. – М. : Госстандарт России : Изд-во стандартов, 2001.

2. Желнова О.А. Основные компетентности подготовки специалистов в системе среднего профессионального образования [Электронный ресурс] : науч. журн. / ISSN 1812-7339, 2008. – Режим доступа : http://www.rae.ru/fs/?section=content&op=show_article&article_id=7781137

3. Порядок представления и правила оформления рукописей статей [Электронный ресурс] : науч. журн. / «Известия ТулГУ», 2012. – Режим доступа : https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CC4QFjAD&url=http%3A%2F%2Fpublishing.tsu.tula.ru%2Fdocs%2Frules.doc&ei=Ed7DVI3RFcacygOm44HgDg&usg=AFQjCNHbuChu7rkpBMKN0BqewcUwM8WnGA&bvm=bv.84349003,d.bGQ&cad=rjt

4. Библиотека электронных ресурсов Национального открытого университета ИНТУИТ [Электронный ресурс] / Ред. Д.Скрипник. – М., 2015. – Режим доступа : http://www.intuit.ru/studies/courses/697/553/lecture/12442

5. Правила оформления научных работ [Электронный ресурс] / Рязанский государственный агротехнический университет. – Р., 2015. – Режим доступа : http://rgatu.ru/index.php?option=com_content&view=category&layout=blog&id=477&Itemid=171

ПРИЛОЖЕНИЕ А

Пример оформления титульного листа

Пример оформления титульного листа, представлен на рисунке А.1.

Министерство образования и науки Российской Федерации
ФГБОУ ВПО
«Тульский государственный университет»
Технический колледж имени С.И. Мосина

Цикловая комиссия
«Информационной Безопасности автоматизированных систем»

К У Р С О В А Я Р А Б О Т А

на тему: «Организация инженерно – технической защиты кабинета главного бухгалтера ООО «Гефест»

Выполнил: студент Сидоров С.А. (4-090305)

Проверил: к.т.н., доцент Симаков А.Ю.

Тула 2014

Рисунок А.1 – Образец оформления титульного листа

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тульский государственный университет»
Технический колледж имени С.И. Мосина

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторно-практических и самостоятельных работ

по междисциплинарному курсу «Криптографические средства защиты информации»

для специальности
10.02.05 Обеспечение информационной безопасности автоматизированных систем

Тула 2024

УТВЕРЖДЕНЫ
Цикловой комиссией информационных технологий
Протокол от «12» 09 2024 г. № 2
Председатель цикловой комиссии И.В. Миляева

Тема: Применение алгоритма Евклида для нахождения НОД.

1 Теоретическая часть

Определение 1. Пусть a, b — целые числа. Говорят, что a кратно b, или a делится на b, или b делит a, если существует такое целое число c, что a = bc.

Определение 2. Наибольшим общим делителем (НОД) целых чисел a_1, a_2,..., a_n при n > 2 называется любое целое число d, удовлетворяющее условиям: $d|a_1,a_2,...,a_n$, для любого $d_1 \in Z$ из условия $d_1|a_1,a_2,...,a_n$ следует, что $d_1|d$.

Таким образом, наибольшим общим делителем нескольких целых чисел является такой их общий делитель, который делится на любой другой их общий делитель.

Определение 3. Разделить целое число *a* на целое число *b* с остатком — это значит найти такие целые числа *q* и *r*, которые удовлетворяют условиям:

1) $a = bq + r$,

2) $0<r<|b|$.

Числа *q* и *r* называются соответственно неполным частным и остатком от деления *a* на *b*.

Теорема 2. Если $a,b \in Z$ и $b \neq 0$, то *a* на *b* можно разделить с остатком, при этом неполное частное и остаток находятся однозначно.

Опишем алгоритм Евклида для нахождения НОД двух чисел *a, b,* из которых хотя бы одно отлично от *0*. Пусть, наЗадача $b \neq 0$. Тогда, разделив *a* на *b* с остатком, получим: $a = bq_1 + r_1$.

Если $r_1 = 0$, то процесс будем считать законченным. В этом случае, очевидно, (a, b) = |b|. Если же $r_1 \neq 0$, то разделим *b* с остатком на r_1. Получим остаток r_2.. Если $r_2 = 0$, то процесс считаем законченным. В противном случае делим с остатком остаток r_1 на r_2. Будем продолжать этот процесс деления с остатком до тех пор, пока не получим остаток, равный нулю. Такой момент обязательно наступит, поскольку остатки неотрицательны и убывают:

$$|b| > r_1 > r_2 > \dots .$$

Весь этот процесс деления чисел с остатком и называют алгоритмом Евклида. Символически его можно записать в следующем виде

$$\begin{aligned}
&a = bq_1 + r_1, && 0 < r_1 < |b|,\\
&b = r_1q_2 + r_2, && 0 < r_2 < r_1,\\
&r_1 = r_2q_3 + r_3, && 0 < r_3 < r_2,\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\\
&r_{n-3} = r_{n-2}q_{n-1} + r_{n-1}, && 0 < r_{n-1} < r_{n-2},\\
&r_{n-2} = r_{n-1}q_n + r_n, && 0 < r_n < r_{n-1},\\
&r_{n-1} = r_nq_{n+1}.
\end{aligned} \qquad (3)$$

Прослеживая систему равенств (3) снизу вверх, нетрудно заметить, что остаток r_n делит числа r_{n-1}, r_{n-2}, ..., r_1, *b, a*. Следовательно, r_n — общий делитель чисел *a, b*. Если *d* — какой-либо другой их общий делитель, то, прослеживая

систему (1) сверху вниз, получим $d|r_1,\dots,r_n$. Значит r_n — наибольший общий делитель чисел a, b.

Теорема 1. Пусть (3) есть результат применения алгоритма Евклида к числам a,b . Тогда

$$r_k = au_k + bv_k,\ k = 1,\dots,n,$$

где u_k, v_k – целые числа определяемые рекуррентными формулами

$$u_k=u_{k-2} - u_{k-1}q_k,\ v_k= v_{k-2} - v_{k-1}q_k$$

и начальными условиями $u_0=0,\ v_0=1,\ u_1=1,\ v_1= -q_1$

Определение 4. Целые числа *a, b* называются взаимно простыми, если их неотрицательный НОД равен 1, т. е. $(a, b) = 1$.

Определение 5. Наименьшим общим кратным (НОК) целых чисел $a_1,a_2,\dots,a_n$ при $n \geq 2$ называется любое целое число m, удовлетворяющее условиям:

$a_1,a_2,\dots,a_n \mid m$,

для любого $m_1 \in Z$ из условия $a_1,a_2,\dots,a_n \mid m_1$ следует, что $m|m_1$.

Теорема 2. Если *a, b* — отличные от нуля целые числа и $(a, b) = d$, то число $m = ab/d$ есть их НОК.

2 Практическая часть

Задание 1. Найти наибольший делитель $d=(a,b)$ и наименьшее общее кратное $m=[a,b]$ целых чисел a,b с помощью алгоритма Евклида, а также целые числа u,v, удовлетворяющие условию $au + bv = d$.

Задача

a = 12103; b = 1425

Решение:

1. Применим алгоритм Евклида к числам $a = 12103$; $b = 1425$:

$$12103 = 1425*8 + 703$$

$$1425 = 703*2 + 19$$

$$703 = 19*37 + 0$$

Таким образом, получили, что найти наибольший делитель

$$d = (12103, 1425) = 19$$

2 Найдем наименьшее общее кратное m=[12103, 1425] по формуле

$$m = \frac{ab}{d};$$

$$m = \frac{12103 \quad * 1425}{19} = 907725$$

3 Для нахождения целых чисел u,v, удовлетворяющие условию $au + bv = d$, воспользуемся теоремой 1. Неполные частные от деления q_k определены в процессе применения алгоритма Евклида (п.1).

Весь процесс вычисления удобно записать в таблице

k	0	1	2
q_k		8	2
u_k	0	1	-2
v_k	1	-8	17

$$u_2 = u_0 - u_1q_1 = 0 - 1*2 = -2$$

$$v_2 = v_0 - v_1q_1 = 1 - (-8)*2 = 17$$

Таким образом, $u = u_2$; $v = v_2$; $u = -2$; $v = 17$

4 Проверим,

12103*(-2) + 1425*17 = -24206+24225 = 19 = d

Задание 2. Проверить являются ли числа *a* и *b* взаимно простыми.

Задача

a = 141; b = 119

Решение

Воспользуемся определением: Целые числа a и b взаимно простые, если их неотрицательный НОД равен 1.

1. Применим алгоритм Евклида к числам a = 141; b = 119

$$141 = 119*1 + 22$$

$$119 = 22*5 + 9$$

$$22 = 9*2 + 4$$

$$9 = 4*2 + 1$$

$$4 = 1*4 + 0$$

Таким образом, получили, что найти наибольший делитель d = (141, 119) = 1, следовательно, числа 141 и 119 являются взаимно простыми.

3 Задания для самостоятельной работы

1 Найти наибольший делитель *d=(a,b)*

а) 164 и 94 б) 321 и 72 в) 474 и 147 г) 522 и 231
д) 2134 и 542 е) 2137 и 1211 ж) 12105 и 1625

2 Найти наименьшее общее кратное *m=[a,b]* целых чисел

а) 172 и 86 б) 452 и 94 в) 776 и 248 г) 548 и 262
д) 234 и 646 е) 2435 и 1325 ж) 11704 и 1128

3 Проверить являются ли числа *a* и *b* взаимно простыми.

а) 173 и 41 б) 453 и 94 в) 873 и 447 г) 541 и 267
д) 643 и 712 е) 3456 и 1421 ж) 11614 и 1191

Тема: Решение сравнений

1 Теоретическая часть

Пусть

$$a_n x^n + a_{n-} x^{n-1} + ... + a_1 x^1 + a_0 \equiv 0 (\mathrm{mod} \quad m) \qquad (5)$$

есть сравнения, в котором $a_0, a_1, ..., a_n \in Z$.

Определение 1. Решением сравнения (5) называют любое целое число, при подстановке которого вместе x сравнения (5) превращается в верное сравнение целых чисел. Сравнения с неизвестным x называют равносильными, если они умеют одно и то же множество решений.

Замечания:

1) из свойства 5 сравнений легко следует, что при замене в сравнении (5) любого из коэффициентов a_j сравнимым с ним по модулю m числом получится сравнение, равносильное сравнению (5). Следовательно, не теряя общности, можно считать, что

$$a \le a_j < m, i = 0, ..., n.$$

2) Если некоторое целое число x_1 есть решение сравнения (5), и $x_1 \equiv x_2 (\mathrm{mod} \quad m)$, то x_2 так же есть решение сравнений (5). Иначе говоря, если целое число x_2 также есть решение сравнений (5). Иначе говоря, если целое число x_1 удовлетворяет сравнению (5), то ему удовлетворяют и все числа класса $[x_1]m$. В связи с этим весь этот класс называют одним решением по модулю m, а число различных классов целых чисел, удовлетворяющих сравнению (5), числом решений по модулю m.

3) Если $m = m_1 m_2$ и $(m_1, m_2) = 1$, то из свойств делимости целых чисел следует, что $m|c$ в том и только том случае, когда $m_i|c$, $i = 1, 2$. Следовательно, в этом случае сравнение (5) равносильно системе сравнений

$$a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x^1 + a_0 \equiv 0 (\mathrm{mod} \quad m_j), \; i = 1,2.$$

Этот факт с учетом наличия канонического разложения числа m сводит решение сравнений к решению сравнений по примарным модулям.

Замечания 1) – 2) позволяют при небольших *m* найти все решения сравнения путем непосредственной проверки, например, подстановкой вместо x чисел любой полной системы вычетов по модулю *m*. Однако при большом m такой метод решения не реален, и возникает задача нахождения более рациональных методов. В общем случае эта задача сложная. Здесь мы рассмотрим лишь сравнения 1-й степеней.

Любое сравнения 1-й степени можно записать в виде

$$ax \equiv b(mod\ m), \qquad (6)$$

где $a, b \in Z$ и m не делит *a*.

Теорема 1. Сравнение (6) разрешимо тогда и только тогда, когда (a, m) =d и d|b. При выполнении последнего условия сравнение (6) имеет равно d решений по модулю m.

Следствие. Если целые числа a, m взаимно просты, то сравнение (6) имеет единственное решение по модулю m.

2 Практическая часть

Задача: Решить сравнение**.**

$$729x \equiv 33 \pmod{321}$$

Решение.

1. Упростим сравнение, заменив в нем коэффициент 729 остатком от деления на 321. Получим сравнение

$$87x \equiv 33 \pmod{321}.$$

2. Проверим условие разрешимости:

Найдем НОД 87 и 321 по алгоритму Евклида.

$$321 = 87*3 + 60$$
$$87 = 60*1 + 27$$
$$60 = 27*2 + 6$$
$$27 = 6*4 + 3$$
$$6 = 3*2 + 0.$$

Таким образом, (321, 87) = 3, кроме этого свободный член 33|3, следовательно, сравнение разрешимо и имеет 3 решения по модулю 321.

3. Воспользовавшись свойствами сравнения, исходное сравнение заменим эквивалентным

$$29x \equiv 11 \pmod{107},$$

которое имеет единственное решение по модулю 107, решим его.

Применим алгоритм Евклида к числам 107 и 29

$$107 = 29*3 + 20$$
$$29 = 20*1 + 9$$
$$20 = 9*2 + 2$$
$$9 = 2*4 + 1$$
$$2 = 1*2 + 0.$$

Получим последний неравный нулю остаток $r_4 = 1$ и систему неполных частных $q_1 = 3$, $q_2 = 1$, $q_3 = 2$, $q_4 = 4$. По ним, воспользовавшись рекуррентной формулой

$$v_k = v_{k-2} - v_{k-1}q_k,$$

найдем число $v = v_4$, при котором $107u + 29v = 1$.

Весь процесс вычисления удобно записать в виде таблице

k	0	1	2	3	4
q_k		3	1	2	4
v_k	1	-3	4	-11	48

Теперь можно записать решение сравнения $x = 48*11 = 528 \equiv 100 \pmod{107}$.

Или решения исходного сравнения будут равны:

$x_1 \equiv 100 \pmod{321}$, $x_2 \equiv 207 \pmod{321}$, $x_3 \equiv 314 \pmod{321}$.

3 Задания для самостоятельной работы

Выяснить какие из сравнений имеют решения и решить их:

а) $3x \equiv 5 \pmod{19}$ б) $5x \equiv 12 \pmod{26}$ в) $8x \equiv 7 \pmod{14}$ г) $18x \equiv 15 \pmod{69}$

д) $285x \equiv 51 \pmod{363}$ е) $2237x \equiv 100 \pmod{1021}$ ж) $12103x \equiv 385 \pmod{1425}$

Тема: Решение задач с элементами теории чисел

1 Теоретическая часть

Определение 1. Два целых числа a, b назовем *сравнимыми по модулю m*, если они при делении на m дают одинаковые остатки.

Обозначение: *a=b(mod m).*

Теорема 1. Сравнение целых чисел по модулю m обладают следующими свойствами для любых a, b, c, d$\in Z$:

1) $a \equiv a \pmod{m}$,
2) $a \equiv b \pmod{m}$, $\Rightarrow b \equiv a \pmod{m}$,
3) $a \equiv b \pmod{m}, b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$,
4) $a \equiv b \pmod{m} \Leftrightarrow m|a-b \Leftrightarrow a=b+mt$ *при некотором* $t \in Z$,
5) $a \equiv b$, $c \equiv d \pmod{m} \Rightarrow a \pm c \equiv b \pm d \pmod{m}$, $ac \equiv bd \pmod{m}$,

(*т. е. сравнения можно почленно складывать, вычитать и перемножать*),

6) если *d* есть общий делитель числа a, b, m, то $a \equiv b \pmod{m} \Leftrightarrow \frac{a}{d} \equiv \frac{b}{d} \left(\mathrm{mod}\ \frac{m}{d}\right)$,

7) если числа *a, b* делятся на числа *d и d* взаимно просто с *m,* то $a \equiv b \pmod{m} \Leftrightarrow \frac{a}{d} \equiv \frac{b}{d} (\mathrm{mod}\ m)$.

Множество целых чисел разбивается на непересекающиеся классы сравнимых по модулю *m* чисел. Так как различные остатки при делении на *m* исчерпываются числами

$$0,1,2,\ldots,m-1,$$

то получим m классов:

$$[0]_m, [1]_m, [2]_m, \ldots, [m-1]_m, \qquad (4)$$

Определение 2. Совокупность чисел, взятых по одному из каждого класса вычетов по модулю *m*, называется полной системой вычетов по модулю *m.*

Классы (4) называются классами вычетов, а их элементы вычетами по модулю *m.*

Определение 3. Совокупность чисел, взятых по одному из каждого класса вычетов по модулю *m*, называется полной системой вычетов по модулю *m*.

Определение 4. Класс вычетов по модулю *m*, состоящий из чисел, взаимно простых с *m*, называется классом, взаимно простым с модулем *m*.

Определение 5. Совокупность чисел, взятых по одному из всех классов, взаимно простых с модулем *m*, называются приведенной системой вычетов по модулю *m*.

Число классов, взаимно простых с модулем *m*, обозначают $\varphi(m)$, а функцию φ, сопоставляющую числу *m* число $\varphi(m)$,, называют функцией Эйлера. $\varphi(m)$ есть число натуральных чисел, не превосходящих *m* и взаимно простых с *m*.
Например, $\varphi(1) = 1$, $\varphi(2) = 1$, $\varphi(8) = 4$
Очевидно, что для простого числа p имеем $\varphi(p) = p - 1$. Общее: $\varphi(p^k) = p^k - p^{k-1}$ при любом натуральном *k*.
Если число *m* имеет каноническое разложение

$$m = p_1^{k1} \ldots p_s^{ks},$$

то

$$\varphi(m) = m(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})\ldots(1 - \frac{1}{p_s})$$

2 Практическая часть

Задача 1. Определить в множестве А элементы сравнимые с числом 6 по модулю 4, если А={3,7,9,10,12,31,29,24,20,17,1,26}

Решение:

1. Найдем остатки от деления числа 6 на модуль 4

   $$6 \equiv 2 \pmod 4.$$

2. Найдем остатки от деления элементов множества А на модуль 4

   $3 \equiv 3 \pmod 4$; $7 \equiv 3 \pmod 4$; $9 \equiv 1 \pmod 4$; $10 \equiv 2 \pmod 4$; $12 \equiv 0 \pmod 4$;

   $31 \equiv 3 \pmod 4$; $29 \equiv 1 \pmod 4$; $24 \equiv 0 \pmod 4$; $20 \equiv 0 \pmod 4$; $17 \equiv 2 \pmod 4$;

   $1 \equiv 1 \pmod 4$; $26 \equiv 2 \pmod 4$

3. Проанализируем полученные результаты и выберем те элементы множества А, которые имеют тот же остаток от деления на модуль 4, который получился в п.1.

   $10 \equiv 2 \pmod 4$; $17 \equiv 2 \pmod 4$; $26 \equiv 2 \pmod 4$

   Таким образом, получаем

   $$6 \equiv 10 \pmod 4;\ 6 \equiv 17 \pmod 4;\ 6 \equiv 26 \pmod 4$$

Задача 2. Определить функцию Эйлера для числа 24.
Решение.
1 Найдем каноническое разложение числа

$$m = 24 = 2^3 * 3$$

2 Найдем функцию Эйлера, используя приведенную формулу.

$$\varphi(24) = 24 * (1 - \frac{1}{2})(1 - \frac{1}{3}) = 24 * \frac{1}{2} * \frac{2}{3} = 8.$$

Таким образом, функцию Эйлера для числа m = 24 равна $\varphi(24) = 8$.

Задача 3. Найти остаток от деления числа 13^{162} на число 17.
Решение.

1. Найдем НОД чисел a = 13 и m = 17.

$$d = (a, m) = (13, 17) = 1,$$

таким образом, числа 13 и 17 взаимно простые.

2. Так как (13,17) = 1 и 17 простое число, то $\varphi(17) = 16$, применяя теорему Эйлера, получим

$$13^{16} \equiv 1 \pmod{17}.$$

Отсюда следует, что

$$13^{162} \equiv (13^{16})^{10} \cdot 13^2 \equiv 13^2 \equiv (-4)^2 \equiv 16 \pmod{17}.$$

Искомый остаток равен 16.

Задача 4. Определить принадлежат ли следующие числа заданного множества A = {3,13,25,37,49,61,71} полной системе вычетов по модулю 7.

Решение.

Воспользуемся определением: Совокупность чисел, взятых по одному из каждого класса вычетов по модулю m, называется полной системой вычетов по модулю m.

1. Найдем остатки от деления элементов множества A на модуль 7

$3 \equiv 3 \pmod 7$; $13 \equiv 6 \pmod 7$; $25 \equiv 4 \pmod 7$; $37 \equiv 2 \pmod 7$; $49 \equiv 0 \pmod 7$;
$61 \equiv 5 \pmod 7$; $71 \equiv 1 \pmod 7$

2. Проанализируем полученные результаты: так как в множестве A присутствует совокупность чисел, взятых по одному из каждого класса вычетов по модулю m (все остатки разные от 0 до 6), то данные числа принадлежат полной системе вычетов по модулю 7.

3 Задания для самостоятельной работы

Задание 1 Определить функцию Эйлера для числа m

а) 41 б) 453 в) 94 г) 541 д) 642 е) 1422 ж) 91

Задание 2 Найти остаток от деления:

а) 5^{223} на число 13 б) 7^{135} на число 11 в) 11^{251} на число 13
г) 13^{216} на число 17 д) 15^{173} на число 13 е) 7^{215} на число 19

Задание 3Определить принадлежат ли следующие числа заданного множества *A* полной системе вычетов по модулю *m*.

а) {0,16,28,38,49,60,70,50,93} по модулю 9
б) {16,26,33,47,60,55} по модулю 6
в) {66,25,37,48,64,79,54,9} по модулю 8
г) {67,28,37,48,68,76,45} по модулю 7

Тема: Применение классических шифров замены

1 Теоретическая часть

Одним из важных подклассов методов замены являются *одноалфавитные* (или моноалфавитные) подстановки, в которых устанавливается однозначное соответствие между каждым знаком a_i исходного алфавита сообщений A и соответствующим знаком e_i зашифрованного текста E. Одноалфавитная подстановка иногда называется также простой заменой, так как является самым простым шифром замены.

Шифр Цезаря применительно к русскому языку (**Ошибка! Источник ссылки не найден.**) состоит в следующем. Каждая буква сообщения заменяется на другую, которая в русском алфавите отстоит от исходной на три позиции дальше. Таким образом, буква А заменяется на Г, Б на Д, и так далее вплоть до буквы Ь, которая заменялась на Я, затем Э на А, Ю на Б и, наконец, Я на В.

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я

Рис. 0.1. Исходный алфавит

Так, например, слово З А М Е Н А после шифрования методом Цезаря превратится в К Г П З Р Г

2 Практическая часть

Задача Пусть, например, перехвачено секретное сообщение Ч С Ю Э Ю Ъ. Противнику известно, что ключ (параметр сдвига n) может принимать значения от 1 до 32. Пытаясь найти значение секретного ключа, мы будем проводить атаку по шифротексту. Рассмотрим способ последовательного перебора всех возможных ключей (это так называемый метод «грубой силы»).

Запишем на 32 строчках все варианты, которые получаются сдвигом каждой буквы на 1, 2, 3, ... , 32 позиции соответственно. Эту операцию можно проводить вручную, а можно составить несложную программу, которая запишет все варианты перебора параметра n в файл. Одна из этих 32 строк будет содержать исходное сообщение (**Ошибка! Источник ссылки не найден.**).

Перехваченная криптограмма: ЧСЮЭЮЪ

1	ШТЯЮЯЫ	17	ЗВОНОК
2	ЩУАЯАЬ	18	ИГПОПЛ
3	ЪФБАБЭ	19	ЙДРПРМ
4	ЫХВБВЮ	20	КЕСРСН
5	ЬЦГВГЯ	21	ЛЁТСТО
6	ЭЧДГДА	22	МЖУТУП
7	ЮШЕДЕБ	23	НЗФУФР
8	ЯЩЁЕЁВ	24	ОИХФХС
9	АЪЖЁЖГ	25	ПЙЦХЦТ
10	БЫЗЖЗД	26	РКЧЦЧУ
11	ВЬИЗИЕ	27	СЛШЧШФ
12	ГЭЙИЙЁ	28	ТМЩШЩХ
13	ДЮКЙКЖ	29	УНЪЩЪЦ
14	ЕЯЛКЛЗ	30	ФОЫЪЫЧ
15	ЁАМЛМИ	31	ХПЬЫЬШ
16	ЖБНМНЙ	32	ЦРЭЬЭЩ

Рис. 0.2. Перебор вариантов для поиска ключа при использовании метода Цезаря

Мы видим, что единственное слово, имеющее смысл – это ЗВОНОК. Это слово располагается на 17 месте. Следовательно, если шифрованный текст сдвинуть на 17 позиций вперед, получится открытый текст. Это означает, что для получения шифрованного текста открытый текст нужно сдвинуть на (33-17)=16 позиций. Таким образом, получили, что при шифровании ключ был равен n=16.

Задание: Подстановка может быть задана с помощью таблицы, например, как показано на рис. 0.3.

Откр. текст	**Шифр 1**	**Шифр 2**		**Откр. текст**	**Шифр 1**	**Шифр 2**		**Откр. текст**	**Шифр 1**	**Шифр 2**
А	В	^		М	Т	№		Ч	М	∑
Б	И	@		Н	Ц	#		Ш	У	∇
В	О	)		О	.	-		Щ	Д	ϒ
Г	А	+		П	Ж	=		Ъ	Э	ℵ
Д	Щ	<		Р	Г	(		Ы	Н	⊕
Е	П	>		С	Л	?		Ь	Ю	×
Ж	К	∀		Т	Х	%		Э	Ы	ω
З	Б	♦		У	С	⊗		Ю	Ш	#
И	Ъ	*		Ф	Ь	!		Я	Е	Δ
К	пробел	♥		Х	Ч	№		пробел	Ф	∞
Л	Р	♠		Ц	З	®		.	Я	♣

Рис. 0.3. Пример таблицы замен для двух шифров

В таблице на рис. 0.3 на самом деле объединены сразу две таблицы. Одна (шифр 1) определяет замену русских букв исходного текста на другие русские буквы, а вторая (шифр 2) – замену букв на специальные символы. Исходным алфавитом для обоих шифров будут заглавные русские буквы (за исключением букв «Ё» и «Й»), пробел и точка.

Зашифрованное сообщение с использованием любого шифра моноалфавитной подстановки получается следующим образом. Берется очередной знак из исходного сообщения. Определяется его позиция в столбце «Откр. текст» таблицы замен. В зашифрованное сообщение вставляется шифрованный символ из этой же строки таблицы замен.

Попробуем зашифровать сообщение «ВЫШЛИТЕ ПОДКРЕПЛЕНИЕ» с использованием этих двух шифров (рис. 0.4). Для этого берем первую букву исходного сообщения «В». В таблице на рис. 0.3 в столбце «Шифр 1» находим для буквы «В» заменяемый символ. Это будет буква «О». Записываем букву «О» под буквой «В». Затем рассматриваем второй символ исходного сообщения – букву «Ы». Находим эту букву в столбце «Откр. текст» и из столбца «Шифр 1» берем букву, стоящую на той же строке, что и буква «Ы». Таким образом получаем второй символ зашифрованного сообщения – букву «Н». Продолжая действовать аналогично, зашифровываем все исходное сообщение (рис. 0.4).

Полученный таким образом текст имеет сравнительно низкий уровень защиты, так как исходный и зашифрованный тексты имеют одинаковые статистические закономерности. При этом не имеет значения, какие символы использованы для замены – перемешанные символы исходного алфавита или таинственно выглядящие знаки.

Открытое сообщение																			
В	Ы	Ш	Л	И	Т	Е		П	О	Д	К	Р	Е	П	Л	Е	Н	И	Е
Зашифрованное сообщение с использованием шифра 1																			
О	Н	У	Р	Ъ	Х	П	Ф	Ж	.	Щ		Г	П	Ж	Р	П	Ц	Ъ	П
Зашифрованное сообщение с использованием шифра 2																			
)	⊕	∇	♠	*	%	>	∞	=	-	<	♥	(	>	=	♠	>	#	*	>

Рис. 0.4. Пример шифрования методом прямой замены

3 Задания для самостоятельной работы

Задание. Определите ключи шифра Цезаря, если известны следующие пары открытый текст - шифротекст:

а) АПЕЛЬСИН - ТВЧЮОДЫА

б) МАНДАРИН – ТЁУЙЁЦОУ

в) ЯБЛОКО – ЗЙФЧУЧ

г) ГРУША – ЮЛОУЫ

д) ВИНОГРАД – ШЯДЕЩЖЦЪР

е) АБРИКОС – ЛМЬФЦЪЭИ

<u>Задание</u> Расшифруйте следующие сообщения, зашифрованные шифром Цезаря, и определите ключ n, $0 < n < 33$.

а) Ю В П Л Ш У Х б) С Ф Ы Ю Б Ш Я Ф У в) Ж К И Л Ш Ъ О Б М

г) Ъ Ё О Ъ Ь Г М д) Е В В Ф М Ш Б Ь Ш е) И Ы Л Р У Д Х Р Т

3. Расшифруйте сообщения, зашифрованные с помощью шифра №1 из рис. 0.3:

а) ТПХ.ЩФБВТПЦН б) ВЖПРЮЛЪЦ

в) Л.РЦПМЦ.ПФБВХТПЦЪП г) И.РЮУ.ЪФОБГНО

д) Ж.ЩО.ЩЦВЕФР.Щ В е) ЛХГ.ЪЛХО.ФОО.ЩА

4.. Расшифруйте сообщения, зашифрованные с помощью шифра №2 из рис. 0.3:

а) ^=>♠×?*# б) =-<?%^#-)♥^

в) ∇*!(∞♦^№#⊕ г) ♥(*=%-^#^♠*♦

д) @♠-♥∞∇*!(-)^#*Δ е) ♥(*=№-+(^№№^

Тема: Применение классических шифров перестановки

1 Теоретическая часть

перестановка по таблице. В этом методе производится запись исходного текста по строкам некоторой таблицы и чтение его по столбцам этой же таблицы. Последовательность заполнения строк и чтения столбцов может быть любой и задается ключом.

Рассмотрим пример. Пусть в таблице кодирования будет 4 столбца и 3 строки (размер блока равен 3×4=12 символов). Зашифруем такой текст:

ЭТО ТЕКСТ ДЛЯ ШИФРОВАНИЯ

Количество символов в исходном сообщении равно 24, следовательно, сообщение необходимо разбить на 2 блока. Запишем каждый блок в свою таблицу по строчкам (рис. 0.5).

1 блок:

Э	Т	О	
Т	Е	К	С
Т		Д	Л

2 блок:

Я		Ш	И
Ф	Р	О	В
А	Н	И	Я

Рис. 0.5. Шифрование методом перестановки по таблице

Затем будем считывать из таблицы каждый блок последовательно по столбцам:

ЭТТТЕ ОКД СЛЯФА РНШОИИВЯ

Можно считывать столбцы не последовательно, а, например, так: третий, второй, первый, четвертый:

ОКДТЕ ЭТТ СЛШОИ РНЯФАИВЯ

В этом случае порядок считывания столбцов и будет ключом.

2 Практическая часть

В случае если размер сообщения не кратен размеру блока, можно дополнить сообщение какими-либо символами, не влияющими на смысл, например, пробелами. Однако это делать не рекомендуется, так как это дает противнику в случае перехвата криптограммы информацию о размере используемой таблицы перестановок (длине блока). После определения длины блока противник может найти длину ключа (количество столбцов таблицы) среди делителей длины блока.

Посмотрим, как зашифровать и расшифровать сообщение, имеющее длину, не кратной размеру таблицы перестановки. Зашифруем слово

ПЕРЕМЕНКА

Количество символов в исходном сообщении равно 9. Запишем сообщение в таблицу по строкам (рис. 0.6), а последние три ячейки оставим пустыми.

П	Е	Р	Е
М	Е	Н	К
А			

Рис. 0.6. Шифрование неполного блока методом перестановки по таблице

Затем будем считывать из таблицы последовательно по столбцам:

ПМАЕЕРНЕК

Для расшифрования вначале определяют число полных столбцов, то есть количество символов в последней строке. Для этого делят размер сообщения (в нашем примере – 9) на количество столбцов или размер ключа (в примере – 4). Остаток от деления будет числом полных столбцов: 9 mod 4 = 1. Следовательно, в нашем примере был 1 полный столбец и три коротких. Теперь можно поставить буквы сообщения на свои места и расшифровать сообщение. Так как ключом при шифровании было число 1234 (столбцы считывались последовательно), то при расшифровании первые три символа (ПМА) записываются в первый столбец таблицы перестановки, следующие два (ЕЕ) – во второй столбец, следующие два (РН) – в третий, и последние два (ЕК) – в четвертый. После заполнения таблицы считываем строки и получаем исходное сообщение ПЕРЕМЕНКА.

3 Задания для самостоятельной работы

1. Зашифруйте сообщения методом перестановки по таблице 5×5. Ключ указывает порядок считывания столбцов при шифровании.

а) ШИРОКОПОЛОСНЫЙ УСИЛИТЕЛЬ (ключ: 41235)
б) ПЕРЕДАЧА ИЗОБРАЖЕНИЯ (ключ: 24513)
в) ТЕЛЕСКОПИЧЕСКАЯ АНТЕННА (ключ: 35241)
г) ПРЕОБРАЗОВАТЕЛЬ КОДОВ (ключ: 43251)
д) НАДЕЖНОЕ ШИФРОВАНИЕ (ключ: 13524)
е) ГЕНЕРАТОР СЛУЧАЙНЫХ ЧИСЕЛ (ключ: 42531)

2. Расшифруйте сообщения, зашифрованные методом перестановки по таблице 4×4 (символ подчеркивания заменяет пробел). Ключ указывает порядок считывания столбцов при шифровании.

а) ЕАУПД_КЕАЗАРЧВ (ключ: 4123)
б) А_НСЫИЛБСАЛЙГ (ключ: 3142)
в) ТДЫЖЕНСИАХА_Н (ключ: 4213)
г) НАНЩ_ЕОЯТАМАНН (ключ: 4321)
д) РНПЛНЕРЙХЕ_ОАИАЕ (ключ: 2413)
е) ЕТ_ЧСЕЙЮРЫЛКНК (ключ: 2143)

Тема: Применение классических шифров гаммирования

1 Теоретическая часть

Частным случаем многоалфавитной подстановки является *гаммирование*. В этом способе шифрование выполняется путем сложения символов исходного текста и ключа по модулю, равному числу букв в алфавите. Если в исходном алфавите, например, 33 символа, то сложение производится по модулю 33. Такой процесс сложения исходного текста и ключа называется в криптографии *наложением гаммы*.

Пусть символам исходного алфавита соответствуют числа от 0 (А) до 32 (Я). Если обозначить число, соответствующее исходному символу, x, а символу ключа – k, то можно записать правило гаммирования следующим образом:

$$z = x + k \pmod N,$$

где z – закодированный символ, N - количество символов в алфавите, а сложение по модулю N - операция, аналогичная обычному сложению, с тем отличием, что если обычное суммирование дает результат, больший или равный N, то значением суммы считается остаток от деления его на N. Например, пусть сложим по модулю 33 символы Г (3) и Ю (31):

$$3 + 31 \pmod{33} = 1,$$

то есть в результате получаем символ Б, соответствующий числу 1.

Наиболее часто на практике встречается двоичное гаммирование. При этом используется двоичный алфавит, а сложение производится по модулю два. Операция сложения по модулю 2 часто обозначается ⊕, то есть можно записать:

$$z = x + k \pmod 2 = x \oplus k.$$

Операция сложения по модулю два в алгебре логики называется также «исключающее ИЛИ» или по-английски XOR.

При использовании методов *перестановки* входной поток исходного текста делится на блоки, в каждом из которых выполняется перестановка символов. Перестановки в классической «докомпьютерной» криптографии получались в результате записи исходного текста и чтения шифрованного текста по разным путям геометрической фигуры.

Простейшим примером перестановки является *перестановка с фиксированным периодом d*. В этом методе сообщение делится на блоки по d символов и в каждом блоке производится одна и та же перестановка. Правило, по которому производится перестановка, является ключом и может быть задано некоторой перестановкой первых d натуральных чисел. В результате сами буквы сообщения не изменяются, но передаются в другом порядке.

2 Практическая часть

Задача 1. Предположим, нам необходимо зашифровать десятичное число 14 методом гаммирования с использованием ключа 12. Для этого вначале необходимо преобразовать исходное число и ключ (гамму) в двоичную форму[1]: $14_{(10)}=1110_{(2)}$, $12_{(10)}=1100_{(2)}$. Затем надо записать полученные двоичные числа друг под другом и каждую пару символов сложить по модулю два. При сложении двух двоичных знаков получается 0, если исходные двоичные цифры одинаковы, и 1, если цифры разные:

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0.$$

Сложим по модулю два двоичные числа 1110 и 1100:

Исходное число	1 1 1 0
Гамма	1 1 0 0
Результат	0 0 1 0

В результате сложения получили двоичное число 0010. Если перевести его в десятичную форму, получим 2. Таким образом, в результате применения к числу 14 операции гаммирования с ключом 12 получаем в результате число 2.

Каким же образом выполняется расшифрование? Зашифрованное число 2 представляется в двоичном виде и снова производится сложение по модулю 2 с ключом:

Зашифрованное число	0 0 1 0
Гамма	1 1 0 0
Результат	1 1 1 0

Переведем полученное двоичное значение 1110 в десятичный вид и получим 14, то есть исходное число.

Таким образом, при гаммировании по модулю 2 нужно использовать одну и ту же операцию как для зашифрования, так и для расшифрования. Это позволяет использовать один и тот же алгоритм, а соответственно и одну и ту же

.

программу при программной реализации, как для шифрования, так и для расшифрования.

Задача 2. для d=6 в качестве ключа перестановки можно взять 436215. Это означает, что в каждом блоке из 6 символов четвертый символ становится на первое место, третий – на второе, шестой – на третье и т.д. Пусть необходимо зашифровать такой текст:

ЭТО_ТЕКСТ_ДЛЯ_ШИФРОВАНИЯ

Количество символов в исходном сообщении равно 24, следовательно, сообщение необходимо разбить на 4 блока. Результатом шифрования с помощью перестановка 436215 будет сообщение

_ОЕТЭТ_ТЛСКДИШР_ЯФНАЯВОИ

Теоретически, если блок состоит из d символов, то число возможных перестановок равно $d!=1\times2\times...\times(d-1)\times d$. В последнем примере d=6, следовательно, число перестановок равно $6!=1\times2\times3\times4\times5\times6=720$. Таким образом, если противник перехватил зашифрованное сообщение из рассмотренного примера, ему понадобится не более 720 попыток для раскрытия исходного сообщения (при условии, что размер блока известен противнику).

3. Задания для самостоятельной работы

Задание 1. Первый байт фрагмента текста в шестнадцатеричном виде имеет вид A5. На него накладывается по модулю два 4-х битовая гамма 0111 (в двоичном виде). Что получится после шифрования?

Задание 2. Первый байт фрагмента текста, зашифрованного методом гаммирования (по модулю 2), в шестнадцатеричном виде имеет вид 9A. До шифрования текст имел первый байт, равный 74 (в шестнадцатеричном виде). Какой ключ использовался при шифровании?

Задание 3. Зашифруйте методом перестановки с фиксированным периодом d=6 с ключом 436215 сообщения:

а) ЖЕЛТЫЙ_ОГОНЬ б) МЫ_НАСТУПАЕМ
в) ШИФРЫ_ЗАМЕНЫ г) КРИПТОГРАФИЯ
д) ПЕРЕСТАНОВКА е) ГАММИРОВАНИЕ

8. Расшифруйте сообщения, зашифрованные методом перестановки с фиксированным периодом d=8 с ключом 64275813:

а) СЛПИЬНАЕ б) РОИАГДВН
в) ОИБСКЫАР г) ИБЛКНАКУ
д) РДАИАНМН е) ЖЛАААНБК

Задание 4. Определите ключи в системе шифрования, использующей перестановку с фиксированным периодом d=5 по парам открытых и зашифрованных сообщений:

а) МОЙ ПАРОЛЬ – ЙПМ ООЬАЛР
б) СИГНАЛ БОЯ – НИСАГО ЛЯБ
в) ШИФРОВАНИЕ – ОИРШФЕАИВН
г) ИНФОРМАЦИЯ – НРФИОАЯЦМИ
д) ОБЩИЙ_КЛЮЧ – ИЩБЙОЮЛКЧ_

е) ОДНА_БУКВА – _НОАДАКБВУ

Тема: Криптоанализ шифра Вижинера

1 Теоретическая часть

Примером многоалфавитной подстановки может служить схема, основанная на использовании таблицы Виженера (рис. 0.7). Этот метод, известный уже в XVI веке, был описан французом Блезом Виженером в «Трактате о шифрах», вышедшем в 1585 году. В этом методе для шифрования используется таблица, представляющая собой квадратную матрицу с числом элементов N×N, где N — количество символов в алфавите. В первой строке матрицы записывают буквы в порядке очередности их в исходном алфавите, во второй — ту же последовательность букв, но с циклическим сдвигом влево на одну позицию, в третьей — со сдвигом на две позиции и т. д.

АБВГДЕ	.....ЭЮЯ
БВГДЕЖ	.....ЮЯА
ВГДЕЖЗ	.....ЯАБ
ГДЕЖЗИ.....	.....АБВ
ДЕЖЭИК.....	.....БВГ
ЕЖЗИКЛ.....	.....ВГД
.....	.....
ЯАБВГД.....	.....ЬЭЮ

Рис. 0.7. Подготовка таблицы шифрования

Для шифрования текста выбирают ключ, представляющий собой некоторое слово или набор символов исходного алфавита. Далее из полной матрицы выписывают подматрицу шифрования, включающую первую строку и строки матрицы, начальными буквами которых являются последовательно буквы ключа (например, если выбрать ключ «весна», то таблица шифрования будет такой, как на рис. 0.8).

АБВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯ

ВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯАБ

ЕЖЗИКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯАБВГД

НОПРСТУФХЦЧШЩЬЫЪЭЮЯАБВГДЕЖЗИКЛМ

СТУФХЦЧШЩЬЫЪЭЮЯАБВГДЕЖЗИКЛМНОПР

Рис. 0.8. Первый этап шифрования – составление подматрицы шифрования

В процессе шифрования (рис. 0.9) под каждой буквой шифруемого текста записывают буквы ключа, повторяющие ключ требуемое число раз, затем шифруемый текст по таблице шифрования (рис. 0.8) заменяют буквами, расположенными на пересечениях линий, соединяющих буквы текста первой строки таблицы и буквы ключа, находящейся под ней.
Например, под первой буквой исходного текста «М» записана буква «В» ключа. В таблице кодирования находим столбец, начинающийся с «М» и строку, начинающуюся с «В». На их пересечении располагается буква «О». Она и будет первым символом зашифрованного сообщения (на рис. 0.9 эта буква выделена прямоугольной рамочкой). Следующая буква исходного сообщения – «Е», символ ключа – тоже «Е». Находим пересечение строки, начинающейся с «Е» и столбца, начинающегося с «Е». Это будет буква «Л» – второй символ зашифрованного сообщения.

```
ИСХОДНЫЙ ТЕКСТ – МЕТОД ПЕРЕСТАНОВКИ   АБВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯ
КЛЮЧ           – ВЕСНА ВЕСНАВЕСНАВЕ   ВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯАБ
ЗАШИФРОВ.ТЕКСТ – ОЛВЬД СЛАТСФЕЭЬВМО   ЕЖЗИКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯАБВГД
                                      НОПРСТУФХЦЧШЩЬЫЪЭЮЯАБВГДЕЖЗИКЛМ
                                      СТУФХЦЧШЩЬЫЪЭЮЯАБВГДЕЖЗИКЛМНОПР
```

Рис. 0.9. Механизм шифрования многоалфавитной заменой

2 Практическая часть

Рассмотрим на примере процесс расшифрования сообщения по методу Виженера. Пусть имеется зашифрованное с помощью ключа ВЕСНА сообщение КЕКХТВОЭЦОТССВИЛ (пробелы при шифровании пропущены). Расшифровка текста выполняется в следующей последовательности (рис. 0.10):
– над буквами шифрованного текста сверху последовательно записывают буквы ключа, повторяя ключ требуемое число раз;
– в строке подматрицы таблицы Вижинера для каждой буквы ключа отыскивается буква, соответствующая знаку шифрованного текста. Находящаяся над ней буква первой строки и будет знаком расшифрованного текста;
– полученный текст группируется в слова по смыслу.

КЛЮЧ	–	ВЕСНАВЕСНАВЕСНАВ
ЗАШИФРОВАННЫЙ ТЕКСТ	–	КЕКХТВОЭЦОТССВИЛ
РАСШИФРОВАННЫЙ ТЕКСТ	–	ЗАЩИТАИНФОРМАЦИИ
ИСХОДНЫЙ ТЕКСТ	–	ЗАШИТА ИНФОРМАЦИИ

Рис. 0.10. Механизм расшифрования

Раскрыть текст, зашифрованный по методу Виженера, способом, рассмотренным в п.2.2.1 невозможно, так как одни и те же символы открытого текста могут быть заменены различными символами зашифрованного текста. С другой стороны, различные буквы открытого текста могут быть заменены одинаковыми знаками зашифрованного текста.

Особенность данного метода многоалфавитной подстановки заключается в том, что каждый из символов ключа используется для шифрования одного символа исходного сообщения. После использования всех символов ключа, они повторяются в том же порядке. Если используется ключ из десяти букв, то каждая десятая буква сообщения шифруется одним и тем же символом ключа. Этот параметр называется *периодом* шифра. Если ключ шифрования состоит из одного символа, то при шифровании будет использоваться одна строка таблицы Виженера, следовательно, в этом случае мы получим моноалфавитную подстановку, а именно шифр Цезаря.

С целью повышения надежности шифрования текста можно использовать подряд два или более зашифрования по методу Виженера с разными ключами (составной шифр Виженера).

На практике кроме метода Виженера использовались также различные модификации этого метода. Например, шифр Виженера с перемешанным один раз алфавитом. В этом случае для расшифрования сообщения получателю необходимо кроме ключа знать порядок следования символов в таблице шифрования.

3 Задания для самостоятельной работы

1. Зашифруйте с помощью таблицы Виженера и ключа ЯБЛОКО сообщения:

а) КРИПТОСТОЙКОСТЬ б) ГАММИРОВАНИЕ
в) ПОДСТАНОВКА г) ПЕРЕСТАНОВКА
д) ШИФРОГРАММА е) ИНФОРМАЦИЯ

2. Пусть исходный алфавит состоит из следующих знаков (символ «_» (подчеркивание) будем использовать для пробела):

АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЪЭЮЯ_

Расшифруйте сообщения, зашифрованные с помощью таблицы Виженера и ключа ОРЕХ:

а) ШВМБУЖНЯ б) ЯБХЪШЮМХ
в) ПРЖВДЪЕ г) ЫРЦЪШЮСПУ
д) ЪРОЕШШОФФВП е) _ЮРЕ_КОФЕЬ_АК

Тема: Применение методов генерации ПСЧ

Использование линейного конгруэнтного генератора для получения ПСЧ

1 Теоретическая часть

Генераторы ПСЧ могут работать по разным алгоритмам. Одним из простейших генераторов является так называемый *линейный конгруэнтный генератор*, который для вычисления очередного числа k_i использует формулу:

$$k_i = (a \times k_{i-1} + b) \bmod c,$$

где a, b, c — некоторые константы, а k_{i-1} — предыдущее псевдослучайное число. Для получения k_1 задается начальное значение k_0.

Достоинством линейных конгруэнтных генераторов ПСЧ является их простота и высокая скорость получения псевдослучайных значений. Линейные конгруэнтные генераторы находят применение при решении задач моделирования и математической статистики, однако в криптографических целях их нельзя рекомендовать к использованию, так как специалисты по криптоанализу научились восстанавливать всю последовательность ПСЧ по нескольким значениям. Например, предположим, что противник может определить значения k_0, k_1 , k_2, k_3. Тогда:

$$k_1 = (a \times k_0 + b) \bmod c,$$

$$k_2 = (a \times k_1 + b) \bmod c$$

$$k_3 = (a \times k_2 + b) \bmod c$$

Решив систему из этих трех уравнений, можно найти a, b и c .

2 Практическая часть

Возьмем в качестве примера a=5,b=3,c=11 и пусть k_0 = 1. В этом случае мы сможем по приведенной выше формуле получать значения от 0 до 10 (так как c = 11). Вычислим несколько элементов последовательности:

k_1 = (5 * 1 + 3) mod 11 = 8;
k_2 = (5 * 8 + 3) mod 11 = 10;
k_3 = (5 * 10 + 3) mod 11 = 9;
k_4 = (5 * 9 + 3) mod 11 = 4;
k_5 = (5 * 4 + 3) mod 11 = 1.

Полученные значения (8, 10, 9, 4, 1) выглядят похожими на случайные числа. Однако следующее значение k_6 будет снова равно 8:

k_6 = (5 * 1 + 3) mod 11 = 8.

А значения k_7 и k_8 будут равны 10 и 9 соответственно:

k_7 = (5 * 8 + 3) mod 11 = 10;
k_8 = (5 * 10 + 3) mod 11 = 9.

Выходит, наш генератор псевдослучайных чисел повторяется, порождая периодически числа 8, 10, 9, 4, 1. К сожалению, это свойство характерно для всех линейных конгруэнтных генераторов. Изменяя значения основных параметров a, b и c, можно влиять на длину периода и на сами порождаемые

значения k_i. Так, например, увеличение числа c в общем случае ведет к увеличению периода. Если параметры a, b и c выбраны правильно, то генератор будет порождать случайные числа с максимальным периодом, равным c. При программной реализации значение c обычно устанавливается равным 2^{b-1} или 2^b, где b — длина слова ЭВМ в битах.

3 Задания для самостоятельной работы

1. Значения k_0, k_1, k_2, k_3, полученные с помощью линейного конгруэнтного генератора, равны: $k_0 = 0$, $k_1 = 12$, $k_2 = 3$, $k_3 = 6$. Найдите параметры a, b и c генератора ПСЧ.

Использование. метода Фибоначчи с запаздыванием для получения ПСЧ

1 Теоретическая часть

Метод Фибоначчи с запаздываниями (Lagged Fibonacci Generator) — один из методов генерации псевдослучайных чисел. Он позволяет получить более высокое «качество» ПСЧ.

Наибольшую популярность фибоначчиевы датчики получили в связи с тем, что скорость выполнения арифметических операций с вещественными числами сравнялась со скоростью целочисленной арифметики, а фибоначчиевы датчики естественно реализуются в вещественной арифметике.

Известны разные схемы использования метода Фибоначчи с запаздыванием. Один из широко распространённых фибоначчиевых датчиков основан на следующей рекуррентной формуле:

$$k_i = \begin{cases} k_{i-a} - k_{i-b}, & \text{если} \quad k_{i-a} \geq k_{i-b} \\ k_{i-a} - k_{i-b} + 1, & \text{если} \quad k_{i-a} < k_{i-b} \end{cases}$$

где k_i — вещественные числа из диапазона [0,1], a, b — целые положительные числа, параметры генератора. Для работы фибоначчиеву датчику требуется знать $\max\{a,b\}$ предыдущих сгенерированных случайных чисел. При программной реализации для хранения сгенерированных случайных чисел необходим некоторый объем памяти, зависящих от параметров a и b.

2 Практическая часть

Задание 1. Вычислим последовательность из первых десяти чисел, генерируемую методом Фибоначчи с запаздыванием начиная с k_5 при следующих исходных данных: $a = 4$, $b = 1$, $k_0=0.1$; $k_1=0.7$; $k_2=0.3$; $k_3=0.9$; $k_4=0.5$:

$k_5 = k_1 - k_4 = 0.7 - 0.5 = 0.2$;
$k_6 = k_2 - k_5 = 0.3 - 0.2 = 0.1$;
$k_7 = k_3 - k_6 = 0.9 - 0.1 = 0.8$;
$k_8 = k_4 - k_7 + 1 = 0.5 - 0.8 + 1 = 0.7$;
$k_9 = k_5 - k_8 + 1 = 0.2 - 0.7 + 1 = 0.5$;
$k_{10} = k_6 - k_9 + 1 = 0.1 - 0.5 + 1 = 0.6$;
$k_{11} = k_7 - k_{10} = 0.8 - 0.6 = 0.2$;
$k_{12} = k_8 - k_{11} = 0.7 - 0.2 = 0.5$;
$k_{13} = k_9 - k_{12} + 1 = 0.5 - 0.5 + 1 = 1$;

$$k_{14} = k_{10} - k_{13} + 1 = 0.6 - 1 + 1 = 0.6;$$

Видно, что генерируемая последовательность чисел внешне похожа на случайную. И действительно, исследования подтверждают, что получаемые случайные числа обладают хорошими статистическими свойствами.

3 Задания для самостоятельной работы

1. Вычислите последовательность из первых десяти чисел, генерируемую методом Фибоначчи с запаздыванием начиная с k_5 при следующих исходных данных:

а) $a = 4$, $b = 1$, $k_0=0.1$; $k_1=0.7$; $k_2=0.3$; $k_3=0.9$; $k_4=0.5$

б) $a = 4$, $b = 2$, $k_0=0.1$; $k_1=0.7$; $k_2=0.3$; $k_3=0.9$; $k_4=0.5$

в) $a = 4$, $b = 1$, $k_0=0.6$; $k_1=0.3$; $k_2=0.5$; $k_3=0.9$; $k_4=0.2$

г) $a = 4$, $b = 2$, $k_0=0.1$; $k_1=0.3$; $k_2=0.5$; $k_3=0.9$; $k_4=0.2$

Генератор ПСЧ на основе алгоритма BBS

1 Теоретическая часть

Широкое распространение получил алгоритм генерации псевдослучайных чисел, называемый алгоритмом *BBS* (от фамилий авторов — L. Blum, M. Blum, M. Shub) или *генератором с квадратичным остатком*. Для целей криптографии этот метод предложен в 1986 году.

Он заключается в следующем. Вначале выбираются два больших простых числа p и q. Числа p и q должны быть оба *сравнимы* с 3 по модулю 4, то есть при делении p и q на 4 должен получаться одинаковый остаток 3. Далее вычисляется число $M = p \cdot q$, называемое целым число Блюма. Затем выбирается другое случайное целое число x, взаимно простое (то есть не имеющее общих делителей, кроме единицы) с M. Вычисляем $x_0 = x^2 \bmod M$. x_0 называется стартовым числом генератора.

На каждом n-м шаге работы генератора вычисляется $x_{n+1} = x_n^2 \bmod M$. Результатом n-го шага является один (обычно младший) бит числа x_{n+1}. Иногда в качестве результата принимают бит чётности, то есть количество единиц в двоичном представлении элемента. Если количество единиц в записи числа четное – бит четности принимается равным 0, нечетное – бит четности принимается равным 1.

2 Практическая часть

Задача

Пусть $p = 11$, $q = 19$ (убеждаемся, что 11 mod 4 = 3, 19 mod 4 = 3). Тогда $M = p \cdot q = 11 \cdot 19 = 209$. Выберем x, взаимно простое с M: пусть $x = 3$. Вычислим стартовое число генератора x_0:

$$x_0 = x^2 \bmod M = 3^2 \bmod 209 = 9 \bmod 209 = 9.$$

Вычислим первые десять чисел x_i по алгоритму BBS. В качестве случайных бит будем брать младший бит в двоичной записи числа x_i:

$x_1=9^2$ mod 209= 81 mod 209= 81; *младший бит: 1*
$x_2=81^2$ mod 209= 6561 mod 209= 82; *младший бит: 0*
$x_3=82^2$ mod 209= 6724 mod 209= 36; *младший бит: 0*
$x_4=36^2$ mod 209= 1296 mod 209= 42; *младший бит: 0*
$x_5=42^2$ mod 209= 1764 mod 209= 92; *младший бит: 0*
$x_6=92^2$ mod 209= 8464 mod 209= 104; *младший бит: 0*
$x_7=104^2$ mod 209= 10816 mod 209= 157; *младший бит: 1*
$x_8=157^2$ mod 209= 24649 mod 209= 196; *младший бит: 0*
$x_9=196^2$ mod 209= 38416 mod 209= 169; *младший бит: 1*
$x_9=169^2$ mod 209= 28561 mod 209= 137; *младший бит: 1*

3 Задания для самостоятельной работы

1. Определите период генератора ПСЧ, построенного по алгоритму BBS при следующих параметрах генератора:
а) $p = 11$, $q = 19$, $x = 3$; б) $p = 19$, $q = 7$, $x = 4$;
в) $p = 11$, $q = 7$, $x = 5$; г) $p = 11$, $q = 11$, $x = 2$.

Принцип работы генератора ПСЧ на основе сдвиговых регистров с обратной связью

1 Теоретическая часть

В теории кодирования и криптографии широко применяются так называемые *сдвиговые регистры с обратной связью*. Они использовались в аппаратуре шифрования еще до начала массового использования ЭВМ и современных высокоскоростных программных шифраторов.
Сдвиговые регистры с обратной связью могут применяться для получения потока псевдослучайных бит. Сдвиговый регистр с обратной связью состоит из двух частей: собственно n-битного сдвигового регистра и устройства обратной связи.

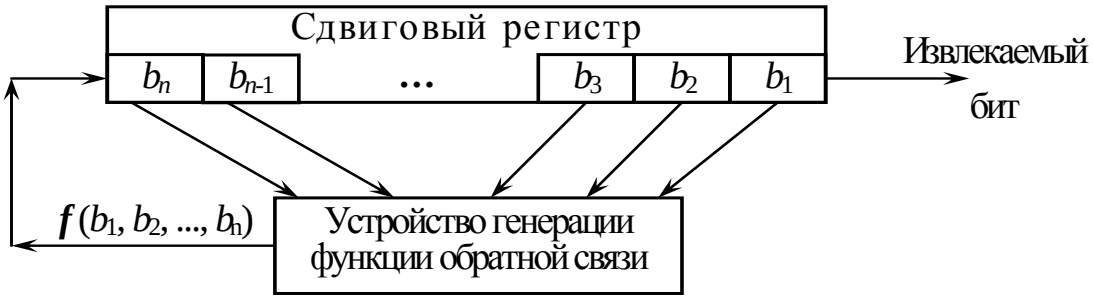


Рис. 0.11. Сдвиговый регистр с обратной связью

Извлекать биты из сдвигового регистра можно только по одному. Если необходимо извлечь следующий бит, все биты регистра сдвигаются вправо на 1 разряд. При этом на вход регистра слева поступает новый бит, который формируется устройством обратной связи и зависит от всех остальных битов сдвигового регистра. За счет этого биты регистра изменяются по определенному закону, который и определяет схему получения ПСЧ. Понятно, что через некоторое количество тактов работы регистра последовательность битов начнет повторяться. Длина получаемой последовательности до начала ее повторения называется *периодом* сдвигового регистра.

Поточные шифры с использованием сдвиговых регистров достаточно долго использовались на практике. Это связано с тем, что они очень хорошо реализуются с помощью цифровой аппаратуры.

Простейшим видом сдвигового регистра с обратной связью является *линейный сдвиговый регистр с обратной связью* (linear feedback shift register – LFSR). Обратная связь в этом устройстве реализуется просто как сумма по модулю 2 всех (или некоторых) битов регистра. Биты, которые участвуют в обратной связи, называются *отводной последовательностью*. Линейные сдвиговые регистры с обратной связью или их модификации часто применяются в криптографии.

2 Практическая часть

Для того, чтобы стало понятнее, как работает сдвиговый регистр с обратной связью, рассмотрим 4-битовый LFSR с отводом от второго и четвертого разрядов, представленный на рис. 0.12.

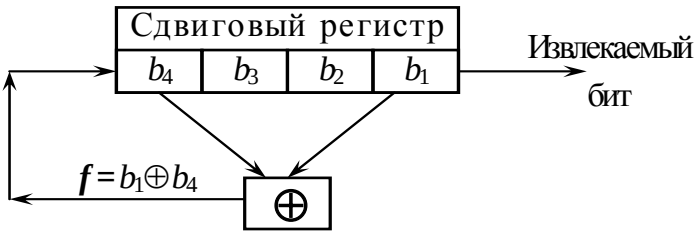


Рис. 0.12. Пример 4-разрядного линейного сдвигового регистра

Запишем в изображенный на рисунке регистр начальное значение 1011. Вычислять последовательность внутренних состояний регистра удобно с помощью таблицы, представленной на рис. 0.13. В таблице отражены первые десять состояний регистра.

Номер состояния	Внутреннее состояние регистра b_4, b_3, b_2, b_1	Результат вычисления функции обратной связи $f = b_1 \oplus b_4$	Извлекаемый бит (b_1)
0	1 0 1 1	0	1
1	0 1 0 1	1	1
2	1 0 1 0	1	0
3	0 1 0 1	1	1
4	1 0 1 0	1	0
5	1 1 0 1	0	1
6	1 1 1 0	1	0
7	0 1 1 1	1	1
8	1 0 1 1	0	1

Рис. 0.13. Последовательность работы линейного сдвигового регистра

На каждом шаге все содержимое регистра сдвигается вправо на один разряд. При этом можно получить в качестве результата один бит. На освободившееся слева место поступает бит, равный результату вычисления функции обратной связи $f = b_1 \oplus b_4$. Выходную последовательность генератора псевдослучайных бит образует последний столбец таблицы (извлекаемый бит).

Линейный сдвиговый регистр размером *n* бит может находиться в одном из 2^n-1 состояний. Поэтому теоретически такой регистр может генерировать псевдослучайную последовательность с максимальным периодом 2^n-1. Линейный сдвиговый регистр с обратной связью будет генерировать циклическую последовательность битов с максимальным периодом только при выборе в качестве отводной последовательности определенных бит. Разработана математическая теория, позволяющая выбрать подходящие номера разрядов для бит отводной последовательности.

Линейные сдвиговые регистры с обратной связью часто использовались и используются до сих пор при шифровании потоков данных. Для повышения криптостойкости в таких устройствах шифрования применяются комбинации нескольких сдвиговых регистров с обратной связью, а также вводятся дополнительные перемешивающие операции. Такие электронные схемы предлагались и выпускались еще до второй мировой войны. Аналогичные принципы заложены и в некоторые поточные шифры, созданные в конце XX-го века, например, в алгоритм A5, использовавшийся в Европе для шифрования сотовых цифровых каналов связи стандарта GSM. Несмотря на то, что некоторые криптоаналитики высказывают сомнения в надежности алгоритмов поточного шифрования с использованием линейных сдвиговых регистров с обратной связью, они положены в основу функционирования различных военных и гражданских устройств связи, используемых до настоящего времени.

Основным недостатком генераторов псевдослучайных чисел на базе линейных сдвиговых регистров является сложность программной реализации. Сдвиги и битовые операции легко и быстро выполняются в электронной аппаратуре,

поэтому в разных странах выпускаются микросхемы и устройства для поточного шифрования на базе алгоритмов с использованием сдвиговых регистров с обратной связью.

3 Задания для самостоятельной работы

1. Определите первые 16 бит псевдослучайной последовательности, получаемой с помощью генератора ПСЧ на основе сдвигового регистра, изображенного на рис. 0.14, если начальное значение регистра равно:

а) 1111; б) 1001;

в) 0001; в) 1110.

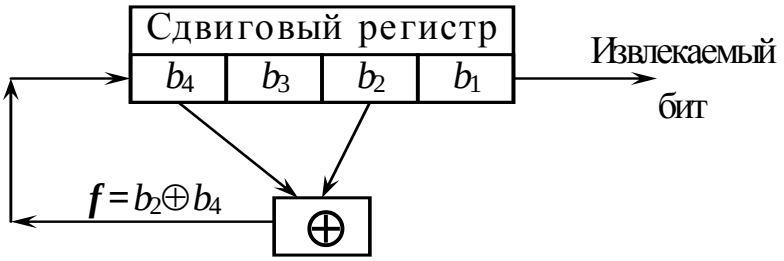


Рис. 0.14. 4-разрядный линейный сдвиговый регистр с обратной связью

2. Определите период псевдослучайных последовательностей, получаемых с помощью генератора ПСЧ на основе сдвигового регистра с обратной связью из задания 1.

Тема: Кодирование информации по Хэммингу

1 Теоретическая часть

При передаче информации могут возникать ошибки, обусловленные воздействием различных факторов. В этом случае для повышения достоверности передачи информации используют специальные коды.

Различают коды, обнаруживающие ошибки, и корректирующие коды, позволяющие обнаруживать место, где произошла ошибка, и исправлять ее. Простейшим способом обнаружения ошибок является проверка на четность. В этом случае к битам передаваемого или хранимого М-разрядного слова добавляется еще один бит — бит четности, значение которого подбирается таким образом, чтобы среди получившихся N разрядов $(N=M+1)$ обязательно было четное число единиц. Такой избыточный код позволяет лишь констатировать факт наличия ошибки в слове даже без указания места, где она произошла.

Принцип построения корректирующих кодов заключается в том, что к каждому хранимому или передаваемому М-разрядному слову добавляют K битов с соответствующим их расположением среди битов М-разрядного слова. Подобные N-разрядные коды $(N=M+K)$ были впервые рассмотрены в 1948 г. Р. Хеммингом и с тех пор обычно называются кодами Хемминга (N,M).

2 Практическая часть

Задача 1. Закодировать сообщение 1101101001110110 помехоустойчивым кодом Хемминга.

Решение.

Рассмотрим принцип построения кода Хемминга для 16-разрядных данных (M= 16), исправляющего все одиночные ошибки.

Сначала определим необходимое число проверочных разрядов *K [K=N-M]* из соотношения:

$$2^{N-M}-1 \geq N.$$

Откуда для *M*= 16 находим N=21 и *K=5.*

Верхние границы *M* и *N* для различных *K* представлены в таблице 3.

Таблица 3

K	1	2	3	4	5	6	7	8	9	10
M	0	1	4	11	26	57	120	247	502	1013
N	1	3	7	15	31	63	127	255	511	1023

Далее все биты N-разрядного слова нумеруются слева направо начиная с 1, при этом все биты, номера которых равны степени числа 2, являются битами четности, а остальные — информационными. Полученный таким образом код Хемминга для 16-разрядного слова данных представлен на рисунке 8.

Нумерация битов кода Хемминга для 16-разрядного слова данных

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	0	1	1	1	0	1	0	1	0	1	0	0	1	1	1	1	0	1	1	0
		15		14	13	12		11	10	9	8	7	6	5		4	3	2	1	0

Нумерация информационных битов 16-разрядного слова данных

Рисунок 8 - Код Хемминга для 16-разрядного слова данных

В 1, 2, 4, 8 и 16 разрядах данного кода располагаются биты четности, а в разрядах 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20 и 21 биты данных слова 16-разрядного исходного слова.

Каждый бит четности используется для контроля лишь определенных разрядов N-разрядного слова. Номера контролируемых разрядов для каждого бита четности приведены в таблица 4.

Таблица 4 – Номера битов четности и контролируемых ими разрядов слов в коде Хемминга

Контрольный разряд/номер бита кода Хемминга	Контролируемые разряды
1/1	1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33...
2/2	2 3 6 7 10 11 14 15 18 19 22 23 26 27 30 31 34...
3/4	4 5 6 7 12 13 14 15 20 21 22 23 28 29 30 31 36...
4/8	8 9 10 11 12 13 14 15 24 25 26 27 28 29 30 31 40...
5/16	16 17 18 1920 21 22 23 24 25 26 27 28 29 30 31 48...
6/32...	32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48...

Первый контрольный разряд контролирует разряды кода Хемминга с номерами
$a_j x2^j + a_{j-1} x2^{j-1} + ... + a_2 x2^2 + a_1 x2^1 + 1 x2^0$, где $j = 1og_2 N$, a_i— произвольное значение. Таким образом, первый контрольный разряд контролирует все нечетные номера.

Второй контрольный разряд контролирует разряды кода Хемминга с номерами
$a_j x2^j + a_{j-1} x2^{j-1} + ... + a_2 x2^2 + 1 x2^1 + a_0 x2^0$.

Соответственно третий контрольный разряд контролирует разряды кода Хемминга с номерами $a_j x2^j + a_{j-1} x2^{j-1} + ... + 1 x2^2 + a_1 x2^1 + a_0 x2^0$и т. д.

Как видно из табл. 2, в число контролируемых разрядов включается и тот разряд, где расположен сам бит четности. При этом содержимое бита четности устанавливается так, чтобы суммарное число единиц в контролируемых им разрядах было четным.

Таким образом в рассматриваемом примере (рисунок 8):

1) первый контрольный разряд в коде Хемминга равен 0, так как биты 3, 5, 7, 9, 11, 13, 15, 17, 19 и 21 содержат восемь единиц;

2) второй контрольный разряд равен 0, так как биты 3, 6, 7, 10,11. 14, 15, 18 и 19 содержат шесть единиц;

3) четвертый контрольный разряд равен 1., так как биты 5, 6, 7, 12. 13, 14, 15, 20 и 21 содержат пять единиц;

4) восьмой контрольный разряд равен 0, так как биты 9, 10, 11, 12, 13, 14 и 15 содержат четыре единицы;

5) шестнадцатый контрольный разряд равен 1, так как биты 17, 18, 19, 20 и 21 содержат три единицы.

Код, образованный значениями контрольных разрядов, называют дополнительным кодом. То есть для 16-разрядного кода данных 1101101001110110 дополнительный код равен **10100.**

Предположим теперь, что из-за воздействия каких-либо возмущающих факторов исчезнет единица в девятом разряде кода Хемминга (одиннадцатом информационном бите 16-разрядного слова данных) (рисунок 9).

Нумерация битов кода Хемминга для 16-разрядного слова данных

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	0	1	1	1	0	1	0	0	0	1	0	0	1	1	1	1	0	1	1	0
		15		14	13	12		11	10	9	8	7	6	5		4	3	2	1	0

Нумерация информационных битов 16-разрядного слова данных

Рисунок 9 - Код Хемминга для 16-разрядного слова данных с искаженным девятым разрядом (11-информационным)

Проверка образовавшегося кода дает следующий результат:

бит четности 2, 4 и 16 правильны, а биты четности 1 и 8 — неправильны.

Получение неправильного значения бита четности 1 указывает на то, что ошибка должна быть в одном из контролируемых им битов: 1,3,5,7,9, 11, 13, 15, 17, 19 или 21. Поскольку бит четности 2 правилен, то правильны и контролируемые им нечетные биты 3, 7, 11, 15 и 19, так что ошибка произошла не в них. Правильность контрольного бита 4 исключает возникновение ошибки в битах 5, 13 и 21, а правильность контрольного бита 16 — в бите 17. Следовательно, под подозрением остаются биты 1 и 9. Так как неправилен и бит

четности 8, который не контролирует бита с номером 1, но контролирует бит с номером 9, то можно сделать вывод об ошибочности бита 9. Инвертирование этого бита (изменение его значения с 0 на 1) исправляет положение — все биты четности становятся правильными.

Таким образом, номер искаженного разряда определяется суммой номеров неправильных битов четности. В нашем примере биты 1 и 8 неправильны, следовательно, искаженный разряд — 9 (9=1+8).

Задания для самостоятельной работы

Задание 1

Закодировать сообщение, используя алгоритм кода Хэмминга:

а) 10111000111110111
б) 110001100011101
в) 110111011101110011
г) 1110110111011100011011
д) 11111000111110001111

Задание 2

Определите верное ли получено сообщение, закодированное кодом Хэмминга. В случае необходимости исправьте ошибку.

а) 0101011000011000111110011011
б) 1101000010101100111011010101001
в) 0100111110111111011011
г) 1010110000101000011011
д) 101100111011011

Тема: Кодирование информации по Хаффману

1 Теоретическая часть

Код Хаффмана (Huffman code) это минимально-избыточный префиксный коде (minimum-redundancy prefix code).

Идея, лежащая в основе кода Хаффмана, достаточно проста. Вместо того чтобы кодировать все символы одинаковым числом бит (как это сделано, например, в ASCII кодировке, где на каждый символ отводится ровно по 8 бит), будем кодировать символы, которые встречаются чаще, меньшим числом бит, чем те, которые встречаются реже.

Первым такой алгоритм опубликовал Дэвид Хаффман (David Huffman) в 1952 году. Алгоритм Хаффмана двухпроходный. На первом проходе строится частотный словарь и генерируются коды. На втором проходе происходит непосредственно кодирование

Кодирование Хаффмана является простым алгоритмом для построения кодов переменной длины, имеющих минимальную среднюю длину. Этот весьма

популярный алгоритм служит основой многих компьютерных программ сжатия текстовой и графической информации. Некоторые из них используют непосредственно алгоритм Хаффмана, а другие берут его в качестве одной из ступеней многоуровневого процесса сжатия. Метод Алгоритм начинает строить кодовое дерево снизу вверх, затем скользит вниз по дереву, чтобы построить каждый индивидуальный код справа налево (от самого младшего бита к самому старшему). Начиная с работ Д.Хаффмана 1952 года, этот алгоритм являлся предметом многих исследований.

2 Практическая часть

Задача. Пусть алфавит состоит из 5-ти букв, с вероятностями вхождения в кодируемое сообщение равными

о	**а**	**с**	**в**	**л**
0.33	0.23	0.17	0.14	0.13

Определить коды букв в случае кодирования по методу Хаффмана.

Решение.
Последовательность расположения букв в списке – в порядке убывания вероятностей. Начало алгоритма заключается в объединении двух букв, имеющих минимальные вероятности, т.е. двух последних в нашем списке. Удаляем их из алфавита, а вместо них вставляем в алфавит слово **вл** с приписанной ему вероятностью, равной сумме вероятностей входящей в него букв. Снова ранжируем полученное по убыванию вероятностей.

о	**вл**	**а**	**с**
0.33	0.27	0.23	0.17

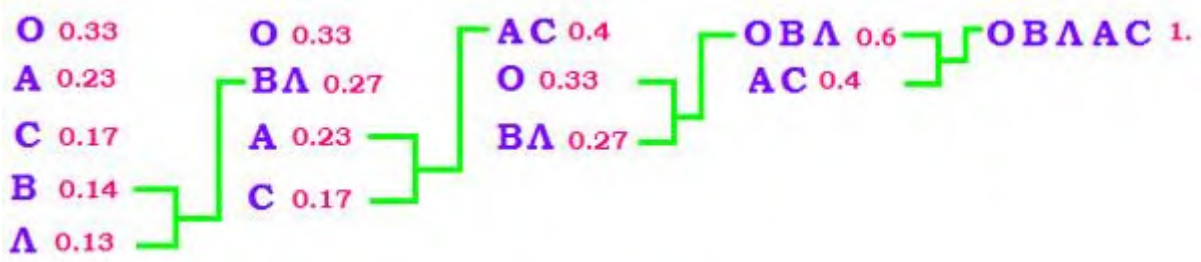
Удаляем две последние (имеющие минимальные вероятности) буквы из списка, а вместо них вставляем в него слово **ас** с приписанной ему вероятностью 0.4:

ас	**о**	**вл**
0.4	0.33	0.27

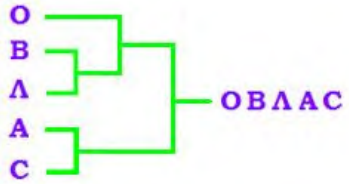
В полученном списке объединяем **о** и **вл**:

овл	**ас**
0.6	0.4

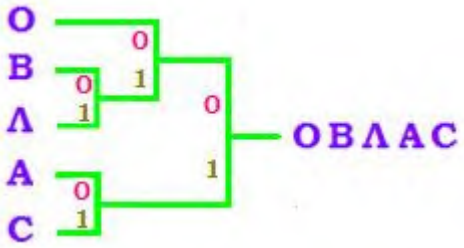
Последним шагом объединяем двухбуквенное и трехбуквенное слова в одно единое **овлас** с приписанной ему вероятностью 1.
Схематично:



Эту схему сокращенно записываем в виде дерева:



Теперь начинаем двигаться по этому дереву в обратном направлении – от корня (пятибуквенного слова) к листьям (буквам). При прохождении каждого узла верхнюю ветвь нумеруем 0, а нижнюю – 1:



В результате каждый лист дерева оказывается пронумерован:

о	в	л	а	с
00	010	011	10	11

Задания для самостоятельной работы

Задание 1.

Закодировать с помощью кода Хаффмана алфавит из 10 букв с вероятностями вхождения в кодируемое сообщение равными

Е	И	А	В	Д	К	З	Б	Г	Ж
0,211	0,207	1,173	0,099	0,084	0,082	0,041	0,038	0,037	0,028

Задание 2.

Закодировать с помощью кода Хаффмана следующее сообщение:

а) ааабббвввааагггггббввддддаааа;

б) пппрлллуууурррраааааллллуууупппаааууурпп;

в) ккннниииикккнндддиииккооонннн

Тема: Алгоритм RC4

1 Теоретическая часть

Алгоритм RC4 состоит из двух этапов. На первом, подготовительном этапе производится инициализация таблицы замен *S*. На втором, основном этапе вычисляются псевдослучайные числа.

Посмотрим, как инициализируется таблица *S*. В начале она заполняется последовательно числами от 0 до 15. Ключ представляется в виде последовательности 4-битовых слов, которыми заполняется другой массив *K*, такого же размера, как *S*. Если ключ оказался короче, чем надо, он повторяется нужное число раз. Затем выполняются следующие действия (*алгоритм 1*):

1. $j = 0$; $i = 0$;
2. $j = (j + S_i + K_i) \bmod 16$;
3. Поменять местами S_i и S_j.
4. $i = i + 1$;
5. Если $i < 16$, то перейти на п.2

В результате выполнения этого алгоритма производится начальное заполнение таблицы замен *S*, причем это начальное перемешивание значений производится в зависимости от секретного ключа.

После того, как таблица *S* подготовлена, можно начинать генерацию случайных *n*-битовых слов. Для этого счетчикам *i* и *j* присваивается начальное значение 0. Затем для получения каждого нового значения z_i выполняются следующие действия (*алгоритм 2*):

1. $i = (i + 1) \bmod 16$;
2. $j = (j + S_i) \bmod 16$;
3. Поменять местами S_i и S_j;
4. $a = (S_i + S_j) \bmod 16$;
5. $z_i = S_a$.

Полученное 4-битовое значение z_i может использоваться в качестве ключа для шифрования очередного 4-битового блока входного потока данных.

2 Практическая часть

Задача 1 Пусть секретный ключ состоит из шести 4-битовых значений (приведем их в десятичном виде): 1, 2, 3, 4, 5, 6. Попробуем сгенерировать последовательность чисел по алгоритму RC4.

Заполним таблицу *S* последовательно числами от 0 до 15.

Номер элемента	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Значение	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Затем подготовим таблицу *K*, записав в нее ключ необходимое количество раз:

Номер	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

элемента																
Значение	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4

Затем перемешаем содержимое таблицы *S*. Для этого будем использовать алгоритм 1, описанный выше. Процесс выполнения представим в виде трассировочной таблицы (рис. 4.5), в которой укажем все производимые действия. При выполнении вычислений необходимо помнить, что все операции сложения выполняются по модулю 16.

Номер пункта алгоритма	*Выполняемое действие (по mod 16)*	*Новое значение* ***i***	*Новое значение* ***j***
1	$j = 0$; $i = 0$	0	
2	$j = j + S_i + K_i = 0 + 0 + 1 = 1$		1
3	Поменять местами S_i и S_j, то есть S_0 и S_1		
4	$i = i + 1$	1	
5	$i < 16$, поэтому перейти на п.2		
2	$j = j + S_i + K_i = 1 + 1 + 1 = 3$		3
3	Поменять местами S_i и S_j, то есть S_1 и S_3		
4	$i = i + 1$	2	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (3 + 2 + 3) mod 16 = 8		8
3	Поменять местами S_i и S_j, то есть S_2 и S_8		
4	$i = i + 1$	3	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (8 + 0 + 4) mod 16 = 12		12
3	Поменять местами S_i и S_j, то есть S_3 и S_{12}		
4	$i = i + 1$	4	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (12 + 4 + 5) mod 16 = 5		5
3	Поменять местами S_i и S_j, то есть S_4 и S_5		
4	$i = i + 1$	5	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (5 + 4 + 6) mod 16 = 15		15
3	Поменять местами S_i и S_j, то есть S_5 и S_{15}		
4	$i = i + 1$	6	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (15 + 6 + 1) mod 16 = 6		6

3	Поменять местами S_i и S_j, то есть S_6 и S_6		
4	$i = i + 1$	7	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (6 + 7 + 2) mod 16 = 15		15
3	Поменять местами S_i и S_j, то есть S_7 и S_{15}		
4	$i = i + 1$	8	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (15 + 2 + 3) mod 16 = 4		4
3	Поменять местами S_i и S_j, то есть S_8 и S_4		
4	$i = i + 1$	9	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (4 + 9 + 4) mod 16 = 1		1
3	Поменять местами S_i и S_j, то есть S_9 и S_1		
4	$i = i + 1$	10	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (1 + 10 + 5) mod 16 = 0		0
3	Поменять местами S_i и S_j, то есть S_{10} и S_0		
4	$i = i + 1$	11	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (0 + 11 + 6) mod 16 = 1		1
3	Поменять местами S_i и S_j, то есть S_{11} и S_1		
4	$i = i + 1$	12	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (1 + 0 + 1) mod 16 = 2		2
3	Поменять местами S_i и S_j, то есть S_{12} и S_2		
4	$i = i + 1$	13	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (2 + 13 + 2) mod 16 = 1		1
3	Поменять местами S_i и S_j, то есть S_{13} и S_1		
4	$i = i + 1$	14	
5	$i < 16$, поэтому перейти на п.2		
2	$j = (j + S_i + K_i)$ mod 16= (1 + 14 + 3) mod 16 = 2		2
3	Поменять местами S_i и S_j, то есть S_{14} и S_2		
4	$i = i + 1$	15	
5	$i < 16$, поэтому перейти на п.2		

2	$j = (j + S_i + K_i)$ mod 16= (2 + 7 + 4) mod 16 = 13		13
3	Поменять местами S_i и S_j, то есть S_{15} и S_{13}		
4	$i = i + 1$	16	
5	$i < 16$ – неверно, поэтому закончить		

Рис. 0.15. Подготовительный этап алгоритма RC4

После выполнения алгоритма 1 получим инициализированную и подготовленную к основному этапу таблицу *S*:

Номер элемента	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Значение	10	13	14	12	2	15	6	4	5	3	1	9	8	7	0	11

После того, как таблица *S* подготовлена, можно начинать генерацию случайных 4-битовых слов. Вычислим первые 5 чисел псевдослучайной последовательности, используя алгоритм 2. Результаты вычисления последовательности значений также представим в виде таблицы (см. рис. 0.16)

	Выполняемое действие (по mod 16)	Новое значение ***i***	Новое значение ***j***	Новое значение ***a***
Вычисление z_1	1. $i = (i + 1) = 0+1=1$	1		
	2. $j = (j + S_i)$ mod 16=(0+13) mod 16=13		13	
	3. Поменять местами S_1 и S_{13}			
	4. $a = (S_i + S_j)$mod 16=(7+13)mod 16=4			4
	5. $\mathbf{z_1 = S_4 = 2}$			
Вычисление z_2	1. $i = (i + 1) = 1+1=2$	2		
	2. $j = (j + S_i)$ mod 16=(13+14) mod 16=11		11	
	3. Поменять местами S_2 и S_{11}			
	4. $a = (S_i + S_j)$mod 16=(9+14)mod 16=7			7
	5. $\mathbf{z_2 = S_7 = 4}$			
Вычисление z_3	1. $i = (i + 1) = 2+1=3$	3		
	2. $j = (j + S_i)$ mod 16=(11+12) mod 16=7		7	
	3. Поменять местами S_3 и S_7			
	4. $a = (S_i + S_j)$mod 16=(4+12)mod 16=0			0
	5. $\mathbf{z_3 = S_0 = 10}$			
Вычисление z_4	1. $i = (i + 1) = 3+1=4$	4		
	2. $j = (j + S_i)$ mod 16=(7+2) mod 16=9		9	
	3. Поменять местами S_4 и S_9			
	4. $a = (S_i + S_j)$mod 16=(3+2)mod 16=5			5
	5. $\mathbf{z_4 = S_5 = 15}$			
Вычисление	1. $i = (i + 1) = 4+1=5$	5		
	2. $j = (j + S_i)$ mod 16=(9+15) mod 16=8		8	
	3. Поменять местами S_5 и S_8			

z_5	4. $a = (S_i + S_j)$mod 16=(5+15)mod 16=4			4
	5. $\mathbf{z_5 = S_4 = 3}$			

Рис. 0.16. Основной этап (вычисление элементов псевдослучайной последовательности) алгоритма RC4.

В результате первые пять значений получились следующие: 7, 13, 2, 12, 5. При необходимости получения большего количества случайных чисел можно продолжить вычисления дальше. При n=4 генерируемые числа будут иметь размер 4 бита, то есть иметь значения от 0 до 15.

3 Задания для самостоятельной работы

Задание 1 Продолжите вычисление псевдослучайной последовательности по алгоритму RC4, и найдите z_6, z_7, z_8, z_9 и z_{10}

Тема: Реализации асимметричного алгоритма RSA

1 Теоретическая часть

Во многих задачах криптографии для заданных чисел c, m требуется находить такое число $d < m$, что

$$cd \bmod m = 1. \qquad (1)$$

Такое d существует тогда и только тогда, когда числа c и m взаимно простые. Число d, удовлетворяющее равенству $cd \bmod m = 1$, называется *инверсией c* по модулю m и часто обозначается $c^{-1} \bmod m$. Данное обозначение для инверсии связано с тем, что равенству $cd \bmod m = 1$ можно переписать в виде

$$cc^{-1} \bmod m = 1.$$

Таким образом, умножение на c^{-1} соответствует делению на c при вычислениях по модулю m.

Инверсию по модулю m также можно вычислять с помощью обобщенного алгоритма Евклида.

Алгоритм RSA шифрования с открытым ключом был предложен одним из первых в конце 70-х годов XX века. Его название составлено из первых букв фамилий авторов: Р.Ривеста (R.Rivest), А.Шамира (A.Shamir) и Л.Адлемана (L.Adleman). Алгоритм RSA является, возможно, наиболее популярным и широко применяемым асимметричным алгоритмом в криптографических системах.

Алгоритм основан на использовании того факта, что задача разложения большого числа на простые сомножители является вычислительно сложной. Криптографическая система RSA базируется на следующих двух фактах из теории чисел:

1) задача проверки числа на простоту является сравнительно легкой;

2) задача разложения чисел вида $n = pq$ (p и q — простые числа); на множители является очень трудной, если мы знаем только n, а p и q — большие числа (это так называемая *задача факторизации*).

Алгоритм RSA представляет собой блочный алгоритм шифрования, где зашифрованные и незашифрованные данные должны быть представлены в виде целых чисел между 0 и n -1 для некоторого n.

Шифрование

Итак, рассмотрим сам алгоритм. Пусть абонент А хочет передать зашифрованное сообщение абоненту Б. В этом случае абонент Б должен подготовить пару (открытый ключ; закрытый ключ) и отправить свой открытый ключ пользователю А.

Первым этапом является генерация открытого и закрытого ключей. Для этого вначале выбираются два больших простых числа P и Q. Затем вычисляется произведение N:

$$N = PQ.$$

Затем вычисляется вспомогательное число f:

$$f = (P - 1)(Q - 1)$$

После этого случайным образом выбирается число $e < f$ и взаимно простое с f.

Далее необходимо найти число d, такое, что

$$ed \bmod f = 1.$$

Числа e и N будут открытым ключом пользователя, а значение d – закрытым ключом. Так как пользователь Б хочет получить зашифрованное сообщение от пользователя А, значит пользователь Б должен отправить свой открытый ключ (e, N) пользователю А. Числа P и Q больше не нужны, однако их нельзя никому сообщать; лучше всего их вообще забыть.

На этом этап подготовки ключей закончен и можно использовать основной протокол RSA для шифрования данных.

Второй этап – шифрование данных. Если абонент А хочет передать некоторые данные абоненту Б, он должен представить свое сообщение в цифровом виде и разбить его на блоки m_1, m_2, m_3, ... , где $m_i < N$. Зашифрованное сообщение будет состоять из блоков c_i.

Абонент A шифрует каждый блок своего сообщения по формуле

$$c_i = m_i^e \bmod N,$$

используя открытые параметры пользователя B, и пересылает зашифрованное сообщение C=(c_1, c_2, c_3, ...) по открытой линии.

Абонент B, получивший зашифрованное сообщение, расшифровывает все блоки полученного сообщения по формуле

$$m_i = c^d \bmod N.$$

Все расшифрованные блоки будут точно такими же, как и исходящие от пользователя А.

Злоумышленник, перехватывающий все сообщения и знающий всю открытую информацию, не сможет найти исходное сообщение при больших значениях P и Q.

2 Практическая часть

Задача:. Равенство (1) означает, что для некоторого целого k имеет место равенство $cd - km = 1$. Учитывая, что c и d взаимно просты, можно преобразовать это равенство следующим образом:

$$m(-k) + cd = \text{НОД}(m,c).$$

Значит, мы можем вычислить c^{-1} mod m (или найти число d) с помощью обобщенного алгоритма Евклида. При этом значение переменной k нас не интересует. Если число d получается отрицательным, то нужно прибавить к нему m, так как по определению число a mod m то берется из множества $\{0,1,..., m - 1\}$.

Рассмотрим пример. Пусть m=9, c=5. Найдем 5^{-1} mod 9. Будем выполнять вычисления по обобщенному алгоритму Евклида, записывая все вычисления по шагам.

q	U			V			T		
	u_1	u_2	u_3	V_1	v_2	V_3	t_1	t_2	t_3
9 div 5 = 1	9	1	0	5	0	1	9 mod 5 = 0	1 - 1·0 = 1	0-1·1 = -1
5 div 4 =1	5	0	1	4	1	-1	5 mod 4=1	0-1·1=-1	1-1·(-1)= =2
4 div 1 =4	4	1	-1	1	-1	2	4 mod 1=0	1-4·0=1	-1-4·2=7
	1	-1	2	0	1	7			

Таким образом, получили, что 5^{-1} mod 9 = 2. Проверим:

$$5 \cdot 2 \text{ mod } 9 = 10 \text{ mod } 9 = 1.$$

Пример вычислений по алгоритмуRSA

Пусть пользователь А хочет передать пользователю Б сообщение. В этом случае вначале пользователь Б должен подготовить открытый и закрытый ключи. Пусть им выбраны, например, следующие параметры:

$$P = 3,\ Q = 11,\ N = 3 \cdot 11 = 33.$$

Тогда $f = (P - 1)(Q - 1) = (3-1)(11-1) = 20$.

Затем пользователь Б выбирает любое число e, не имеющее общих делителей с f (это необходимо для того, чтобы зашифрованное сообщение можно было потом однозначно восстановить). Пусть $e = 13$. Это число будет одним из компонентов открытого ключа.

Далее необходимо найти число d, которое можно будет использовать в качестве закрытого ключа для расшифрования сообщения. Значение d должно удовлетворять соотношению

$$ed \text{ mod } f = 1.$$

Для малых значений f число d можно найти подбором. В общем случае для поиска значения d нужно использовать обобщенный алгоритм Евклида, приведенный в приложении 1. В нашем случае подходит $d = 17$. (Проверяем:

$$13 \cdot 17 \text{ mod } 20 = 221 \text{ mod } 20 = 1.)$$

Теперь пользователь Б должен запомнить свой закрытый ключ 17, отправить открытый ключ (13, 33) пользователю А и уничтожить числа $P = 3$ и $Q = 11$.

Пользователь А, получивший открытый ключ (13, 33), увидев, что N=33, разбивает исходное сообщение на три блока, причем значение каждого меньше N: m_1=8, m_2=27, m_3,=5. Затем он шифрует каждый блок:

$$c_1 = 8^{13} \bmod 33 = 17,$$

$$c_2 = 27^{13} \bmod 33 = 15,$$

$$c_3 = 5^{13} \bmod 33 = 26$$

Зашифрованное сообщение, состоящее из трех блоков (17, 15, 26) передается пользователю Б, который, используя свой закрытый ключ d = 17 и N=33, расшифровывает сообщение:

$$m_1 = 17^{17} \bmod 33 = 8,$$

$$m_2 = 15^{17} \bmod 33 = 27,$$

$$m_3 = 26^{17} \bmod 33 = 5.$$

Таким образом, абонент Б расшифровал сообщение от абонента А.

3 Задания для самостоятельной работы

1. В системе RSA найдите недостающие параметры и опишите процесс передачи сообщения m пользователю Б, если пользователь Б выбрал следующие параметры P, Q, e:

а) $P = 3$, $Q = 11$, $e = 7$, $m = 10$; б) $P = 3$, $Q = 11$, $e = 13$, $m = 14$;
в) $P = 5$, $Q = 7$, $e = 13$, $m = 10$; г) $P = 5$, $Q = 9$, $e = 7$, $m = 8$;
д)$P = 5$, $Q = 9$, $e = 11$, $m = 17$; е)$P = 11$, $Q = 17$, $e = 13$, $m = 20$.

2. Сформируйте цифровую подпись по алгоритму RSA для сообщения m при следующих параметрах пользователя:

а) $P = 3$, $Q = 11$, $d = 17$, $m = 10$, б) $P = 5$, $Q = 7$, $d = 13$, $m = 10$,
в) $P = 3$, $Q =$ l 1, $d = 7$, $m = 15$, г) $P = 5$, $Q = 7$, $d = 29$, $m = 8$.

3. Абоненту Иванову от абонента Петрова пришло сообщение m, подписанное подписью Петрова c, сформированной по алгоритму RSA. Известны открытые ключи абонента Петрова: $N = 55$, $e = 7$. Проверьте его цифровую подпись для указанных m и c:

а) $m = 2$, $c = 19$; б) $m = 6$, $c = 42$; в) $m = 4$, $c = 49$.

4. Пользователю системы RSA с параметрами $N = 33$, $e = 3$ передано зашифрованное сообщение $c = 9$. Расшифруйте это сообщение, взломав систему RSA пользователя.

Тема: Применение различных функций хеширования, анализ особенностей хешей

1 Теоретическая часть

Хеш-функцией (hash function) называется функция, которая для строки произвольной длины вычисляет некоторое целое значение или некоторую другую строку фиксированной длины. Математически это можно записать так:

$$h = H(M),$$

где M – исходное сообщение, называемое иногда *прообразом*, а h – результат, называемый значением хеш-функции или хеш-значением.

Смысл хеш-функции состоит в определении характерного признака прообраза – значения хеш-функции. Это значение обычно имеет определенный фиксированный размер, например, 64 или 128 бит. Значение хеш-функции может быть в дальнейшем проанализировано для решения какой-либо задачи. Хеш-функции широко применяются в современной криптографии.

Простейшая хеш-функция может быть составлена с использованием операции «сумма по модулю 2» следующим образом: получаем входную строку, складываем все байты строки по модулю 2 и байт-результат возвращаем в качестве значения хеш-фукнции. Длина значения хеш-функции составит в этом случае 8 бит независимо от размера входного сообщения.

Сформулируем основные требования, предъявляемые к криптографическим хеш-функциям:

– хеш-функция должна быть применима к сообщению любого размера;

– вычисление значения функции должно выполняться достаточно быстро;

– при известном значении хеш-функции должно быть трудно (практически невозможно) найти подходящий прообраз M;

– при известном сообщении M должно быть трудно найти другое сообщение M' с таким же значением хеш-функции, как у исходного сообщения;

– должно быть трудно найти какую-либо пару случайных различных сообщений с одинаковым значением хеш-функции.

Создать хеш-функцию, которая удовлетворяет всем перечисленным требованиям – задача непростая. Необходимо также помнить, что на вход функции поступают данные произвольного размера, а хеш-результат не должен получаться одинаковым для данных разного размера.

В настоящее время на практике в качестве хеш-функций применяются функции, обрабатывающие входное сообщение блок за блоком и вычисляющие хеш-значение h_i для каждого блока M_i входного сообщения по зависимостям вида:

$$h_i = H(M_i, h_{i-1}),$$

где h_{i-1} – результат, полученный при вычислении хеш-функции для предыдущего блока входных данных.

В результате выход хеш-функции h_n является функцией от всех n блоков входного сообщения.

2 Практическая часть

Задача Пусть исходное сообщение, переведенное в цифровой вид, было следующим (в шестнадцатеричном формате):

3E 54 A0 1F B4

Переведем сообщение в двоичный вид, запишем байты друг под другом и сложим биты в каждом столбике по модулю 2:

0011 1110
0101 0100
1010 0000
0001 1111
1011 0100

0110 0101

Результат ($0110\ 0101_{(2)}$ или $65_{(16)}$) и будет значением хеш-функции.

Однако такую хеш-функцию нельзя использовать для криптографических целей, например для формирования электронной подписи, так как достаточно легко изменить содержание подписанного сообщения, не меняя значения контрольной суммы.

Поэтому рассмотренная хеш-функция не годится для криптографических применений. В криптографии хеш-функция считается хорошей, если трудно создать два прообраза с одинаковым значением хеш-функции, а также, если у выхода функции нет явной зависимости от входа.

3 Задания для самостоятельной работы

1. Пусть хеш-функция $y = h(x_1 x_2 \ldots x_n)$ определяется как результат выполнения операции «сумма по модулю 2» для всех байтов сообщения, представленного в двоичном виде. Для каждого из шести сообщений, записанных в левом столбце, найдите соответствующий результат вычисления хеш-функции из правого столбца. Все сообщения и значения хеш-функции представлены в шестнадцатеричном формате.

Сообщения	Значения хеш-функции
а) 0A3 69 2C;	а) 38;
б) 82 0F B5;	б) 1B;
в) 0DA 14 90;	в) 0F9;
г) 32 01 BF;	г) 8C;
д) 9E A6 23;	д) 0E6;
е) 10 BE 57;	е) 5E

Тема: Применение протокола Диффи-Хеллмана для обмена ключами шифрования

1 Теоретическая часть

Алгоритм основан на трудности вычислений дискретных логарифмов. Попробуем разобраться, что это такое. В этом алгоритме, как и во многих других алгоритмах с открытым ключом, вычисления производятся по модулю некоторого большого простого числа *P*. Вначале специальным образом подбирается некоторое натуральное число А, меньшее *P*. Если мы хотим зашифровать значение *X*, то вычисляем

$$Y = A^X \bmod P.$$

Причем, имея *X*, вычислить *Y* легко. Обратная задача вычисления *X* из *Y* является достаточно сложной. Экспонента *X* как раз и называется *дискретным логарифмом Y*. Таким образом, зная о сложности вычисления дискретного логарифма, число *Y* можно открыто передавать по любому каналу связи, так как при большом модуле *P* исходное значение *X* подобрать будет практически невозможно. На этом математическом факте основан алгоритм Диффи-Хеллмана для формирования ключа.

Формирование общего ключа

Пусть два пользователя, которых условно назовем пользователь 1 и пользователь 2, желают сформировать общий ключ для алгоритма симметричного шифрования. Вначале они должны выбрать большое простое число Р и некоторое специальное число *A*, $1 < A < P$-1, такое, что все числа из интервала [1, 2,..., *P*-l] могут быть представлены как различные степени значения *A* по модулю *P* (другими словами, $A^1 \bmod P$, $A^2 \bmod P$, $A^3 \bmod P$, …, $A^K \bmod P$ ($K < P$) – это разные числа из интервала [1, 2,..., *P*-l]). Числа *A* и *P* должны быть известны всем абонентам системы и могут выбираться открыто. Это будут, так называемые, *общие параметры*.

Затем первый пользователь выбирает число X_1 (X_1<P), которое желательно формировать с помощью датчика случайных чисел. Это будет закрытый ключ первого пользователя, и он должен держаться в секрете. На основе закрытого ключа пользователь 1 вычисляет число

$$Y_1 = A^{X_1} \bmod P,$$

которое он посылает второму абоненту.

Аналогично поступает и второй пользователь, генерируя X_2 и вычисляя

$$Y_2 = A^{X_2} \bmod P.$$

Это значение пользователь 2 отправляет первому пользователю.

После этого у пользователей должна быть информация, указанная в следующей таблице:

	Общие параметры	Открытый ключ	Закрытый ключ
Пользователь 1		Y_1	X_1
Пользователь 2	*P, A*	Y_2	X_2

Из чисел Y_1 и Y_2, а также своих закрытых ключей каждый из абонентов может сформировать общий секретный ключ Z для сеанса симметричного шифрования. Вот как это должен сделать первый пользователь:

$$Z = (Y_2)^{X_1} \bmod P.$$

Никто другой кроме пользователя 1 этого сделать не может, так как число X_1 секретно. Второй пользователь может получить то же самое число Z, используя свой закрытый ключ и открытый ключ своего абонента следующим образом:

$$Z = (Y_1)^{X_2} \bmod P.$$

Если весь протокол формирования общего секретного ключа выполнен верно, значения Z у одного и второго абонента должны получится одинаковыми. Причем, что самое важное, противник, не зная секретных чисел X_1 и X_2, не сможет вычислить число Z. Не зная X_1 и X_2, злоумышленник может попытаться вычислить Z, используя только передаваемые открыто P, A, Y_1 и Y_2. Безопасность формирования общего ключа в алгоритме Диффи-Хеллмана вытекает из того факта, что, хотя относительно легко вычислить экспоненты по модулю простого числа, очень трудно вычислить дискретные логарифмы. Для больших простых чисел размером сотни и тысячи бит задача считается неразрешимой, так как требует колоссальных затрат вычислительных ресурсов.
Пользователи 1 и 2 могут использовать значение Z в качестве секретного ключа для шифрования и расшифрования данных. Таким же образом любая пара абонентов может вычислить секретный ключ, известный только им.

2 Практическая часть

Пусть два абонента, желающие обмениваться через Интернет зашифрованными сообщениями, решили сформировать секретный ключ для очередного сеанса связи. Пусть они имеют следующие общие параметры:

$$P = 11,\ A = 7.$$

Каждый абонент выбирает секретное число X и вычисляет соответствующее ему открытое число Y. Пусть выбраны

$$X_1 = 3,\ X_2 = 9.$$

Вычисляем

$$Y_1 = 7^3 \bmod 11 = 2,$$
$$Y_2 = 7^9 \bmod 11 = 8.$$

Затем пользователи обмениваются открытыми ключами Y_1 и Y_2. После этого каждый из пользователей может вычислить общий секретный ключ:

Пользователь 1: $Z = 8^3 \bmod 11 = 6.$

Пользователь 2: $Z = 2^9 \bmod 11 = 6.$

Теперь они имеют общий ключ 6, который не передавался по каналу связи.

3 Задания для самостоятельной работы

1. Вычислите секретные ключи Y_1 , Y_1 и общий ключ Z для системы Диффи-Хеллмана с параметрами:

а) $P = 13$, $A = 7$, $X_1 = 4$, $X_2 = 5$; б) $P = 13$, $A = 7$, $X_1 = 3$, $X_2 = 10$;
в) $P = 19$, $A = 3$, $X_1 = 5$, $X_2 = 7$; г) $P = 19$, $A = 3$, $X_1 = 10$, $X_2 = 5$;
д) $P = 19$, $A = 13$, $X_1 = 5$, $X_2 = 7$; е) $P = 19$, $A = 13$, $X_1 = 3$, $X_2 = 10$.

Тема: Применение протокола Эль-Гамаля для обмена ключами шифрования

-

1 Теоретическая часть

Сообщение, предназначенное для шифрования, должно быть представлено в виде одного числа или набора чисел, каждое из которых меньше P. Пусть пользователь 1 хочет передать пользователю 2 сообщение m. В этом случае последовательность действий следующая.

1. Первый пользователь выбирает случайное число k, взаимно простое с P-1, и вычисляет числа

$$r = A^k \bmod P,\ e = m \cdot Y_2^{\,k} \bmod P,$$

где Y_2 – открытый ключ пользователя 2. Число k держится в секрете.

2. Пара чисел (r, e), являющаяся шифротекстом, передается второму пользователю.

3. Второй пользователь, получив (r,e), для расшифрования сообщения вычисляет

$$m = e \cdot r^{P-1-X_2} \bmod P,$$

где X_2 – закрытый ключ пользователя 2. В результате он получает исходное сообщение m.

Если злоумышленник узнает или перехватит P, A, Y_2, r, e, то он не сможет по ним раскрыть m. Это связано с тем, что противник не знает параметр k, выбранный первым пользователем для шифрования сообщения m. Вычислить каким-либо образом число k практически невозможно, так как это задача дискретного логарифмирования. Следовательно, злоумышленник не может вычислить и значение m, так как m было умножено на неизвестное ему число. Противник также не может воспроизвести действия законного получателя сообщения (второго абонента), так как ему не известен закрытый ключ X_2 (вычисление X_2 на основании Y_2 — также задача дискретного логарифмирования).

По аналогичному алгоритму может производиться и согласование ключа, используемого для симметричного шифрования больших объемов данных. Более того, алгоритм Эль-Гамаля на практике целесообразно использовать именно для согласования общего ключа сессии, а не прямого шифрования больших сообщений. Это связано с тем, что в алгоритме используются операции возведения в степень и умножения по большому модулю. Так же как и в

алгоритмах RSA и Диффи-Хеллмана, операции производятся над большими, состоящими из нескольких сотен или тысяч бит, числами. Поэтому шифрование больших сообщений производится крайне медленно, кроме того, длина зашифрованного сообщения в два раза превышает длину открытого текста.

2 Практическая часть

Пусть два абонента, обменивающиеся через Интернет зашифрованными сообщениями, имеют следующие общие параметры:

$$P = 11,\ A = 7.$$

Кроме того, пользователи 1 и 2 имеют пары закрытых и открытых ключей, вычисляемые так же, как в п. 5.3.3:

Пользователь 1: закрытый ключ $X_1 = 3$, открытый ключ $Y_1 = 7^3 \bmod 11 = 2$,

Пользователь 2: закрытый ключ $X_2 = 9$, открытый ключ $Y_2 = 7^9 \bmod 11 = 8$.

Первый абонент желает передать второму сообщение. Для этого первый абонент запрашивает из центра распределения ключей открытый ключ второго абонента $Y_2 = 8$. Теперь он может зашифровать свое сообщение, которое в числовом виде пусть имеет значение m=9.

Первый абонент выбирает случайно число k, например $k = 7$. k должно быть взаимно простым с P-1. Значение $k = 7$ не имеет общих делителей с P-1=10, значит оно нам подходит. Первый абонент шифрует свое сообщение по формулам:

$$r = A^k \bmod P = 7^7 \bmod 11 = 6,$$

$$e = m \cdot Y_2^{\,k} \bmod P = 9 \cdot 8^7 \bmod 11 = 7$$

Пара чисел (6, 7) будет представлять собой шифротекст и передается второму пользователю. Второй пользователь, получив (6,7) и используя свой закрытый ключ $X_2 = 9$ для расшифрования сообщения, вычисляет

$$m = e \cdot r^{P-1-X_2} \bmod P = 7 \cdot 6^{11-1-9} \bmod 11 = 7 \cdot 6^1 \bmod 11 = 9\ .$$

В результате он действительно получает исходное сообщение m.

2 Задания для самостоятельной работы

1. В системе связи, применяющей шифр Эль-Гамаля, пользователь 1 желает передать сообщение m пользователю 2. Найдите недостающие параметры и опишите процесс передачи сообщения при следующих заданных параметрах P, A, X_2, k, m:

а) $P = 11$, $A = 7$, $X_2 = 9$, $k = 7$, $m = 8$, б) $P = 11$, $A = 7$, $X_2 = 3$, $k = 5$, $m = 8$,
в) $P = 17$, $A = 3$, $X_2 = 11$, $k = 7$, $m = 10$, г) $P = 19$, $A = 2$, $X_2 = 3$, $k = 5$, $m = 10$,
д) $P = 19$, $A = 7$, $X_2 = 8$, $k = 5$, $m = 10$, е) P=29, A=2, $X_2 = 13$, $k = 5$, $m = 10$,

2. Абоненты некоторой сети применяют цифровую подпись по алгоритму Эль-Гамаля с общими параметрами $P = 17$, $A = 3$. Для указанных секретных параметров X и k абонента Петрова найдите его открытый ключ и вычислите цифровую подпись для заданного сообщения m:

а) $X = 3$, $k = 9$, $m = 5$; б) $X = 5$, $k = 9$, $m = 7$;
в) $X = 3$, $k = 7$, $m = 11$; г) $X = 11$, $k = 3$, $m = 10$;
д) $X = 13$, $k = 11$, $m = 9$ е) $X = 17$, $k = 11$, $m = 15$.

МИНОБРНАУКИ России
ФГБОУ ВО «Тульский государственный университет»
Технический колледж им. С.И. Мосина

Методические рекомендации по выполнению и защите рефератов

по дисциплине: «Основы информационной безопасности»

по специальности:

10.02.05 «Обеспечение информационной безопасности автоматизированных систем»

Тула 2024

УТВЕРЖДЕНЫ
Цикловой комиссий информационных технологий
Протокол от «12» 09 2024 г. № 2
Председатель цикловой комиссии И.В. Миляева

Составитель Суворова Эрминэ Айказовна, преподаватель
Технического колледжа им. С.И. Мосина

ВВЕДЕНИЕ

Реферат - краткое изложение в письменном виде и в форме публичного выступления содержания книги, научной работы, результатов исследования.

Реферат является самостоятельной письменной работы студента. Реферат - работа, касающаяся какой-то одной достаточно узкой темы и обозначающая основные общепринятые точки зрения на данную тему. В реферате необходимо осветить конкретный вопрос, при этом не требуется наличия большого фактического материала, глубокого анализа и фундаментальных выводов. [1]

Основной задачей при подготовке реферата является создание ситуаций максимально приближенной, для подготовки студента к последующей защите выпускной квалификационной работы [2]. Структурно задача делиться на два этапа:

- Самостоятельная подготовка, во время которой студенты самостоятельно определяют круг первоочередных задач, составляют план работы, докладывают о выполнении плана преподавателю, принимают решение о готовности к выступлению.
- Публичная защита или доклад о результатах исследования, ознакомление аудитории с использованными методами решения, для закрепления у обучаемых навыков выступления перед группами лиц.

1 Структура реферата

Реферат выполняется в строгом соответствии с ГОСТ 7.32-2001 «Отчет о научно-исследовательской работе. Структура и правила оформления». Должен включать оглавление, введение, несколько глав (2-3), заключение и список использованных источников. [1,3,5]

Во введении реферата должны быть: актуальность темы реферата; цель работы; задачи, которые нужно решить, чтобы достигнуть указанной цели; краткая характеристика структуры реферата (введение, три главы, заключение и библиография); краткая характеристика использованной литературы.

Объем введения для реферата - 1-1,5 страницы.

Главы реферата могут делиться на пункты и подпункты, рекомендуется заканчивать выводами.

В заключении должны быть ответы, на поставленные во введении задачи и дан общий вывод. Объем заключения реферата - 1-1,5 страницы.

Общий объём реферата составляет 18-24 страницы.

Список использованных источников для реферата должен включать не менее 5 (пяти) позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы.

У реферата могут быть приложения - рисунки, схемы, слайды презентации и прочее.

2 Оформление реферата

Текст печатается на белой бумаге формата А4 в книжной ориентации. Используется шрифт: обычный - Times New Roman размером 14 пунктов, интервал 1,5, отступ для абзаца 1,25 см. Цвет шрифта черный. Выбор шрифта и интервала не случаен: Times New Roman – один из наиболее удобных и легких для чтения шрифтов, а полуторный интервал оптимален для восприятия текста. Текст необходимо размещать только на одной стороне листа. Поля оформляются следующим образом: верхнее, нижнее — 20мм, правое — 10 мм, левое поле необходимо для переплета, поэтому оно шире — 30 мм. Нумерация учитывает все страницы, но на титульном листе и на содержании номера страниц не проставляются. На всех остальных листах номер обозначается внизу посередине арабскими цифрами. Если в основном тексте используются формулы, они должны набираться в редакторе формул Microsoft Equation в размере, соответствующем остальному тексту. На рисунке 1 представлен образец настройки параметров страницы.

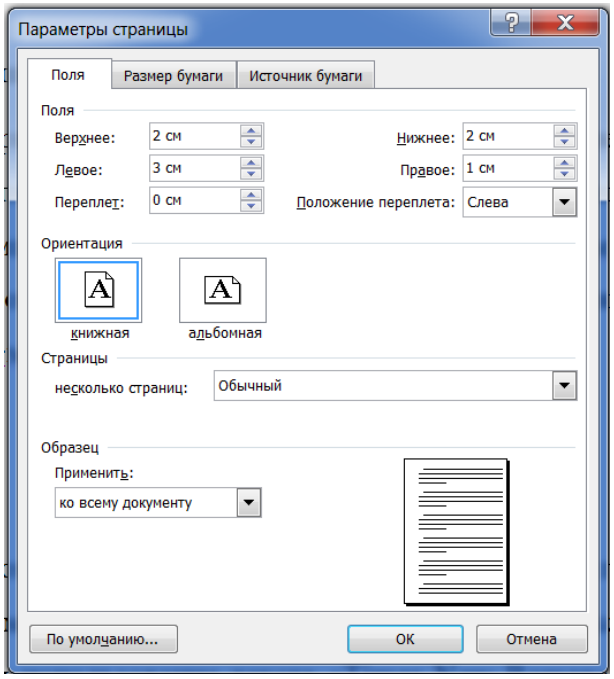


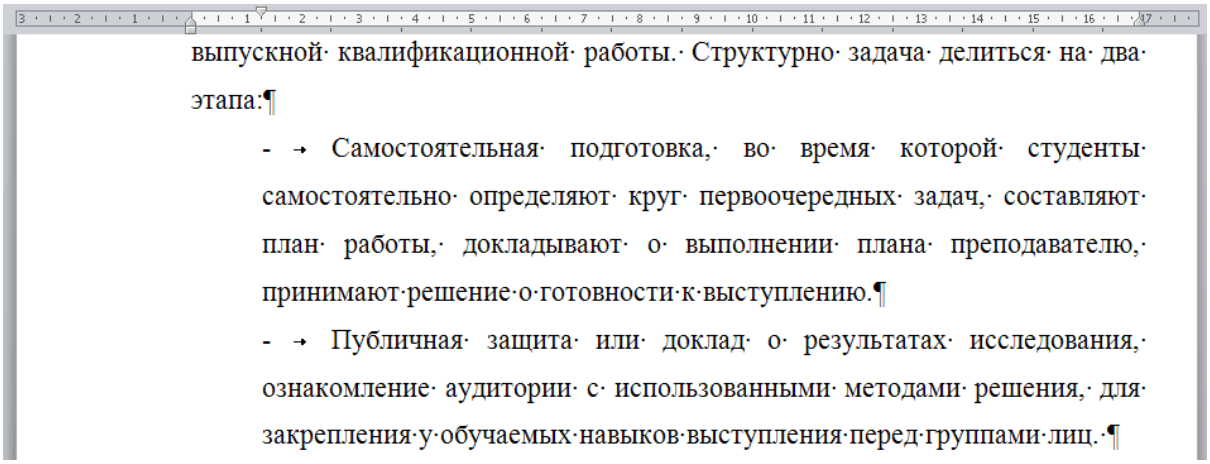
Рисунок 1 - Образец настройки параметров страницы

Допускается использование текста «Times New Roman» с меньшим размером кегля, то есть 8-13 пунктов, при оформлении текста таблиц, пояснительных надписей на рисунках, схемах, диаграммах.

Каждая из частей реферата начинается с новой страницы. Заголовки без нумерации пишутся заглавными буквами и размещаются по центру строки. Заголовки с нумерацией пишется строчными буквами с заглавной, размещается «по ширине страницы» и с отступом красной строки. Между заголовком и последующим текстом оставляется пустая строка.

Главы реферата могут делиться на пункты. Точка после номера не ставится. Номер пункта реферата включает номер соответствующей главы, отделяемый от собственного номера точкой, например: «1.3». Заголовки не должны иметь переносов и подчеркиваний, но допускается выделять их «жирностью» или курсивом. Между заголовком (названием главы) и подзаголовком (названием пункта) оставляется две строки.

Текст реферата, размещается с центрированием «по ширине страницы». Абзацы выделяются красной строкой с отступом не менее 1,25-1,27 см. внутри пунктов могут быть перечисления, перед каждой позицией ставиться дефис или при необходимости ссылки в тексте документа на одно из перечислений, строчную букву, после которой ставиться скобка. Пример внешнего вида набора текста показан на рисунке 2.


выпускной· квалификационной· работы.· Структурно· задача· делиться· на· два· этапа:¶

- → Самостоятельная· подготовка,· во· время· которой· студенты· самостоятельно· определяют· круг· первоочередных· задач,· составляют· план· работы,· докладывают· о· выполнении· плана· преподавателю,· принимают·решение·о·готовности·к·выступлению.¶
- → Публичная· защита· или· доклад· о· результатах· исследования,· ознакомление· аудитории· с· использованными· методами· решения,· для· закрепления·у·обучаемых·навыков·выступления·перед·группами·лиц.·¶

Рисунок 2 - Пример внешнего вида набора текста

Нумерация рисунков, таблиц и формул обозначается арабскими цифрами может быть сквозной или в пределах главы. Рисунки (схемы, диаграммы) сопровождаются пояснительными подписями. При этом подпись размещается по центру страницы, сокращение слова рисунок (Рис.) не допускается. Название рисунка следует через дефис. Точка в конце названия не ставиться, если название состоит из 2 и более предложений, то они разделяются точками. Рисунки помещаются после первого упоминания в тексте, или на следующей странице. На все рисунки должны быть ссылки в тексте. Между рисунком и текстом оставляется пустая строка. Например: образец оформления рисунка представлен на рисунке 3.

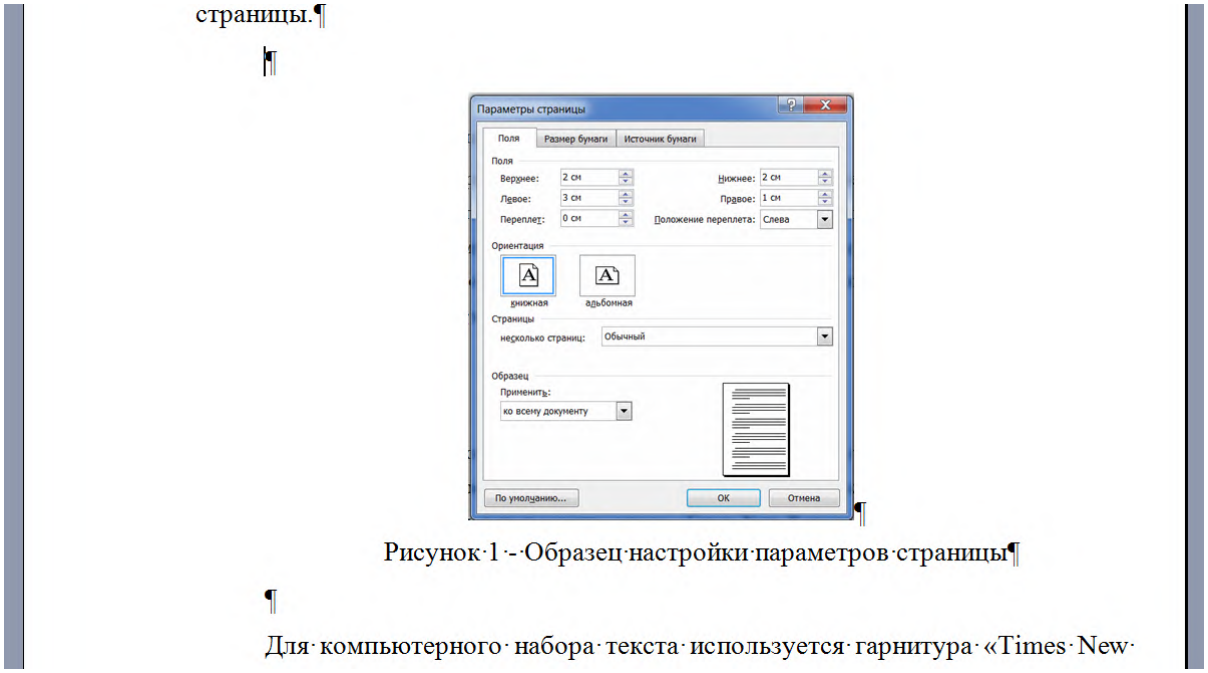


Рисунок 3 – Образец оформления рисунка

Статистический материал рекомендуется оформлять в виде таблицы. Таблицу помещают после первого упоминания в тексте. Над левым верхним углом таблице помещается надпись "Таблица" с указанием ее порядкового номера. Таблицы нумеруются последовательно арабскими цифрами или в пределах главы. Затем следует заголовок таблицы. При ссылке на таблицу указывается ее номер, например: (таблица 1 или таблица 2.3). Таблицы помещаются после первого упоминания в тексте, или на следующей странице,

сокращение слова таблица (Табл.) не допускается. Образец оформления таблицы представлен на рисунке 4.

Результаты·распределения·участников·«Конкурса»·по·рабочим·группам·представлены·в·таблице·5.¶

¶

Таблица·5·-·Распределение·участников·«Конкурса»·по·рабочим·группам¶

№¶ п/п¤	инженер·-·начальник Отдела·Защиты·Информации¤	старший·техник·по·программно-аппаратной·защите¤	техник·по·инженерно-технической·защите¤	техник¤
1¤	Акользина·Ирина·Сергеевна¤	Королев·Александр·Евгеньевич¤	Есипов·Евгений·Игоревич¤	Антошина·Оксана·Александровна·¤
2¤	Белоголовский·Денис·Сергеевич¤	Кудинова·Надежда·Андреевна¤	Арсеньев·Владислав·Александрович¤	Черковский·Никита·Янович·¤
3¤	Большаков·Максим·Андреевич¤	Власова·Юлия·Семёновна¤	Аминов·Ян·Владиславович¤	Волкова·Анастасия·Викторовна·¤
4¤	Власкин·Павел·Сергеевич¤	Тузов·Михаил·Юрьевич¤	Фадина·Ольга·Леонидовна¤	Глибин·Александр·Викторович·¤

¶

Рисунок 4 – Образец оформления таблицы

Математические формулы и зависимости размещаются непосредственно в тексте, нумеруются последовательно арабскими цифрами или в пределах главы. При ссылке на формулу указывается ее номер, например: (формула 1 или формула 2.3).

Обозначения символов используемых в формуле приводиться непосредственно под формулой. При этом используется правила для оформления примечаний, символы располагаются последовательно, текст набирается размером 12 pt, междустрочный интервал равен единице. Формулы следующие одна за другой и не разделённые текстом, разделяют запятой. Образец оформления формулы представлен на рисунке 5.

Для оценки глубины распространения оседающего аэрозоля, образованного линейным источником, используем формулу

$$\Gamma_1 \approx 3{,}5 \cdot 10^{-7} \beta \exp\left\{0{,}2 \ln\left[\frac{2 \cdot 10^6 k_{B2} \cdot G_{ое}}{\beta \cdot l}\right] - \ln \Delta_{доп}\right\}^{0{,}5}, \qquad (2.3)$$

где: H – высота выливания ОВ, м; $\bar{u}(H)$ – средняя интегральная скорость ветра в слое от по-

Рисунок 5 – Образец оформления формул

Материал, дополняющий текст работы, размещается в приложениях. Приложениями могут быть таблицы, схемы, диаграммы, чертежи, расчеты и т.д. Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ.

Пример - ПРИЛОЖЕНИЕ А

Каждое приложение следует начинать с новой страницы. Вверху первой страницы каждого приложения посередине рабочей строки прописными буквами печатают слово «ПРИЛОЖЕНИЕ» и его обозначение. Приложение должно иметь заголовок, который записывают по центру рабочей строки с прописной буквы отдельной строкой.

На все точные числовые данные, прямые цитаты и определения, требуются ссылки на список использованных источников. Обозначаются в тексте реферата в квадратных скобках с указанием номера источника по списку литературы (рисунок 6)

большого фактического материала, глубокого анализа и фундаментальных выводов. [1]

Основной задачей при подготовке реферата является создание ситуаций максимально приближенной, для подготовки студента к последующей защите

Рисунок 6 - Обозначение ссылки на список использованных источников

Список использованных источников для реферата обычно должен включать 5-12 позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы. Источники указываются в той же последовательности, в которой они располагаются по тексту.

Образец заполнения списка использованных источников представлен на рисунке 7.

Список использованных источников

1. **ГОСТ 7.32-2001** «Отчет о научно-исследовательской работе. Структура и правила оформления» [Текст]. – Введ. 2001-05-22. – М. : Госстандарт России : Изд-во стандартов, 2001.

2. **Желнова О.А.** Основные компетентности подготовки специалистов в системе среднего профессионального образования [Электронный ресурс] : науч. журн. / ISSN 1812-7339, 2008. – Режим доступа : http://www.rae.ru/fs/?section=content&op=show_article&article_id=7781137

3. Порядок представления и правила оформления рукописей статей [Электронный ресурс] : науч. журн. / «Известия ТулГУ», 2012. – Режим доступа : https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CC4QFjAD&url=http%3A%2F%2Fpublishing.tsu.tula.ru%2Fdocs%2Fru

Рисунок 7 - Образец заполнения списка использованных источников

Обратите внимание, что при указании интернет ресурса, обязательно указывается и его название и электронный адрес.

3 Оформление текста доклада

Для и изложения и публичной защиты результатов исследования по теме реферата рекомендуется подготовить устный доклад продолжительностью 5-6 минут. В докладе следует отразить тему и цель исследования, основные задачи, использованные методы и способы их решения, желательно указать источники и авторов обладающих авторитетом в исследуемой области. Раскрыть и обосновать суть работы.

В виду того, что текст доклада является личным вспомогательным материалом докладчика, материал может быть представлен в любом виде (текста, схемы, диаграммы, таблицы и т.д.). Текст доклада формируется из расчета 3 листа формата А4, 16 pt, междустрочный интервал 1, на 5-6 минут доклада, спокойного размеренного монолога.

Текст доклада рекомендуется начинать со следующей формулировки:

Вашему вниманию представляется доклад по теме …..

Задача докладчика, в своём выступлении отразить суть доклада и быть готовым отвечать на возможные вопросы по раскрытой тематике.

По окончании выступления, обучаемый должен подвести вывод, в котором передаётся суть основной мысли доклада.

4 Мультимедийное сопровождение защиты реферата

При изложении и публичной защите результатов исследования поощряется использование средств мультимедийного сопровождения, в виде слайдов, схем, видеороликов и презентаций. Для более полной наглядности отображения сопутствующих материалов, поощряется использование помощников.

Слайды презентации должны отвечать общим требованиям оформления реферата, но с небольшими поправками, обусловленными незначительным искажением текста и графических объектов при отображении с помощью LCD-проектора.

Текст для оформления слайдов должен отвечать следующим требованиям, используется шрифт «Arial», для заголовков высота 24 pt, для текста слайда 20 pt. Текст располагается с междустрочным интервалам равным единице, распределённым по ширине слайда. Слайды за исключением первого (титульного) нумеруются сквозной нумерацией в правом верхнем углу. Фон и цвет текста подбирается таким образом что бы исключить затруднение в распознавании текста или графических объектов.

Использование анимации приветствуется в случаях отображения некоторых функциональных особенностей изделий, выделения текста, какой либо области слайда или показе видеофайлов. В других случаях использование анимации нежелательно.

На доклад протяжённостью 5-6 минут, составляется 4-6 слайдов, включая титульный.

5 Регламент защиты реферата

Перед заслушиванием, преподаватель даёт одну-две минуты докладчику и его помощнику для подготовки к выступлению, запуску и проверке презентации на предмет её готовности к показу, положению трибуны, наличию указки других средств необходимых для проведения доклада.

По мере готовности преподаватель представляет слово докладчику. При этом один из помощников выступающего является секретарём, задача которого заключается в фиксировании времени начала и конца доклада, а так же конспектировании вопросов и ответов по окончании доклада. Второй помощник выступает в качестве оператора и меняет раздаточный материал или слайды презентации.

По окончании доклада, преподаватель предоставляет 1-2 минуты выступающему для отдыха и подготовки к ответу на вопросы, затем открывает дискуссию по теме доклада. В случаях абсолютной индифферентности аудитории преподаватель начинает задавать вопросы самостоятельно, с целью выяснить уровень знаний докладчика по теме.

В случае затруднения докладчика в ответе на вопрос, ответить может любой студент из аудитории, таким образом аудитория может перейти к открытой дискуссии.

Подводя итог выступления, преподаватель обязан дать оценку выступлениям докладчиков, и наиболее активных участников, в том числе секретаря.

6 Критерии оценки докладчика

Критерии оценивания докладчика складываются из требований к оформлению реферата, слайдов презентации, качеству представления материала на выступлении, ответов на вопросы по докладу [2].

При этом, тема, цели и задачи доклада должны соответствовать теме выступления. Акцентирование внимания на основных идеях соответствующих целям и задачам. Язык изложения материала понятен аудитории и вызывает интерес у аудитории. Очевидная актуальность, точность и полезность содержания, обосновывается и подтверждается достоверной информацией об исторических справках и текущих событиях.

Использование в презентации графических иллюстраций, статистики, диаграмм и графиков, экспертных оценок, примеров, сравнений, цитат.

Тематическая последовательность, логика и переходы выступления от вступления к основной части, от одной основной идеи (части) к другой, от одного слайда к другому.

В заключении повторение основных целей и задач выступления, подведение итогов. Допускается короткое и запоминающееся высказывание в конце (цитата, афоризм).

6.1 Выступление

Оценка «отлично» подразумевает свободное знание текста доклада и ориентацию в сопутствующем мультимедийном сопровождении, внятное и чёткое изложение доклада.

Оценка «хорошо» подразумевает знание текста доклада и ориентацию в сопутствующем мультимедийном сопровождении, изложение материала производится чтением доклада.

Оценка «удовлетворительно» подразумевает слабое знание текста доклада, отсутствие мультимедийного сопровождения, изложение материала

производится чтением доклада с ошибками, не правильной постановкой ударений.

Оценка «неудовлетворительно» подразумевает не знание материала и отсутствие мультимедийного сопровождения, путанное и нечёткое изложение доклада.

6.2 Оформление реферата

Оценка «отлично» подразумевает полное раскрытие темы исследования, приведение примеров из истории или современности, практическую направленность работы, строгое соответствие оформления текста по ГОСТ «Отчёт о НИР».

Оценка «хорошо» подразумевает раскрытие темы исследования, практическую направленность работы, незначительные ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

Оценка «удовлетворительно» подразумевает незначительные отклонения от темы исследования, практическую направленность работы, ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

Оценка «неудовлетворительно» подразумевает выполнение работы не по теме исследования, грубейшие ошибки в оформлении текста по ГОСТ «Отчёт о НИР».

6.3 Оформление презентации

Оценка «отлично» подразумевает наличие презентации соответствующей рекомендациям указанным в пункте 4.

Оценка «хорошо» подразумевает наличие презентации с незначительными отклонениями от рекомендаций указанным в пункте 4.

Оценка «удовлетворительно» подразумевает наличие презентации выполненной не по рекомендациям указанным в пункте 4.

Оценка «неудовлетворительно» отсутствует мультимедийное сопровождение и раздаточный материал.

6.4 Ответы на вопросы аудитории

Оценка «отлично» подразумевает свободное знание материала, быстрые и точные ответы на поставленные вопросы, приведение примеров из истории или современности.

Оценка «хорошо» подразумевает знание материала, правильные ответы на поставленные вопросы.

Оценка «удовлетворительно» подразумевает слабое знание материала, ответы на поставленные вопросы слабые и неуверенные.

Оценка «неудовлетворительно» подразумевает не знание материала отсутствие ответов на вопросы аудитории.

Кроме выступающего, в процессе дискуссии любой участник обсуждения задав своевременный, правильно обоснованный вопрос может быть оценен положительной оценкой.

7 Темы рефератов

1. Основные задачи информационной безопасности
2. Автоматизированные и автоматические системы управления и контроля
3. Руководящие документы в сфере защиты информации
4. Виды обеспечения информационной безопасности
5. Организационное обеспечение информационной безопасности
6. Методическое обеспечение информационной безопасности
7. Правовое обеспечение информационной безопасности
8. Инженерно-техническое обеспечение информационной безопасности
9. Программно-аппаратное обеспечение информационной безопасности
10. Кадровое обеспечение информационной безопасности
11. Информационные источники, интернет ресурс, обеспечение информационной безопасности
12. Вредоносное программное обеспечение, вирусы, «логические бомбы», «баннеры»
13. Администрирование и разграничение прав доступа, администратор ЛВС, владелец (хозяин) ЛВС, пользователь ЭВМ (ЛВС), VIP-пользователь ЭВМ (ЛВС), «гость» в ЭВМ (ЛВС)
14. Политика конфиденциальности и безопасности
15. Исполнительные органы власти (органы надзора) в сфере Информационной безопасности ФСТЭК, ФСБ, УВД отдел «К»
16. Преступления в сфере информационных технологий
17. Био-информационная система
18. Динамическая и статическая системы защиты информации
19. Организация защиты информации в автоматизированной системе управления.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1.1 Основные источники:

1. Галатенко, В. А. Основы информационной безопасности : учебное пособие / В. А. Галатенко. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 266 с. — ISBN 978-5-4497-0675-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/97562.html— Режим доступа: для авторизир. пользователей

2. Нестеров, С.А. Основы информационной безопасности. [Электронный ресурс] — Электрон.дан. — СПб. : Лань, 2017. — 324 с. — Режим доступа: http://e.lanbook.com/book/90153 — Загл. с экрана.

3. Мельников В.П.Информационная безопасность : учебник / В.П. Мельников, А.И. Куприянов. — Москва : КноРус, 2018. — 267 с. — Для СПО. — ISBN 978-5-406-05072-9.- Режим доступа: https://www.book.ru/book/924214, по паролю

4. Внуков, А. А. Основы информационной безопасности: защита информации : учеб. пособие для СПО / А. А. Внуков. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 240 с. — (Серия : Профессиональное образование). — ISBN 978-5-534-10711-1.— Режим доступа: https://www.biblio-online.ru/book/osnovy-informacionnoy-bezopasnosti-zaschita-informacii-431332

5. Литвиненко, В.И. Основы информационной безопасности : учебное пособие / Литвиненко В.И., Козлов Е.С. — Москва : КноРус, 2020. — 199 с. — ISBN 978-5-406-00904-8. Текст : электронный // ЭБС Book.ru [сайт]. —URL: https://book.ru/book/934627

1.2. Дополнительные источники:

1. Организационное и правовое обеспечение информационной безопасности : учебник и практикум для среднего профессионального образования / Т. А. Полякова, А. А. Стрельцов, С. Г. Чубукова, В. А. Ниесов ; ответственный редактор Т. А. Полякова, А. А. Стрельцов. — Москва : Издательство Юрайт, 2020. — 325 с. — (Профессиональное образование). — ISBN 978-5-534-00843-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451933

2. Горбенко, А. О. Основы информационной безопасности (введение в профессию) : учебное пособие / А. О. Горбенко. — Санкт-Петербург : Интермедия, 2017. — 335 с. — ISBN 978-5-4383-0136-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/66797.html. — Режим доступа: для авторизир. пользователей

1.3. Периодические издания:

1. Журнал «Вестник компьютерных и информационных технологий»
2. Журнал «Информатика и ее применения»
3. Журнал «Защита информации. Инсайд»
4. Вопросы кибербезопасности. Научный, периодический, информационно-методический журнал с базовой специализацией в области информационной безопасности.. URL: http://cyberrus.com/
5. Безопасность информационных технологий. Периодический рецензируемый научный журнал НИЯУ МИФИ. URL: http://bit.mephi.ru/

1.4. Интернет-ресурсы:

1. Электронный читальный зал “БИБЛИОТЕХ” : учебники авторов ТулГУ по всем дисциплинам.- Режим доступа: https://tsutula.bibliotech.ru/, по паролю.- Загл. С экрана.
2. ЭБС IPRBooks универсальная базовая коллекция изданий.-Режим доступа: http://www.iprbookshop.ru/, по паролю.- .- Загл. с экрана.
3. Научная Электронная Библиотека eLibrary – библиотека электронной периодики, режим доступа: http://elibrary.ru/ , по паролю.- Загл. с экрана.
4. НЭБ КиберЛенинка научная электронная библиотека открытого доступа, режим доступа http://cyberleninka.ru/ ,свободный.- Загл. с экрана.
5. Единое окно доступа к образовательным ресурсам: портал [Электронный ресурс]. - Режим доступа :http: //window.edu.ru. -Загл. с экрана.

ПРИЛОЖЕНИЕ А

Пример оформления титульного листа, представлен на рисунке А.1.

Минобрнауки России
ФГБОУ ВО «Тульский государственный университет»
Технический колледж Им. С.И. Мосина

РЕФЕРАТ

по дисциплине: «Основы информационной безопасности»

на тему «……..»

Выполнил:
студент группы 1-100205 И.И. Иванов

Проверил:
Преподаватель Э.А. Суворова

Тула 2021

Рисунок А.1 – Образец оформления титульного листа реферата

Министерство науки и высшего образования
Российской Федерации
ФГБОУ ВО «Тульский государственный университет»
Технический колледж им. С.И. Мосина

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРАКТИЧЕСКИМ ЗАНЯТИЯМ ПО ДИСЦИПЛИНЕ

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

для специальности

10.02.05 Обеспечение информационной безопасности автоматизированных систем

2024

УТВЕРЖДЕНА
цикловой комиссией информационных технологий
Протокол от «12» 09 2024 № 2
Председатель цикловой комиссии _____ И.В. Миляева

СОДЕРЖАНИЕ

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1 4

ОПРЕДЕЛЕНИЕ ОБЪЕКТОВ ЗАЩИТЫ НА ТИПОВОМ ОБЪЕКТЕ ИНФОРМАТИЗАЦИИ 4

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2 7

КЛАССИФИКАЦИЯ ЗАЩИЩАЕМОЙ ИНФОРМАЦИИ ПО ВИДАМ ТАЙНЫ И СТЕПЕНЯМ КОНФИДЕНЦИАЛЬНОСТИ 7

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3 10

ОПРЕДЕЛЕНИЕ УГРОЗ ОБЪЕКТА ИНФОРМАТИЗАЦИИ И ИХ КЛАССИФИКАЦИЯ 10

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4 17

РАБОТА В СПРАВОЧНО-ПРАВОВОЙ СИСТЕМЕ С НОРМАТИВНЫМИ И ПРАВОВЫМИ ДОКУМЕНТАМИ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ 17

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5 20

ВЫБОР МЕР ЗАЩИТЫ ИНФОРМАЦИИ ДЛЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА 20

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1

ОПРЕДЕЛЕНИЕ ОБЪЕКТОВ ЗАЩИТЫ НА ТИПОВОМ ОБЪЕКТЕ ИНФОРМАТИЗАЦИИ

Краткие теоретические сведения

Объект информатизации – совокупность информационных ресурсов, средств и систем информатизации, используемых в соответствии с заданной информационной технологией, и систем связи вместе с помещениями (транспортными средствами), в которых они установлены. Под объектами информатизации, аттестуемыми по требованиям безопасности информации, понимаются автоматизированные системы различного уровня и назначения, системы связи, отображения вместе с помещениями, в которых они установлены, предназначенные для обработки и передачи информации, подлежащей защите, а также сами помещения, предназначенные для ведения конфиденциальных переговоров.

Защищаемыми объектами информатизации в соответствии с СТР-К являются:

– средства и системы информатизации, в том числе информационно-вычислительные комплексы, сети и системы, средства и системы связи и передачи данных, технические средства приема, передачи и обработки информации, программные средства, используемые для обработки конфиденциальной информации;
– технические средства и системы, не обрабатывающие непосредственно конфиденциальную информацию, но размещенные в помещениях, где она обрабатывается (циркулирует);
– защищаемые помещения.

Объект защиты информации - информация или носитель информации, или информационный процесс, которые необходимо защищать в соответствии с поставленной целью защиты информации.

В соответствии с данным определением можно классифицировать объекты защиты в соответствии (рис.1).

Рис.1. Классификация объектов защиты

Контрольные вопросы

1. Дайте определение объекта информатизации?
2. Приведите пример объекта информатизации?
3. Дайте определение защищаемого объекта информатизации?
4. Приведите пример защищаемого объекта информатизации?
5. Дайте определение объекта защиты информации?
6. Приведите классификацию объектов защиты информации?

Задание для самостоятельной работы

1. Ответьте письменно на контрольные вопросы.
2. Для объекта информатизации, заданного преподавателем выполните описание, которое включает: общее описание объекта (местоположение, тип, категория объекта), разделение объекта на элементы, принадлежащие к определенному типу, определение источника информации, выделение среди элементов источника сигнала, выделение потенциальной среды распространения сигнала.
3. Для объекта информатизации, заданного преподавателем определите цель защиты информации и определите объекты защиты.
4. Составьте отчет по объектам защиты на заданном типовом объекте информатизации.

Примеры объектов информатизации

1. Выделенные помещения, в которых ведутся закрытые переговоры.
2. Выделенные помещения, в которых ведутся лекции, семинары по обсуждению информации закрытого характера, медицинские учреждения, банки.
3. Объекты, на которых проводится обработка информации закрытого характера, объединенных локальной вычислительной сетью.
4. Объекты, на которых проводится обработка информации закрытого характера, объединенных распределенной вычислительной сетью.

Варианты заданий

1	Отделение коммерческого банка
2	Поликлиника
3	Колледж
4	Офис страховой компании
5	Рекрутинговое агентство
6	Интернет-магазин
7	Центр оказания государственных услуг
8	Отделение полиции
9	Аудиторская компания
10	Дизайнерская фирма
11	Офис интернет-провайдера
12	Офис адвоката
13	Компания по разработке ПО для сторонних организаций
14	Агентство недвижимости
15	Туристическое агентство
16	Офис благотворительного фонда
17	Издательство
18	Консалтинговая фирма

19	Рекламное агентство
20	Отделение налоговой службы
21	Офис нотариуса
22	Бюро перевода (документов)
23	Научно проектное предприятие
25	Редакция газеты
26	Гостиница
28	Городской архив
29	Диспетчерская служба такси
30	Железнодорожная касса

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2

КЛАССИФИКАЦИЯ ЗАЩИЩАЕМОЙ ИНФОРМАЦИИ ПО ВИДАМ ТАЙНЫ И СТЕПЕНЯМ КОНФИДЕНЦИАЛЬНОСТИ

Краткие теоретические сведения

Существуют различные признаки, по которым классифицируется информация. С точки зрения защиты информации наиболее интересной является классификация по категории доступа.

Документированная информация с ограниченным доступом по условиям ее правового режима подразделяется на информацию, отнесенную к государственной тайне, и конфиденциальную. Отнесение информации к государственной тайне осуществляется в соответствии с Законом Российской Федерации "О государственной тайне". Существует три степени секретности такой информации:

Особой важности

Совершенно Секретно

Секретно

Информация в зависимости от порядка ее предоставления или распространения подразделяется на:

информацию, свободно распространяемую;

информацию, предоставляемую по соглашению лиц, участвующих в соответствующих отношениях;

информацию, которая в соответствии с федеральными законами подлежит предоставлению или распространению;

информацию, распространение которой в Российской Федерации ограничивается или запрещается.

К общедоступной информации относятся общеизвестные сведения и иная информация, доступ к которой не ограничен. Обладатель информации, ставшей общедоступной по его решению, вправе требовать от лиц, распространяющих такую информацию, указывать себя в качестве источника такой информации. То есть, если журнал публикуется в Интернете и находится в открытом доступе по решению его создателей, они вправе требовать ссылки на источник в случае использования информации из него где-либо еще. К общедоступной также относится информация, доступ к которой нельзя ограничить. Примером может служить информация о состоянии окружающей среды, о деятельности органов государственной власти и органов местного самоуправления, документы, накапливаемые в открытых фондах библиотек и архивов. Так же в эту категорию можно отнести нормативные правовые акты, затрагивающие права, свободы и обязанности человека и гражданина, правовое положение организаций и полномочия государственных органов, органов местного самоуправления.

К видам конфиденциальной информации можно отнести следующее:

Персональные данные - сведения о фактах, событиях и обстоятельствах частой жизни гражданина, позволяющие идентифицировать его личность, за исключением сведений, подлежащих распространению в средствах массовой информации в установленном федеральными законами случаях;

Тайна следствия и судопроизводства - сведения, составляющие тайну следствия и судопроизводства, а также сведения о защищаемых лицах и мерах государственной защиты, осуществляемой в соответствии с ФЗ;

Служебная тайна - служебные сведения, доступ к которым ограничен органами государственной власти в соответствии с Гражданским кодексом Российской Федерации и федеральными законами;

Профессиональная тайна - сведения, связанные с профессиональной деятельностью, доступ к которым ограничен в соответствии с Конституцией Российской Федерации и феде-

ральными законами (врачебная, нотариальная, адвокатская тайна, тайна переписки, телефонных переговоров, почтовых отправлений, телеграфных и иных сообщений и т.д.);

Коммерческая тайна - сведения, связанные с коммерческой деятельностью, доступ к которым ограничен в соответствии с Гражданским кодексом Российской Федерации и федеральными законами;

Сведения о сущности изобретения - сведения о сущности изобретения, полезной модели или промышленного образца до официальной публикации информации о них.

На настоящее время защита информации на государственном уровне сконцентрирована вокруг вопросов обеспечения конфиденциальности информации. Тем не менее, современное развитие информатизации требует уделять больше внимания обеспечению таких свойств информации, как целостность и доступность. На практике же есть достаточно много форм деятельности, где доля конфиденциальной информации сравнительно мала. Например, для открытой информации приоритетными направлениями будет обеспечение целостности и доступности информации. Для платежных документов, отправляемых через системы дистанционного банковского обслуживания, наибольшую важность представляет целостность информации, так как если документ будет подделан, владелец может получить колоссальный финансовый ущерб. Следовательно, традиционный подход к защите информации с точки зрения обеспечения только конфиденциальности, требует существенной модернизации.

Контрольные вопросы

1. Дайте классификацию информации по категории доступа.
2. Какие степени секретности определены для информации, отнесенной к государственной тайне?
3. Как подразделяется информация в зависимости от порядка ее предоставления или распространения?
4. Что относится к общедоступной информации?
5. Что относится к конфиденциальной информации?
6. Дайте определения понятиям персональные данные, тайна следствия и судопроизводства, служебная тайна, профессиональная тайна, коммерческая тайна, сведения о сущности изобретения?
7. Какие сведения, в соответствии с законодательством, не могут быть отнесены к информации с ограниченным доступом?
8. Какая система обозначения сведений, составляющих государственную тайну, принята в РФ?
9. Назовите группу видов ущерба, возникающего при утечке сведений, составляющих государственную тайну.
10. Что в соответствии с законодательством РФ представляет собой засекречивание информации.
11. В чем заключается разница между понятием «конфиденциальная информация» и «тайна»?
12. В чем состоит сложность служебной тайны с точки зрения определения ее правового режима?

Задание для самостоятельной работы

1. Ответьте письменно на контрольные вопросы.
2. Изучите основные положения следующих документов: Об информации, информационных технологиях и о защите информации: Федеральный закон от 27.07.2006 г. № 149-ФЗ, О связи: Федеральный закон от 07.07.2003 г. № 126-ФЗ. Об электронной цифровой подписи: Федеральный закон от 10.01.2002 г. № 1-ФЗ., О коммерческой тайне: Федеральный закон от 29.07.2004 г. № 98-ФЗ., О персональных данных: Федеральный закон от 27.07.2006 г. № 152.-ФЗ.
3. Выполните практическое задание. Ваша группа является частью команды организации-

разработчика, которая участвует в проекте создания комплексной системы защиты информационной системы (ПЗ №1). Ваша задача заключается в сборе информации о объекте автоматизации, определении состава защищаемой информации, классификации этой информации, в том числе по видам тайн и степеням конфиденциальности. По итогам работы ваша группа составляет сведения об основные информационных пакетах, классифицируемых по вышеуказанному признаку. По итогам вашей работы будут формулироваться дальнейшие требования к проектируемой системе, которые впоследствии потребуется внести в техническое задание на создание системы.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3

ОПРЕДЕЛЕНИЕ УГРОЗ ОБЪЕКТА ИНФОРМАТИЗАЦИИ И ИХ КЛАССИФИКАЦИЯ

Краткие теоретические сведения

Для того чтобы построить эффективную систему информационной безопасности, необходимо в первую очередь определить потенциальные и реальные угрозы технического проникновения на защищаемый объект, возможные каналы для несанкционированного доступа и утечки защищаемой информации.

Данная работа базируется на знании природы возникновения технических каналов утечки информации и методов ведения технической разведки. Правильное определение потенциальных угроз на предпроектном этапе построения системы защиты позволит в дальнейшем выбрать наиболее оптимальные меры и средства защиты.

При выявлении технических каналов утечки информации необходимо рассматривать всю совокупность элементов защиты, включающую основное оборудование технических средств обработки информации, соединительные линии, распределительные и коммутационные устройства, системы электропитания, системы вентиляции и т.п.

Наряду с основными техническими средствами, непосредственно связанными с обработкой и передачей конфиденциальной информации, необходимо учитывать и вспомогательные технические средства и системы (ВТСС), такие, как технические средства открытой телефонной, факсимильной, громкоговорящей связи, системы охранной и пожарной сигнализации, электрификации, радиофикации, электробытовые приборы и др. Наибольшее внимание следует уделить вспомогательным средствам, имеющие линии, выходящие за пределы контролируемой зоны.

В качестве каналов утечки больше внимания следует уделить вспомогательным средствам, имеющим линии, выходящие за пределы контролируемой зоны, а также посторонним проводам и кабелям, проходящим через помещения, где установлены основные и вспомогательные технические средства, металлические трубы систем отопления, водоснабжения и другие токопроводящие металлоконструкции.

При оценке защищенности помещений от утечки речевой информации необходимо учитывать возможность ее прослушивания, как из соседних помещений, так и с улицы. Следует проводить оценку возможности ведения разведки с использованием лазерных микрофонов. Интерес могут вызывать каналы утечки за счет вибраций, возникающих под давлением акустических волн, в твердых телах (ограждениях, трубах и т.п.).

Оценка защищенности объекта включает в себя анализ режима работы и охраны объекта, с целью моделирования действий по скрытному проникновению на них (неконтролируемому пребыванию) посторонних лиц. Режим работы специалистов сторонних организаций, приобретение, установка и ремонт мебели, оргтехники и т.п. Т.е. всю совокупность условий, позволяющих внедрить на объект специальные закладные устройства перехвата информации (микропередатчики, возможность установки миниатюрных микрофонов с подключением к внешним линиям и т.д.). А также определение наиболее эффективных, для использования на разных уровнях проникновения, средств технической разведки.

Большое, а иногда решающее, значение при оценке угрозы может иметь знание наиболее вероятного противника, его финансовых и оперативных возможностей, знание личностных качеств постоянного персонала, временных работников и другая дополнительная информация.

Согласно ***ГОСТ Р 50922-96 "Защита информации. Основные термины и определения"*** несанкционированный доступ (НСД) к информации – деятельность, направленная на предотвращение получения защищаемой информации заинтересованным субъектом с нарушением установленных правовыми документами или собственником, владельцем информации прав или правил доступа к защищаемой информации.

Заинтересованным лицом в данном случае может выступать:

- государство;
- юридическое лицо;
- группа физических лиц или одно физическое лицо;
- общественная организация.

Проблеме защиты информации от НСД посвящен ряд руководящих документов ФСТЭК (соответственно, бывш.Гостехкомиссии). Определение НСД там более узкое, чем в ГОСТе.

Несанкционированный доступ к информации - доступ к информации, нарушающий правила разграничения доступа с использованием штатных средств, предоставляемых средствами вычислительной техники (СВТ) или автоматизированными системами (АС)

Под штатными средствами понимается совокупность программного, микропрограммного и технического обеспечения средств вычислительной техники или автоматизированных систем. Несмотря на это, действие внедренной программной закладки ("червя" и т.п.), результатом которого стало попадание защищаемой информации к злоумышленнику, также можно рассматривать как факт НСД.

К основным угрозам НСД можно отнести следующее:

- угрозы проникновения в операционную среду компьютера с использованием штатного программного обеспечения (средств операционной системы или прикладных программ общего применения);
- угрозы создания нештатных режимов работы программных (программно-аппаратных) средств за счет преднамеренных изменений служебных данных, игнорирования предусмотренных в штатных условиях ограничений на состав и характеристики обрабатываемой информации, искажения (модификации) самих данных и т.п.;
- угрозы внедрения вредоносных программ (программно-математического воздействия).

Кроме этого, возможны комбинированные угрозы, представляющие собой сочетание указанных угроз. Например, за счет внедрения вредоносных программ могут создаваться условия для НСД в операционную среду компьютера.

В общем случае, ***комплекс программно - технических средств и организационных мер*** по защите информации от НСД реализуется в рамках системы защиты информации от НСД (СЗИ НСД), условно состоящей из следующих четырех подсистем:

- управления доступом;
- регистрации и учета;
- криптографической;
- обеспечения целостности.

Источником угрозы НСД может быть нарушитель, носитель с вредоносной программой или аппаратная закладка. Модель нарушителя – абстрактное (формализованное или неформализованное) описание нарушителя.

При разработке модели нарушителя определяются следующие предположения:

- категория лиц, к которой относится нарушитель;
- мотивы нарушителя;
- цели нарушителя;
- квалификация и техническая оснащенность нарушителя;
- возможные действия нарушителя;
- время действия (постоянно, в определенные интервалы).

Выделяют две категории нарушителей по отношению к АС – внутренние и внешние. Внешними нарушителями могут быть:

- разведывательные службы государств;
- криминальные структуры;
- конкуренты;
- недобросовестные партнеры;

– внешние субъекты (физические лица).

Злоумышленник – это нарушитель, намеренно идущий на нарушение.

Внешний нарушитель может осуществлять НСД следующими способами:

- через автоматизированные рабочие места, подключенные к сетям общего пользования или сетям международного обмена (например, Интернет);
- с помощью программного воздействия на защищаемую информацию (программные закладки, вирусы и др.);
- через элементы автоматизированной системы, которые побывали за пределами контролируемой зоны (например, съемные носители информации);

Внутренними нарушителями могут быть следующие категории лиц:

- пользователи АС;
- персонал АС;
- сотрудники службы безопасности;
- руководители различных уровней.

По уровню знаний нарушителей можно классифицировать следующим образом:

- знает функциональные особенности АС, основные закономерности формирования в ней массивов данных и потоков запросов к ним, умеет пользоваться штатными средствами;
- обладает высоким уровнем знаний и опытом работы с техническими средствами системы и их обслуживания;
- обладает высоким уровнем знаний в области программирования и вычислительной техники, проектирования и эксплуатации автоматизированных информационных систем;
- знает структуру, функции и механизм действия средств защиты, их сильные и слабые стороны.

По уровню возможностей (используемым методам и средствам):

- применяющий только агентурные методы получения сведений;
- применяющий пассивные средства (технические средства перехвата без модификации компонентов системы);
- использующий только штатные средства и недостатки систем защиты для ее преодоления (несанкционированные действия с использованием разрешенных средств), а также компактные магнитные носители информации, которые могут быть скрытно пронесены через посты охраны;
- применяющий методы и средства активного воздействия (модификация и подключение дополнительных технических средств, подключение к каналам передачи данных, внедрение программных закладок и использование специальных инструментальных и технологических программ).

По времени действия:

- в процессе функционирования АС (во время работы компонентов системы);
- в период неактивности компонентов системы (в нерабочее время, во время плановых перерывов в ее работе, перерывов для обслуживания и ремонта и т.п.);
- как в процессе функционирования АС, так и в период неактивности компонентов системы.

По месту действия:

- без доступа на контролируемую территорию организации;
- с контролируемой территории без доступа в здания и сооружения;
- внутри помещений, но без доступа к техническим средствам АС;
- с рабочих мест конечных пользователей (операторов) АС;
- с доступом в зону данных (серверов баз данных, архивов и т.п.);

– с доступом в зону управления средствами обеспечения безопасности АС.

Могут учитываться следующие ограничения и предположения о характере действий возможных нарушителей:

– работа по подбору кадров и специальные мероприятия затрудняют возможность создания коалиций нарушителей, т.е. объединения (сговора) и целенаправленных действий по преодолению подсистемы защиты двух и более нарушителей;
– нарушитель, планируя попытки НСД, скрывает свои несанкционированные действия от других сотрудников.

Модель нарушителей может иметь разную степень детализации.

Содержательная модель нарушителей отражает систему принятых руководством объекта защиты взглядов на контингент потенциальных нарушителей, причины и мотивацию их действий, преследуемые цели и общий характер действий в процессе подготовки и совершения акций воздействия.

Сценарии воздействия нарушителей определяют классифицированные типы совершаемых нарушителями акций с конкретизацией алгоритмов и этапов, а также способов действия на каждом этапе.

Математическая модель воздействия нарушителей представляет собой формализованное описание сценариев в виде логико-алгоритмической последовательности действий нарушителей, количественных значений, параметрически характеризующих результаты действий, и функциональных (аналитических, численных или алгоритмических) зависимостей, описывающих протекающие процессы взаимодействия нарушителей с элементами объекта и системы охраны. Именно этот вид модели используется для количественных оценок уязвимости объекта и эффективности охраны.

Контрольные вопросы

1. Дайте определение понятию несанкционированный доступ?
2. Что можно отнести к основным угрозам НСД?
3. Как внешний нарушитель может осуществлять НСД?
4. Перечислите категории лиц, которые могут быть внутренними нарушителями?
5. Дайте классификацию нарушителей по уровню знаний?
6. Дайте классификацию нарушителей по уровню возможностей?
7. Дайте классификацию нарушителей по времени действия?
8. Дайте классификацию нарушителей по месту действия?

Задание для самостоятельной работы

1. Ответьте письменно на контрольные вопросы.
2. Рассмотрим угрозы объекта информатизации.

На рисунке 2 приведена примерная схема защищаемого объекта.

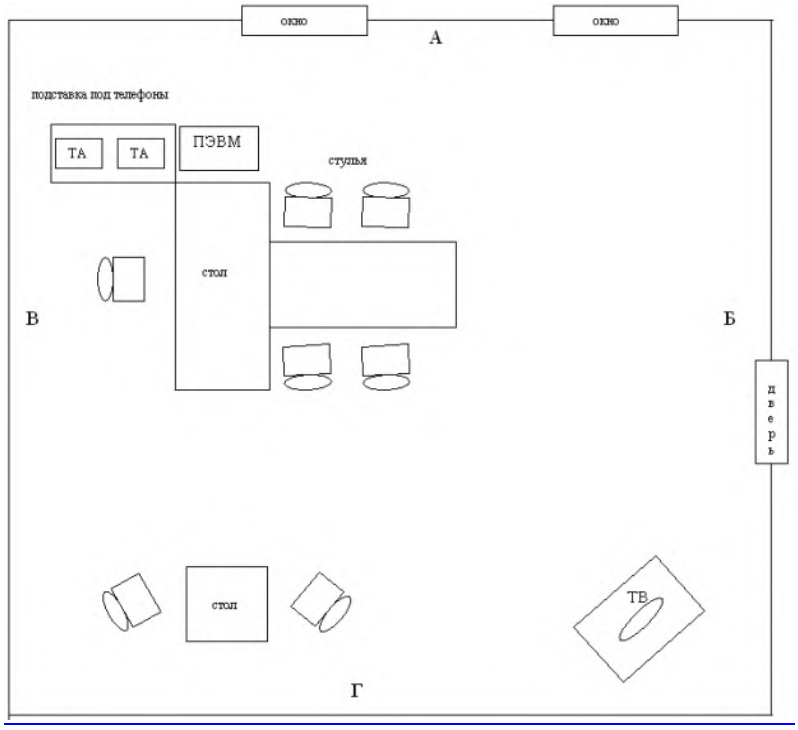


Рис.2. Схема объекта

1. Защищаемое помещение расположено на четвертом этаже 7-этажного здания. Все здание принадлежит одной организации:

Сверху расположены служебные помещения.

Снизу расположены технические помещения (туалет, электрощитовая).

Со стороны стены Б расположена приемная.

Со стороны стены Г расположен общий коридор.

Стороны А и В выходят на улицы с интенсивным пешеходным и транспортным движением.

2. Окна помещения оборудованы шторами, смотрят на жилой дом, расположенный на расстоянии 30 метров.

3. Из мебели в помещении установлены рабочий и журнальный столы, стулья, подставки под: телефоны, ПЭВМ и телевизор.

4. Из основных технических средств в помещении установлен телефон внутренней конфиденциальной связи, ПЭВМ, включенная в локальную сеть.

5. Из вспомогательных технических средств, в помещении установлен телефон ГТС, телевизор, радиотрансляционный приемник. Помещение оборудовано системой пожарной и охранной сигнализации, линии которых выходят на пульт дежурного охранника. Помещение электрифицировано (освещение, питание оборудования).

6. Помещение оборудовано системой вытяжной вентиляции, короб которой проложен вдоль коридора и поднимается на крышу здания. Радиаторы отопления установлены вдоль стены А. Трубы отопления спускаются в подвал.

7. Режим работы учреждения предусматривает свободное передвижение сотрудников и посетителей в рабочее время. В ночное время помещение закрывается на ключ, сдается под охрану дежурному. Системы связи обслуживаются штатным сотрудником. Системы жизнеобеспечения (отопление, канализация) обслуживаются по заявке приходящим сотрудником.

8. Доступ штатных сотрудников к служебной информации не разграничен.

Рассмотрим угрозы объекта информатизации

1. Просмотр помещения со стороны улицы, ввиду того, что помещение находится на 4 этаже, невозможен. Так как возможен просмотр помещения извне, со стороны жилого дома с помощью оптических приборов, существует потенциальный канал утечки информации.

Если организационными мероприятиями (соответствующим инструктажем ответственных лиц) введено обязательное зашторивание окон во время проведением совещаний, работы с документами и т.п., то реального визуально оптического канала утечки информации нет.

2. В качестве дополнительных мер можно ввести периодический контроль за соблюдением сотрудниками правила зашторивания, а также поставить тонированные или рифленые стекла.

3. Так как возможно прослушивание помещения, со стороны улицы и жилого дома, через открытые окна и форточки с помощью направленных микрофонов, существует потенциальный канал утечки акустический информации.

4. Однако, если организационными мероприятиями введено обязательное закрытие окон и форточек во время проведения совещаний, реального акустического канала утечки информации нет.

В качестве дополнительной меры можно установить кондиционер или приобрести генератор белого шума и включать его во время проведения совещаний.

5. Так как возможен съем информации о ведущихся в помещении разговорах с оконных стекол, за счет их вибрации, при использовании лазерного микрофона, при расположении поста перехвата в жилом доме, существует еще один потенциальный канал утечки акустической информации.

6. В данном случае с помощью одних организационных мероприятий устранить канал утечки не представляется возможным. Однако реальное существование канала утечки может быть констатировано лишь после проведения инструментальных измерений.

По результатам инструментальной проверки будет определяться необходимость проведение защитного мероприятия, например установка рифленых стекол или зашумление стекол и пространства между ними.

Меры защиты:

Установка стекол с рифленой поверхностью и кондиционера. Решение представляется оптимальным, т.к. акустический и визуально оптический каналы устраняются при минимальных финансовых затратах. Также, в дальнейшем, обеспечивается удобство эксплуатации объекта и исключается негативный человеческий фактор.

Самостоятельная часть работы

1. Выявить оставшиеся, потенциально возможные каналы утечки информации (с учетом исходных данных, используя, при необходимости оговорки);

2. Смоделировать возможные действия технических разведок, определить реальные каналы утечки информации;

При оценке вероятности использования технической разведкой потенциальных каналов утечки информации следует принимать во внимание окружающую обстановку, а именно: скрытное размещение поста перехвата (для прослушивания и просмотра помещения) на улице с интенсивным движением затруднительно, т.к. подозрительные лица, транспортные средства и т.п. привлекают к себе внимание, легко визуально обнаруживаются; скрытное размещение поста перехвата (для прослушивания и просмотра помещения, установки лазерного микрофона) в жилом здании, если, например, арендовать квартиру с окнами расположенными напротив окон защищаемого помещения, вполне реализуемо.

Необходимо, если имеется такая возможность, проверить благонадежность (лояльность) жильцов в квартирах, потенциально пригодных для организации поста перехвата (сдаются ли квартиры, проживают ли в квартирах потенциальные конкуренты, имеются ли лица бывшие в конфликте с законом и т.п.). Возможности организации постов перехвата на технических этажах и т.п.

В случае получения в ходе проверки положительных данных можно заключить, что защитные мероприятия не требуются вообще. С точки зрения защиты от случайных утечек, например прослушивания, можно заключить, что улица с интенсивным автомобильным и пешеходным движением создает достаточно сильную акустическую помеху, за которой разговоры случайными прохожими различаться не будут. При необходимости в этом можно убедиться экспериментально.

В случае получения в ходе проверки отрицательных или неоднозначных данных оптимальным остается вариант указанный в заключение первого этапа.

3. Доказать целесообразность и предложить проведение тех или иных защитных мероприятий.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4

РАБОТА В СПРАВОЧНО-ПРАВОВОЙ СИСТЕМЕ С НОРМАТИВНЫМИ И ПРАВОВЫМИ ДОКУМЕНТАМИ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Краткие теоретические сведения

Работы по созданию общегосударственной автоматизированной системы правовой информации (ОГАСПИ) ведутся давно, с 70-х годов прошлого века. Эти работы являлись частью работ по созданию общегосударственной автоматизированной системы (ОГАС). Разрабатываются также государственные и негосударственные справочно-информационные системы. Рассмотрим некоторые основные такие системы.

Система "Консультант Плюс". Сеть этой системы объединяет несколько сотен региональных информационных центров, производящих поставку справочных правовых систем, сервисное обслуживание и передачу информации пользователям. Контент "Консультант Плюс" – самый объемный из всех аналогичных систем.

Система позволяет обновлять информационный банк пользователя новыми документами и изменениями без его полной замены. Региональные информационные центры централизованно и регулярно получают в электронном виде текущие дополнения и изменения из удаленной базы Координационного Центра Сети. Пользователь получает их по электронным или иным коммуникациям. В результате выполнения кратковременной процедуры обновления, информационный банк пользователя становится идентичным эталонному информационному банку.

Первой негосударственной справочно-информационной правовой системой является ***система "Гарант"***. Нормативные акты поступают в базу системы "Гарант" непосредственно из Администрации Президента, Правительства, Министерства финансов, Центрального банка, Федеральной таможенной службы, Государственной налоговой службы и Высшего Арбитражного Суда РФ и других официальных органов.

Система "Гарант" позволяет делать выборку информации и основных документов по всем разделам банковского, таможенного, жилищного, земельного законодательства, уголовному и административному праву и др.

Правовая информационно-справочная система "Кодекс", которая имеет в базе данных около нескольких десятков тысяч нормативных актов РФ, а также инструкций, приложений и комментариев, практика применения, консультация, ответы на типовые вопросы и др.

Одной и ранних систем является ***информационно-справочная система "Эталон"*** Министерства юстиции РФ. "Эталон" имеет полнотекстовую обновляемую регулярно базу данных с действующими официальными редакциями текстов нормативных актов, а также информацией о деятельности судебных органов и Министерства юстиции РФ и др.

Правовую информатизацию России регламентирует Указ Президента Российской Федерации от 28 июня 1993 г. N 966 "О Концепции правовой информатизации России", а также Федеральная программа "Электронная Россия".

При создании таких сложных информационных систем приходится разрабатывать очень сложные программные комплексы, состоящие из большого числа программ различной функциональной ориентации. Немаловажное значение имеет также и профессиональная подготовка пользователей таких систем.

Контрольные вопросы

1. Какова роль и значение информационно-правовых систем в обществе и праве?
2. Каковы типы информационно-правовых систем и их примеры?
3. Каковы основные возможности и назначение системы "Консультант Плюс"?
4. Каковы основные возможности и назначение системы "Кодекс"?
5. Каковы основные возможности и назначение системы "Эталон"?
6. Каковы основные возможности и назначение системы "Гарант"?

Задание для самостоятельной работы

1. Ответьте письменно на контрольные вопросы
2. Опишите возможные варианты и деловые ситуации использования информационной системы "Консультант Плюс".
3. Опишите возможные варианты и деловые ситуации использования информационной системы "Кодекс".
4. Опишите возможные варианты и деловые ситуации использования информационной системы типа "Гарант".
5. Откройте главную страницу "Консультант Плюс"

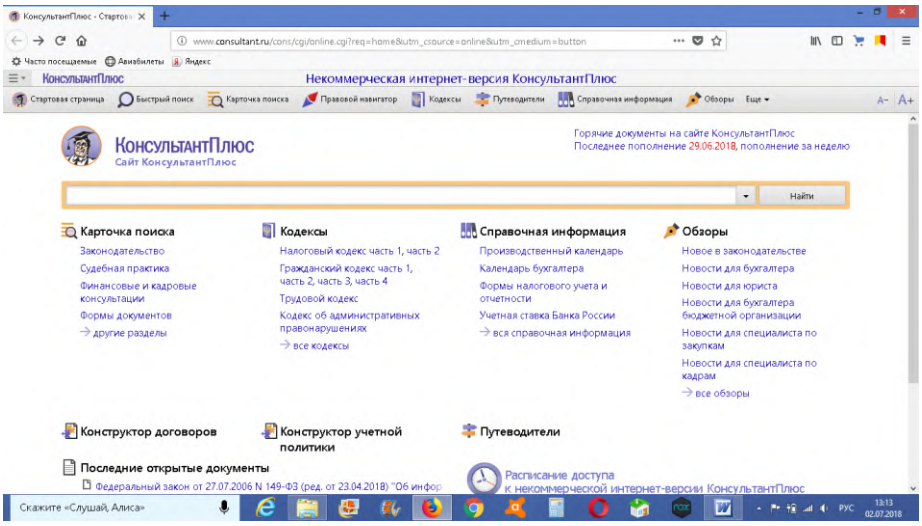


Рис.1.

6. Откройте ссылку "Законодательство".
7. Найдите Федеральный Закон "Об информации, информационных технологиях и о защите информации"

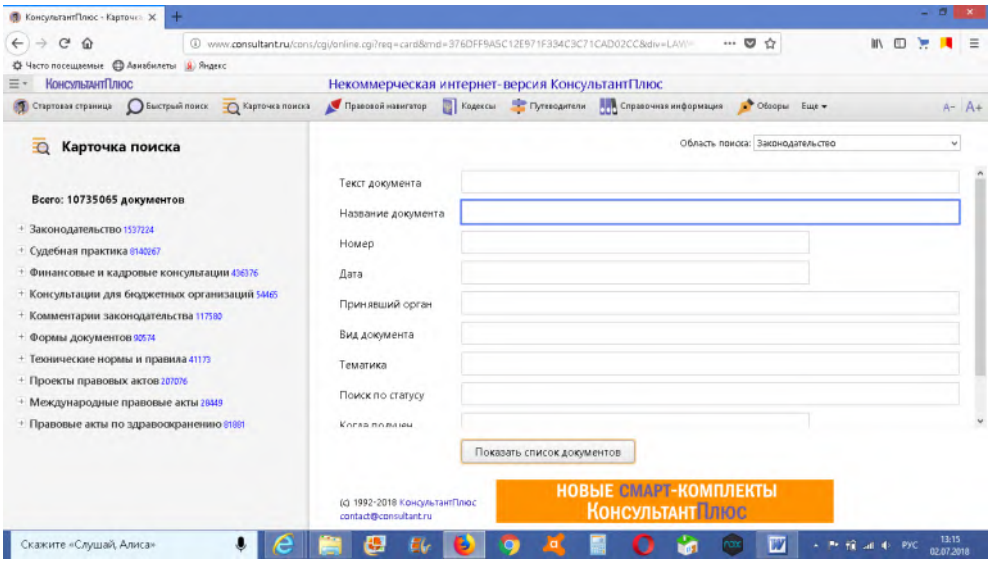


Рис.2.

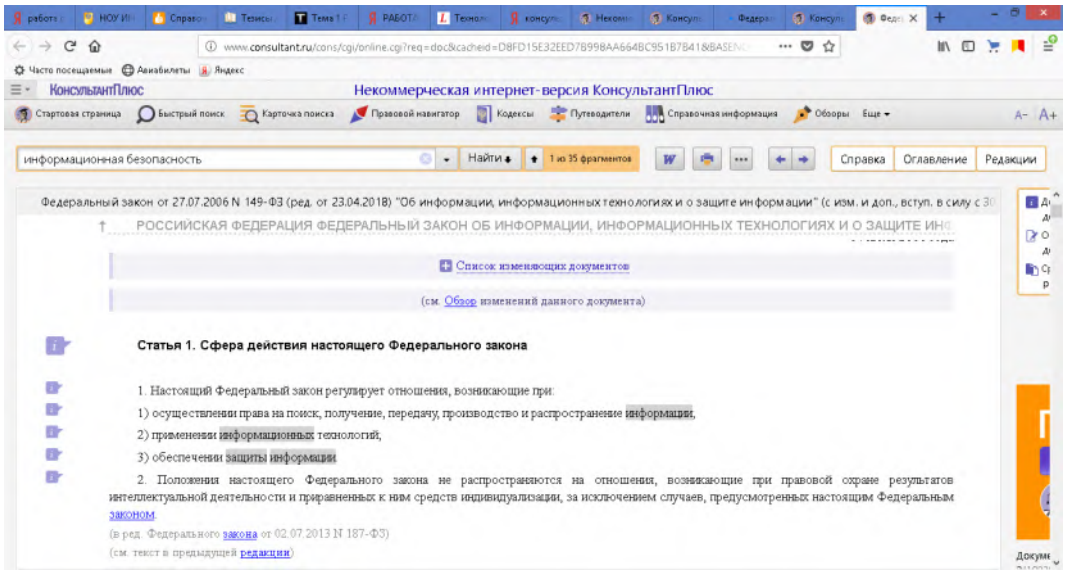


Рис.3.

8. Изучите содержание закона. Выпишите его основные положения.
9. Какие еще законы и нормативные акты Вы можете найти с помощью такой системы. Выпишите названия трех из них.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5

ВЫБОР МЕР ЗАЩИТЫ ИНФОРМАЦИИ ДЛЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА

Краткие теоретические сведения

Подавляющее большинство информации в современном мире обрабатывается в автоматизированных системах (АС). Следовательно, АС является "наиболее популярным" объектом защиты. Все действующие АС, обрабатывающие конфиденциальную информацию, соответственно, нуждающиеся в защите от НСД, классифицируются в соответствии с Руководящим документом ФСТЭК (соответственно, бывш.Гостехкомиссии) «Автоматизированные системы. Защита от НСД к информации. Классификация АС и требования по защите информации».

В соответствии с документом, классификация АС включает следующие этапы:

1. Разработка и анализ исходных данных.
2. Выявление основных признаков АС, необходимых для классификации.
3. Сравнение выявленных признаков АС с классифицируемыми.
4. Присвоение АС соответствующего класса защиты информации от НСД.

Исходными данными для классификации АС являются:

1. Перечень защищаемых информационных ресурсов АС и их уровень конфиденциальности.
2. Перечень лиц, имеющих доступ к штатным средствам АС, с указанием их уровня полномочий.
3. Матрица доступа или полномочий субъектов доступа по отношению к защищаемым информационным ресурсам АС.
4. Режим обработки данных в АС.

Выбор класса АС производится заказчиком и разработчиком с привлечением специалистов по защите информации.

К числу определяющих признаков классификации АС относится следующее:

- наличие в АС информации различного уровня конфиденциальности;
- уровень полномочий субъектов доступа АС на доступ к конфиденциальной информации;
- режим обработки данных в АС - коллективный или индивидуальный.

Устанавливаются 9 классов защищённости АС от НСД к информации, каждый класс характеризуется определённой минимальной совокупностью требований по защите. Классы подразделяются на 3 группы:

III группа – классы 3Б и 3А.

Классы соответствуют автоматизированным системам, в которых работает один пользователь, допущенный ко всей информации в АС, размещённой на носителях одного уровня конфиденциальности.

II группа – классы 2Б и 2А.

Классы данной группы соответствуют автоматизированным системам, в которых пользователи имеют одинаковые права доступа(полномочия) ко всей информации в АС, обрабатываемой и (или) хранимой на носителях различного уровня конфиденциальности.

I группа – классы 1Д, 1Г, 1В, 1Б и 1А.

Включает многопользовательские АС, в которых одновременно обрабатывается или хранится информация разных уровней конфиденциальности. Не все пользователи имеют доступ ко всей информации в АС.

Выделяют 4 подсистемы для обеспечения защиты от НСД:

- управления доступом;
- регистрации и учета;

– криптографическая;
– обеспечения целостности.

Более детально требования в зависимости от класса защищенности описаны в Руководящем документе ФСТЭК (соответственно, бывш.Гостехкомиссии) «Автоматизированные системы. Защита от НСД к информации. Классификация АС и требования по защите информации». Пересмотр класса защищенности АС производится в обязательном порядке, если произошло изменение хотя бы одного из критериев, на основании которых он был установлен.

Если АС, классифицированная ранее, включается в состав вычислительной сети или системы и соединяется с другими техническими средствами линиями связи различной физической природы, образуемая при этом АС более высокого уровня классифицируется в целом, а в отношении АС нижнего уровня классификация не производится.

Если объединяются АС различных классов защищенности, то интегрированная АС должна классифицироваться по высшему классу защищенности входящих в нее АС, за исключением случаев их объединения посредством межсетевого экрана, когда каждая из объединяющихся АС может сохранять свой класс защищенности.

Классификация СВТ

Гостехкомиссия России, ныне ставшая ФСТЭК, разделила понятия АС и средств вычислительной техники (СВТ). Выделение СВТ в отдельную категорию обусловлено тем, что СВТ представляют собой компоненты при построении АС, то есть если СВТ не интегрированы в АС и не решают какой-то прикладной задачи, они не содержат пользовательской информации. Помимо этого, АС является более широким понятием, включающим в себя персонал, помещения, технологии обработки информации, полномочия пользователей системы.

Пример: ПЭВМ с установленной на ней операционной системой формально является СВТ. Но если ее поставить в конкретную комнату, закрепить за ней пользователя, выполняющего какую-то работу, - АС. Следует отметить, что средства защиты информации также относятся к СВТ.

СВТ - совокупность программных и технических элементов систем обработки данных, способных функционировать самостоятельно или в составе других систем.

СВТ классифицируются в соответствии с Руководящим документом ФСТЭК (соответственно, бывш.Гостехкомиссии) «СВТ. Защита от НСД к информации. Показатели защищенности от НСД».

Руководящий документ устанавливает классификацию средств вычислительной техники по уровню защищённости от НСД к информации на базе перечня показателей защищённости и совокупности описывающих их требований.

Устанавливаются 7 классов защищённости СВТ от НСД к информации, при этом самый низкий класс – седьмой, самый высокий – первый. Каждый класс разбит на 4 группы:

I. 7 класс – СВТ, которые были представлены к оценке, однако не удовлетворяют требованиям более высоких классов.

II. 6 и 5 классы – дискреционная защита.

III. 4, 3 и 2 классы – мандатная защита.

IV. 1 класс – верифицированная защита.

Требования ужесточаются с уменьшением номера класса.

Классы являются иерархически упорядоченными: каждый последующий класс содержит требования всех предыдущих. Выбор класса защищенности СВТ для автоматизированных систем, создаваемых на базе защищенных СВТ, зависит от грифа секретности обрабатываемой в АС информации, условий эксплуатации и расположения объектов системы.

В общем случае требования предъявляются к следующим показателям защищённости:

– Дискреционный принцип контроля доступа.
– Мандатный принцип контроля доступа.
– Очистка памяти.
– Изоляция модулей.
– Маркировка документов.

Контрольные вопросы

1. В соответствии с Руководящим документом ФСТЭК «Автоматизированные системы. Защита от НСД к информации. Классификация АС и требования по защите информации» дайте определения следующим понятиям.

- классификация АС
- исходные данные для классификации АС
- классы защищённости АС от НСД к информации
- подсистемы для обеспечения защиты от НСД

2. В соответствии с руководящим документом ФСТЭК «СВТ. Защита от НСД к информации. Показатели защищенности от НСД» дайте классификацию средств вычислительной техники по уровню защищённости от НСД к информации на базе перечня показателей защищённости и совокупности описывающих их требований

Задание для самостоятельной работы

1. Письменно ответить на контрольные вопросы.
2. Для АС (ПЗ № 1) определите классы защищенности.
3. Выполните выбор мер защиты.

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тульский государственный университет»
Технический колледж имени С.И. Мосина

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению практических работ
по дисциплине

«Организационно-правовое обеспечение информационной безопасности»

по специальности 10.02.05 Обеспечение информационной безопасности автоматизированных систем

Тула 2024

УТВЕРЖДЕНЫ

Цикловой комиссий информационных технологий
Протокол от «12» 09 2024 г. № 2

Председатель цикловой комиссии И.В. Миляева

Составитель: Суворова Эрминэ Айказовна, преподаватель
Технического колледжа им. С.И. Мосина ТулГУ

Содержание

Практическое занятие № 1 4
Работа с нормативными документами 4
Практическое занятие № 2 7
Защита информации, содержащейся в информационных системах общего пользования 7
Практическое занятие № 3 11
Разработка базового блока документов для обеспечения информационной безопасности ИСПДн 11
Практическое занятие № 4 14
Подготовка документов к получению лицензии 14
Практическое занятие № 5 16
Подготовка документов к сертификации 16
Практическое занятие № 6 19
Подготовка документов к аттестации объектов информатизации службы информационной безопасности 19
Практическое занятие № 7 26
Составление трудового договора сотрудника службы информационной безопасности 26

Практическое занятие № 1

Работа с нормативными документами

Цель работы: Получить навыки по работе с законодательством РФ в области обеспечения информационной безопасности.

Краткие теоретические сведения.

При разработке законодательных и других правовых и нормативных документов, а также при организации защиты информации важно правильно ориентироваться во всем блоке действующей законодательной базы в этой области. Базовым законом в области защиты информации является Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 29.07.2017) "Об информации, информационных технологиях и о защите информации"

(далее для краткости - "Закон об информации"), который регламентирует отношения, возникающие при формировании и использовании информационных ресурсов Российской Федерации на основе сбора, накопления, хранения, распространения и предоставления потребителям документированной информации, а также при создании и использовании информационных технологий, при защите информации и прав субъектов, участвующих в информационных процессах и информатизации.

В соответствие с этим Законом должны быть приведены ранее изданные Президентом Российской Федерации и Правительством Российской Федерации правовые акты, а также все законодательство России (статья 25 Закона).

Закон гласит:

- информационные ресурсы делятся на ГОСУДАРСТВЕННЫЕ и НЕГОСУДАРСТВЕННЫЕ (статья 6, часть 1);
- государственные информационные ресурсы Российской Федерации формируются в соответствии со сферами ведения как: федеральные информационные ресурсы; информационные ресурсы, находящиеся в совместном ведении Российской Федерации и субъектов Российской Федерации; информационные ресурсы субъектов Российской Федерации (статья 7, часть 1);
- государственные информационные ресурсы являются ОТКРЫТЫМИ и ОБЩЕДОСТУПНЫМИ. Исключение составляет документированная информация, отнесенная законом к категории ОГРАНИЧЕННОГО ДОСТУПА (статья 10, часть 1);
- документированная информация с ограниченного доступа по условиям ее правового режима подразделяется на информацию, отнесенную к ГОСУДАРСТВЕННОЙ ТАЙНЕ, и КОНФИДЕНЦИАЛЬНУЮ (статья 10, часть 2).
- конфиденциальная информация - документированная информация, доступ к которой ограничивается в соответствии с законодательством Российской Федерации (статья 2);
- ПЕРСОНАЛЬНЫЕ ДАННЫЕ о гражданах, включаемые в состав федеральных информационных ресурсов, информационных ресурсов совместного ведения, информационных ресурсов субъектов Российской Федерации, информационных ресурсов местного самоуправления, а также получаемые и собираемые негосударственными организациями, отнесены к категории конфиденциальной информации (статья 11, часть 1).

Из этого Закона следует:

- информация из любой области знаний и деятельности в принципе является открытой и общедоступной, если законодательством не предусмотрено ограничение доступа к ней в установленном порядке;
- категория "конфиденциальная информация", в соответствии с понятием, приведенным выше из статьи 2 "Закона об информации", объединяет все виды защищаемой информации (тайн). Это относится и к государственным и к негосударственным информационным ресурсам. При этом, исключение составляет информация, отнесенная к государственной тайне: она к конфиденциальной информации не относится, а является составной частью информации с ограниченным доступом (основание - статья 10, часть 2 указанного Закона).

- Отнесение информации к категориям осуществляется:
- к государственной тайне - в соответствии с Законом Российской Федерации "О государственной тайне";
- к конфиденциальной информации - в порядке, установленном законодательством РФ;
- к персональным данным о гражданах - федеральным законом.

О режимах защиты информации

В соответствии с "Законом об информации" режим защиты информации устанавливается (статья 21):

- в отношении сведений, отнесенных к ГОСУДАРСТВЕННОЙ ТАЙНЕ, - уполномоченными органами на основании Закона Российской Федерации "О государственной тайне";
- в отношении конфиденциальной информации - СОБСТВЕННИКОМ информационных ресурсов или уполномоченным лицом на основании "Закона об информации";
- в отношении персональных данных - отдельным федеральным законом.

Принципиальным здесь является положение, что режим защиты конфиденциальной информации определяет ее собственник, то есть соответствующий орган государственной власти или управления, организация, учреждение, предприятие.

О категории "служебная тайна"

Категория "СЛУЖЕБНАЯ ТАЙНА" ранее применялась для обозначения сведений ведомственного характера с грифом "секретно" и за ее разглашение предусматривалась уголовная ответственность. В настоящее время эта категория из Уголовного кодекса изъята и в прежнем ее понимании из правового поля исчезла в связи с принятием в июле Закона "О государственной тайне" по двум причинам:

- во-первых, информация с грифом "секретно" теперь составляет государственную тайну;
- во-вторых, применение грифов секретности для других категорий информации не допускается в соответствии со статьей 8 указанного Закона.

Вместе с тем, Гражданским кодексом Российской Федерации, предусмотрена категория "служебная тайна" в сочетании с категорией "коммерческая тайна". Статья 139 Кодекса гласит:

1. Информация составляет служебную или коммерческую тайну в случае, если информация имеет действительную или потенциальную коммерческую ценность в силу неизвестности ее третьим лицам, к ней нет свободного доступа на законном основании и обладатель информации принимает меры к охране ее конфиденциальности.
2. Информация, составляющая служебную или коммерческую тайну, защищается способами, предусмотренными настоящим Кодексом и другими законами.

О контроле состояния защиты конфиденциальной информации

Статьей 21 (часть 3) "Закона об информации" предусмотрен контроль со стороны органов государственной власти за соблюдением требований к защите информации с ограниченным доступом, порядок которого определяет Правительство Российской Федерации.

Это означает, что контроль состояния защиты должен охватывать все три составляющие информации с ограниченным доступом, входящей в государственные информационные ресурсы:

- информацию, составляющую государственную тайну;
- конфиденциальную информацию;
- персональные данные о гражданах.

При этом, контроль в равной степени должен охватывать и негосударственные структуры при наличии у них (при передаче им на законном основании) указанных видов информации, входящих в государственные информационные ресурсы.

Практические задания

1. Дайте определение понятия «государственная тайна», приведите пример и дайте перечень законов, регулирующих эту информацию.

2. Дайте определение понятия «коммерческая тайна», приведите пример и дайте перечень законов, регулирующих эту информацию.
3. Дайте определение понятия «служебная тайна», приведите пример и дайте перечень законов, регулирующих эту информацию.
4. Напишите перечень информации, относимой к коммерческой тайне для кондитерской фабрики.
5. Напишите заявление в суд в вольной форме с просьбой наказать похитителя коммерческой информации.

Контрольные вопросы

1. Что является основным источником права в области обеспечения информационной безопасности Российской Федерации?
2. Какой закон является базовым в области защиты информации? Регулирует ли он работу с информацией попадающей под определение «государственная тайна»?
3. Что такое документированная информация?
4. Имеет ли федеральный закон «Об информации, информатизации и защите информации» недостатки? Если да, то какие?
5. Можно ли сведения о состоянии экономики и потребностях населения, отнесенные к государственной тайне считать информацией с ограниченным доступом?
6. Считается ли конфиденциальной информация о заработанной плате сотрудников коммерческой организации?
7. Каковы функции Межведомственной комиссии?
8. Относится ли к функциям ФСТЭК координация деятельности органов государственной власти по защите государственной тайны в интересах разработки и выполнения государственных программ и нормативных документов?
9. Что является жизненно важными интересами в информационной сфере для личности, общества и государства

Практическое занятие № 2

Защита информации, содержащейся в информационных системах общего пользования

Цель занятия: Закрепление знаний об организации защиты информации, содержащейся в информационных системах общего пользования.

Краткие теоретические сведения

Организация защиты информации, содержащейся в информационных системах общего пользования регламентирована приказом ФСТЭК РФ от 31 августа 2010 г. №489.

Информационные системы общего пользования должны обеспечивать:

1. сохранность и неизменность обрабатываемой информации при попытках несанкционированных или случайных воздействий на нее в процессе обработки или хранения (далее - целостность информации);
2. беспрепятственный доступ пользователей к содержащейся в информационной системе общего пользования информации (далее - доступность информации);
3. защиту от действий пользователей в отношении информации, не предусмотренных правилами пользования информационной системой общего пользования, приводящих в том числе к уничтожению, модификации и блокированию информации (далее - неправомерные действия).

Информационные системы общего пользования включают в себя средства вычислительной техники, информационно-вычислительные комплексы и сети, средства и системы передачи, приема и обработки информации, средства изготовления, тиражирования документов и другие технические средства обработки речевой, графической, видео- и буквенно-цифровой информации, программные средства (операционные системы, системы управления базами данных и т.п.), средства защиты информации, применяемые в информационных системах.

Информация, содержащаяся в информационной системе общего пользования, является общедоступной. Информационные системы общего пользования в зависимости от значимости содержащейся в них информации и требований к ее защите разделяются на два класса.

1. К I классу относятся информационные системы общего пользования Правительства Российской Федерации и иные информационные системы общего пользования в случае, если нарушение целостности и доступности информации, содержащейся в них, может привести к возникновению угроз безопасности Российской Федерации. Отнесение информационных систем общего пользования к I классу проводится по решению руководителя соответствующего федерального органа исполнительной власти.
2. Ко II классу относятся информационные системы общего пользования, не указанные в подпункте 1 настоящего пункта.

Защита информации, содержащейся в информационных системах общего пользования, достигается путем исключения неправомерных действий в отношении указанной информации. Методы и способы защиты информации в информационных системах общего пользования определяются оператором информационной системы общего пользования и должны соответствовать настоящим Требованиям. Достаточность принятых мер по защите информации в информационных системах общего пользования оценивается при проведении мероприятий по созданию данных систем, а также в ходе мероприятий по контролю за их функционированием.

Работы по защите информации в информационных системах общего пользования являются неотъемлемой частью работ по созданию данных систем. Размещение информационных систем общего пользования, специальное оборудование и охрана помещений, в которых находятся технические средства, организация режима обеспечения безопасности в этих помещениях должны обеспечивать сохранность носителей информации и средств защиты информации, а также исключать возможность неконтролируемого проникновения или пребывания в этих помещениях посторонних лиц.

Защиту информации в информационных системах общего пользования обеспечивает оператор информационной системы общего пользования. В информационных системах общего пользования должны быть обеспечены:

1. поддержание целостности и доступности информации;
2. предупреждение возможных неблагоприятных последствий нарушения порядка доступа к информации;
3. проведение мероприятий, направленных на предотвращение неправомерных действий в отношении информации;
4. своевременное обнаружение фактов неправомерных действий в отношении информации;
5. недопущение воздействия на технические средства информационной системы общего пользования, в результате которого может быть нарушено их функционирование;
6. возможность оперативного восстановления информации, модифицированной или уничтоженной вследствие неправомерных действий;
7. проведение мероприятий по постоянному контролю за обеспечением их защищенности;

возможность записи и хранения сетевого трафика.

Мероприятия по обеспечению защиты информации в информационных системах общего пользования включают в себя:

1. определение угроз безопасности информации, формирование на их основе модели угроз;
2. разработку на основе модели угроз системы защиты информации, обеспечивающей нейтрализацию предполагаемых угроз с использованием методов и способов защиты информации, предусмотренных для соответствующего класса информационных систем общего пользования;
3. проверку готовности средств защиты информации к использованию с составлением заключений о возможности их эксплуатации;
4. установку и ввод в эксплуатацию средств защиты информации в соответствии с эксплуатационной и технической документацией;
5. обучение лиц, использующих средства защиты информации, применяемые в информационной системе общего пользования, правилам работы с ними;
6. учет применяемых средств защиты информации, эксплуатационной и технической документации к ним;
7. контроль за соблюдением условий использования средств защиты информации, предусмотренных эксплуатационной и технической документацией;
8. проведение разбирательств и составление заключений по фактам несоблюдения условий использования средств защиты информации, которые могут привести к нарушению безопасности информации или другим нарушениям, снижающим уровень защищенности информационной системы общего пользования, разработку и принятие мер по предотвращению возможных опасных последствий подобных нарушений;
9. описание системы их защиты.

Для разработки и осуществления мероприятий по защите информации в информационных системах общего пользования оператором информационной системы общего пользования назначается структурное подразделение или должностное лицо (работник), ответственные за обеспечение защиты информации.

Запросы пользователей на получение информации, содержащейся в информационных системах общего пользования, а также факты предоставления информации по этим запросам регистрируются автоматизированными средствами информационных систем общего пользования в электронном журнале обращений. Содержание электронного журнала обращений периодически проверяется соответствующими должностными лицами (работниками) оператора информационной системы общего пользования.

При обнаружении нарушений порядка доступа к информации оператор информационной системы общего пользования организует работы по выявлению причин нарушений и устранению этих причин в установленном порядке. Подсистема информационной безопасности должна обеспечивать восстановление информации в информационной системе общего

пользования, модифицированной или уничтоженной вследствие неправомерных действий в отношении такой информации. Время восстановления процесса предоставления информации пользователям не должно превышать 8 часов.

Реализация требований по обеспечению защиты информации в средствах защиты информации возлагается на их разработчиков. При создании и эксплуатации информационных систем общего пользования должны выполняться следующие требования по защите информации:

1. В информационных системах общего пользования I класса:
   - использование средств защиты информации от неправомерных действий, в том числе средств криптографической защиты информации (электронной цифровой подписи, при этом средства электронной цифровой подписи обязательно должны применяться к публикуемому информационному наполнению), сертифицированных ФСБ России;
   - использование средств обнаружения вредоносного программного обеспечения, в том числе антивирусных средств, сертифицированных ФСБ России;
   - использование средств контроля доступа к информации, в том числе средств обнаружения компьютерных атак, сертифицированных ФСБ России;
   - использование средств фильтрации и блокирования сетевого трафика, в том числе средств межсетевого экранирования, сертифицированных ФСБ России;
   - осуществление локализации и ликвидации неблагоприятных последствий нарушения порядка доступа к информации;
   - осуществление записи и хранения сетевого трафика при обращении к государственным информационным ресурсам за десять и более последних дней и предоставление доступа к записям по запросам уполномоченных государственных органов, осуществляющих оперативно-розыскную деятельность;
   - обеспечение защиты от воздействий на технические и программные средства информационных систем общего пользования, в результате которых нарушается их функционирование, и несанкционированного доступа к помещениям, в которых находятся данные средства, с использованием технических средств охраны, в том числе систем видеонаблюдения, предотвращающих проникновение в помещения посторонних лиц;
   - осуществление регистрации действий обслуживающего персонала и пользователей;
   - обеспечение резервирования технических и программных средств, дублирования носителей и массивов информации;
   - использование сертифицированных в установленном порядке систем обеспечения гарантированного электропитания (источников бесперебойного питания);
   - осуществление мониторинга их защищенности уполномоченным подразделением ФСБ России;
   - введение в эксплуатацию только после направления оператором информационной системы общего пользования в ФСБ России уведомления о готовности ввода информационной системы общего пользования в эксплуатацию и ее соответствии настоящим Требованиям.
2. В информационных системах общего пользования II класса:
   - использование средств защиты информации от неправомерных действий, сертифицированных ФСБ России и (или) ФСТЭК России с учетом их компетенции, в том числе средств криптографической защиты информации (электронной цифровой подписи, при этом средства электронной цифровой подписи должны применяться к публикуемому информационному наполнению);
   - использование средств обнаружения вредоносного программного обеспечения, в том числе антивирусных средств, сертифицированных ФСБ России и (или) ФСТЭК России с учетом их компетенции;

- использование средств контроля доступа к информации, в том числе средств обнаружения компьютерных атак, сертифицированных ФСБ России и (или) ФСТЭК России с учетом их компетенции;
- использование средств фильтрации и блокирования сетевого трафика, в том числе средств межсетевого экранирования, сертифицированных ФСБ России и (или) ФСТЭК России с учетом их компетенции;
- осуществление локализации и ликвидации неблагоприятных последствий нарушения порядка доступа к информации;
- осуществление записи и хранения сетевого трафика при обращении к государственным информационным ресурсам за последние сутки и более и предоставление доступа к записям по запросам уполномоченных государственных органов, осуществляющих оперативно-розыскную деятельность;
- обеспечение защиты от воздействий на технические и программные средства информационных систем общего пользования, в результате которых нарушается их функционирование, и несанкционированного доступа к помещениям, в которых находятся данные средства;
- осуществление регистрации действий обслуживающего персонала;
- обеспечение частичного резервирования технических средств и дублирования массивов информации;
- использование систем обеспечения гарантированного электропитания (источников бесперебойного питания);
- осуществление мониторинга их защищенности уполномоченным подразделением ФСБ России;
- введение в эксплуатацию только после направления оператором информационной системы общего пользования в ФСТЭК России уведомления о готовности ввода информационной системы общего пользования в эксплуатацию и ее соответствии настоящим Требованиям.

Практические задания

Для заданной преподавателем информационной системы, составить документ, который содержит следующую информацию:

1. угрозы безопасности информации, модели угроз;
2. описание системы защиты информации, обеспечивающей нейтрализацию предполагаемых угроз с использованием методов и способов защиты информации, предусмотренных для соответствующего класса информационных систем общего пользования;
3. заключений о возможности эксплуатации средств защиты информации;
4. учет применяемых средств защиты информации, эксплуатационной и технической документации к ним;
5. определение лиц, осуществляющих контроль за соблюдением условий использования средств защиты информации, предусмотренных эксплуатационной и технической документацией;

Контрольные вопросы

1. Дайте определение информационной системы общего пользования?
2. Особенности информационных системах общего пользования I класса?
3. Особенности информационных системах общего пользования II класса?
4. Как достигается защита информации, содержащейся в информационных системах общего пользования?
5. Что регламентировано приказом ФСТЭК РФ от 31 августа 2010 г. №489?

Практическое занятие № 3

Разработка базового блока документов для обеспечения информационной безопасности ИСПДн

Краткие теоретические сведения

Цель занятия: Закрепление знаний о видах конфиденциальной информации: персональных данных, служебной тайне, коммерческой тайне, банковской тайне, тайне следствия и судопроизводства, профессиональной тайне; о правовом обеспечении защиты конфиденциальной информации; о правовом регулировании отношений предприятия с другими предприятиями, организациями и гражданами по защите коммерческой тайны.

Краткие теоретические сведения

Конфиденциальная информация – документированная информация, доступ к которой ограничивается в соответствии с законодательством Российской Федерации, не являющаяся общедоступной информацией и в случае разглашения способная нанести ущерб правам и охраняемым законом интересам предоставившего ее лица.

Перечень сведений конфиденциального характера изложен в Указе Президента РФ № 188 и в общих чертах относит к таковым следующую информацию: персональные данные, сведения, составляющие тайну следствия и судопроизводства, служебную, профессиональную и коммерческую тайну и сведения о сущности изобретения, полезной модели или промышленного образца до официальной публикации информации о них.

Персональные данные. Тайна личной и семейной жизни

В настоящее время в России не существует закона, посвященного охране неприкосновенности частной жизни, законодательно урегулированы лишь некоторые вопросы, связанные с этой проблемой (например, охрана личной и семейной тайны).

Статья 24 Конституции запрещает сбор, хранение, использование и распространение информации о частной жизни лица без его согласия.

Коммерческая и служебная тайна

Закон относит к коммерческой и служебной тайне сведения, имеющие действительную или потенциальную коммерческую ценность в силу их неизвестности третьим лицам, к которым нет доступа на законном основании и по отношению к которым их обладатель принимает меры по охране их конфиденциальности.

Определение перечня сведений, составляющих коммерческую тайну, находится в компетенции их обладателя.

К коммерческой тайне может относиться научно-техническая, технологическая, производственная информация, в том числе секреты производства (ноу-хау), финансово-экономическая и иная информация, которая имеет действительную или потенциальную коммерческую ценность в силу неизвестности ее третьим лицам, к которой нет свободного доступа на законном основании и к охране конфиденциальности которой обладатель коммерческой тайны принял следующие меры:

- определил перечень информации, составляющей коммерческую тайну;
- ограничил свободный доступ к ней путем установления порядка обращения с этой информацией и контроля его соблюдения;
- организовал договорное регулирование отношений с работниками и с контрагентами по вопросам условий передачи и использования информации, составляющей коммерческую тайну;
- нанес на материальные носители информации, составляющей коммерческую тайну, и (или) сопроводительные документы гриф «Коммерческая тайна» с указанием ее обладателя.

Режим коммерческой тайны считается установленным после принятия обладателем коммерческой тайны указанных мер.

Основными признаками коммерческой тайны современного предприятия является конфиденциальность сведений, отношений, переписки и переговоров между сторонами или сделок. К другим признакам коммерческой тайны, проявляющимся при ее разглашении, относятся:

- нанесение экономического ущерба компании;
- возникновение у компании убытков в виде упущенной выгоды;
- снижение экономической, технической эффективности деятельности компании, в том числе и внешнеэкономической;
- нанесение ущерба имиджу компании и дискредитация его как добросовестного, надежного партнера по внешнеэкономической и иной деятельности;
- нанесение ущерба престижу и репутации партнера, с которым заключается или осуществлена коммерческая сделка.

Статья 183 Уголовного кодекса также предусматривает ответственность за незаконные получение и разглашение сведений, составляющих коммерческую тайну.

Уголовная ответственность наступает, во-первых, за собирание таких сведений путем похищения документов, подкупа или угроз или иным незаконным способом с целью разглашения или незаконного использования и, во-вторых, за незаконные разглашение или использование сведений, составляющих коммерческую тайну, без согласия их владельца, совершенные из корыстной или иной личной заинтересованности и причинившие крупный ущерб.

Правовое обеспечение защиты конфиденциальной информации на предприятии

Для защиты конфиденциальной информации на предприятии, должны быть разработаны следующие документы:

- Перечень сведений, составляющих конфиденциальную информацию предприятия;
- Договорное обязательство о неразглашении КИ;
- Инструкция по защите конфиденциальной информации;
- Инструкция о работе с иностранными фирмами и их представителями;
- Соглашение о конфиденциальности (между организациями).

Инструкция по обеспечению безопасности конфиденциальной информации должна быть разработана в соответствии с нормативными актами, нормативно-методических документов по делопроизводству и архивному делу Российской Федерации. Она должна состоять из следующих разделов:

1. Общие положения.
2. Конфиденциальная информация.
3. Ответственность за разглашение конфиденциальной информации.
4. Система допуска сотрудников к сведениям, составляющим конфиденциальной информации.
   4.1. Круг лиц, имеющих право давать на доступ к конфиденциальной информации.
   4.2. Порядок оформления разрешения на доступ к конфиденциальным документам.
   4.3. Порядок доступа на совещания по вопросам, содержащим конфиденциальные сведения.
5. Подготовка и издание конфиденциальных документов.
6. Учет, прохождение и отправление изданных конфиденциальных документов.
7. Прием, учёт и прохождение поступивших документов.
8. Учет документов выделенного хранения.
9. Учёт журналов и картотек.
10. Организация хранения конфиденциальных документов.
11. Организация и технология контроля исполнения конфиденциальных документов.
12. Размножение документов
13. Уничтожение документов
14. Составление и оформление дел с грифом «Конфиденциально»
15. Формирование и оформление дел
16. Проверка наличия конфиденциальных документов.
17. Подготовка конфиденциальных документов на архивное хранение

18. Порядок передачи конфиденциальных документов в архив.

Практические задания

1-18. В соответствии с номером раздела выбрать свой вариант и разработать документ, являющийся составной частью «Инструкции по обеспечению безопасности конфиденциальной информации».

19. Разработать текст соглашения о конфиденциальности (между организациями).
20. Разработать инструкцию о работе с иностранными фирмами и их представителями.
21. Составить список сведений, составляющих коммерческую тайну конкретного предприятия.

Контрольные вопросы

1. Дайте определение конфиденциальной информации.
2. Государственная тайна является конфиденциальной информацией? Ответ обоснуйте, опираясь на действующее законодательство.
3. Перечислите виды конфиденциальной информации.
4. Что такое «персональные данные»? Приведите примеры.
5. Дайте определение коммерческой тайне.
6. После каких мер, принятых обладателем коммерческой тайны, режим коммерческой тайны считается установленным?
7. Какие документы должны быть разработаны для защиты конфиденциальной информации на предприятии?
8. Перечислите основные разделы «Инструкция по обеспечению безопасности конфиденциальной информации».

Практическое занятие № 4

Подготовка документов к получению лицензии

Цель работы: Закрепление знаний о видах деятельности в области защиты информации, подлежащих обязательному государственному лицензированию, изучение нормативно-правовых документов.

Краткие теоретические сведения

Лицензирование

Полный перечень видов деятельности в области защиты информации, подлежащих обязательному государственному лицензированию, определён в Законе Российской Федерации "О государственной тайне" от 21.07.1993 N 5485-1 (ред. от 26.07.2017) и Федеральном законе "О лицензировании отдельных видов деятельности" (04.05.2011 N 99-ФЗ).

В законе "О государственной тайне" определены лицензируемые виды деятельности в области защиты информации, содержащей сведения, отнесённые к государственной тайне, а в Законе "О лицензировании отдельных видов деятельности" - в области защиты конфиденциальной информации.

Лицензированию Федеральным агентством в соответствии с его компетенцией подлежит деятельность по следующим направлениям:

– создание средств защиты информации;

– осуществление мероприятий и оказание услуг по защите государственной тайны;

– деятельность, связанная с шифровальными средствами;

– предоставление услуг в области шифрования информации;

– выявление технических устройств скрытого съема информации, электронных закладных устройств и программных закладок в технических средствах и помещениях государственных структур;

– создание систем и комплексов телекоммуникаций органов государственной власти Российской Федерации;

– создание закрытых (защищенных) с использованием шифровальных средств систем и комплексов телекоммуникаций;

– создание и реализация средств выявления технических устройств скрытого съема информации, электронных закладных устройств и программных закладок.

Указанные направления включают определенное множество отдельных видов деятельности к которым относятся разработка, производство, реализация (продажа), эксплуатация, монтаж, установка (инсталляция), наладка, сертификационные испытания, ввоз в страну, вывоз из страны и другие..

Основные лицензионные требования и условия

В соответствии с вышеперечисленными законодательными документами основными лицензионными требованиями и условиями являются:

— заявление о выдаче лицензии (с указанием: наименования и организационно-правовой формы, юридического и почтового адреса предприятия, номера его расчетного счета в банке, кода ОКПО; вида деятельности, на осуществление которого должна быть выдана лицензия; срока действия лицензии);

— перечень имеющегося на предприятии производственного, испытательного и контрольно-измерительного оборудования, необходимого для обеспечения заявленных видов деятельности; перечень имеющейся на предприятии нормативной и методической документации, необходимой для обеспечения заявленных видов деятельности;

— сведения о составе и квалификации научного, инженерно-технического персонала предприятия (подразделения), занимающегося защитой информации по заявленным видам деятельности; представление органа государственной власти; копии учредительных документов (с представлением оригиналов, в случае, если копии не заверены нотариусом); копия свидетельства о государственной регистрации предприятия;

— материалы экспертизы, подтверждающие наличие необходимых условий для осуществления заявленных видов деятельности, а также профессиональную пригодность руководителя предприятия-заявителя, или лиц, уполномоченных им для руководства лицензируемой деятельностью;

— справка о постановке на учет в налоговом органе, об уплате налогов и представлении налоговой отчетности; документ, подтверждающий оплату рассмотрения заявления.

Основанием для отказа в выдаче лицензии является:

- наличие в документах, представленных заявителем, недостоверной или искаженной информации;
- отрицательное экспертное заключение, установившее несоответствие условий, необходимых для осуществления заявленных видов деятельности, соответствующим требованиям;
- недостаточный уровень профессиональной подготовки руководителя предприятия-заявителя, или лиц, уполномоченных им для руководства лицензируемой деятельностью;
- нарушение заявителем законодательства Российской Федерации, требований соблюдения налоговой дисциплины и нормативных документов по защите информации;
- невыполнение заявителем предписаний и распоряжений государственных органов или приостановление ими деятельности предприятия в соответствии с законами Российской Федерации.

Практические задания

1. Разработать текст заявления с просьбой выдать лицензию для деятельности, связанной с шифровальными устройствами;
2. Разработать текст заявления с просьбой выдать лицензию для осуществления мероприятий и оказанию услуг по защите государственной тайны;
3. Разработать текст заявления с просьбой выдать лицензию для предоставления услуг в области шифрования информации;

Контрольные вопросы

1. Что такое лицензия в области защиты информации в соответствии с положением о государственном лицензировании?
2. Что такое лицензирование в области защиты информации в соответствии с положением о государственном лицензировании?

Практическое занятие № 5

Подготовка документов к сертификации

Цель работы: Закрепление знаний о правовом регулировании сертификации и лицензировании в сфере защиты информации. Получение навыков по средствам и методам метрологии в сфере защиты информации.

Краткие теоретические сведения

Стандартизация

Стандартизация - это деятельность по установлению норм, правил и характеристик (далее - требования) в целях обеспечения:

безопасности продукции, работ и услуг для окружающей среды, жизни, здоровья и имущества; технической и информационной совместимости, а также взаимозаменяемости продукции;

качества продукции, работ и услуг в соответствии с уровнем развития науки, техники и технологии; единства измерений;

экономии всех видов ресурсов;

безопасности хозяйственных объектов с учетом риска возникновения природных и техногенных катастроф и других чрезвычайных ситуаций;

обороноспособности и мобилизационной готовности страны.

Государственное управление стандартизацией в Российской Федерации, включая координацию деятельности государственных органов управления Российской Федерации, взаимодействие с органами власти республик в составе Российской Федерации, краев, областей, автономной области, автономных округов, городов, с общественными объединениями, в том числе с техническими комитетами по стандартизации, с субъектами хозяйственной деятельности, осуществляет Комитет Российской Федерации по стандартизации, метрологии и сертификации (Госстандарт России).

Госстандарт России формирует и реализует государственную политику в области стандартизации, осуществляет государственный контроль и надзор за соблюдением обязательных требований государственных стандартов, участвует в работах по международной (региональной), стандартизации, организует профессиональную подготовку и переподготовку кадров в области стандартизации, а также устанавливает правила применения международных (региональных) стандартов, правил, норм и рекомендаций по стандартизации на территории Российской Федерации, если иное не установлено международными договорами Российской Федерации.

Метрология

Государственное управление деятельностью по обеспечению единства измерений в Российской Федерации осуществляет Комитет Российской Федерации по стандартизации, метрологии и сертификации (Госстандарт России). К компетенции Госстандарта России относится:

- межрегиональная и межотраслевая координация деятельности по обеспечению единства измерений в Российской Федерации;
- представление Правительству Российской Федерации предложений по единицам величин, допускаемым к применению;
- установление правил создания, утверждения, хранения и применения эталонов единиц величин;
- определение общих метрологических требований к средствам, методам и результатам измерений;
- осуществление государственного метрологического контроля и надзора;
- осуществление контроля за соблюдением условий международных договоров Российской Федерации о признании результатов испытаний и поверки средств измерений;

– руководство деятельностью Государственной метрологической службы и иных государственных служб обеспечения единства измерений;
– участие в деятельности международных организаций по вопросам обеспечения единства измерений.

Сертификация

Постоянное усложнение средств и систем обработки информации, программных продуктов увеличивает вероятность возникновения непреднамеренных дефектов, которые могут исказить информацию или повлиять на процесс её обработки, вместе с тем нельзя исключить возможность преднамеренного воздействия на информацию как в условиях недобросовестной конкуренции, так и информационного противоборства.

Для того, чтобы защита информации была максимально эффективна, средства её обработки и защиты должны соответствовать определённым требованиям, подтверждённым объективной и независимой оценкой. Одной из форм такой оценки соответствия требованиям нормативных документов по защите информации является сертификация.

Сертификации Федеральным агентством в соответствии с его компетенцией подлежат:

– средства (системы, комплексы) криптографической защиты информации;
– средства выявления закладных устройств и программных закладок;
– защищенные технические средства обработки информации;
– закрытые (защищенные) информационные системы и комплексы телекоммуникаций.

Порядок проведения сертификации средств защиты информации ФСБ основан на следующих основных принципах и правилах:

1. Обязательность сертификации изделий, обеспечивающих защиту государственной тайны, или обязательность сертификации которых установлена нормативными актами Российской Федерации.

2. Обязательность использования криптографических алгоритмов, являющихся стандартами или ранее рекомендованных либо разработанных ФСБ.

3. Принятие на сертификацию только изделий от заявителей, имеющих лицензию ФСБ на соответствующие виды деятельности.

4. Принятие на сертификацию только готовых изделий в целом, а не их составных частей или отдельных компонент.

5. Процедура сертификации осуществляется в отношении только технических средств или технической части системы защиты, с учетом условий их эксплуатации.

6. Двухступенчатость процесса сертифицикации при независимости организаций, проводящих экспертизу и сертификационные испытания: сертификация средств защиты информации осуществляется Центральным органом по сертификации, а испытания проводятся в аккредитованных испытательных центрах (лабораториях).

7. Дифференцированность подхода к уровню защиты различных видов информации.

8. Принятие Центральным органом по сертификации и испытательными центрами (лабораториями) на себя ответственности за выполнение возложенных на них функций в соответствии с действующим законодательством и договорными обязательствами.

9. Сертификационные испытания средств криптографической защиты информации могут осуществляться только аккредитованными ФСБ испытательными сертификационными центрами. Действительное соответствие изделия государственным стандартам и гарантию удовлетворения требований по безопасности к шифровальным средствам соответствующего класса удостоверяет только сертификат ФСБ.

10. Принятие и неуклонное соблюдение заявителем правил, установленных в системе сертификации.

Практические задания

1. Разработать текст заявления с просьбой провести сертификацию системы защиты информации использующую шифровальные средств;

2. Разработать текст заявления с просьбой выдать сертификат оборудованию, разработанному для защиты телекоммуникационных систем;
3. Соберите документы и правильно составьте запрос на прохождение вашим оборудованием процедуры на контроль единства измерений.
4. Полностью смоделируйте процесс получения сертификата средств криптографической защиты информации

Контрольные вопросы

1. На основании чего выдаются сертификат.
2. Что является основными сертификационными требованиями и условиями?
3. В каком нормативном акте определены полномочия ФСТЭК России (бывш. Гостехкомиссия России)?
4. Что такое средство измерения?
5. Когда проходит сертификация средств защиты информации?
6. Какие организации могут осуществлять сертификационные испытания средств криптографической защиты?

Практическое занятие № 6

Подготовка документов к аттестации объектов информатизации службы информационной безопасности

Цель занятия: Закрепить знания об основных понятиях в области аттестации объекта информатизации по требованиям безопасности, рассмотреть участников стандартной схемы аттестации и этапы аттестации.

Краткие теоретические сведения

Деятельность по аттестации объектов информатизации по требованиям безопасности информации осуществляет ФСТЭК России (бывш. Гостехкомиссия России). Для начала дадим определение объекта информатизации.

Объект информатизации - совокупность информационных ресурсов, средств и систем обработки информации, используемых в соответствии с заданной информационной технологией, средств обеспечения объекта информатизации, помещений или объектов (зданий, сооружений, технических средств), в которых они установлены, или помещения и объекты, предназначенные для ведения конфиденциальных переговоров.

Аттестация объектов информатизации (далее аттестация) - комплекс организационно-технических мероприятий, в результате которых посредством специального документа - "Аттестата соответствия" подтверждается, что объект соответствует требованиям стандартов или иных нормативно-технических документов по безопасности информации, утвержденных ФСТЭК России (Гостехкомиссией России). Наличие аттестата соответствия в организации дает право обработки информации с уровнем секретности (конфиденциальности) на период времени, установленный в аттестате.

Аттестация производится в порядке, установленном "Положением по аттестации объектов информатизации по требованиям безопасности информации" от 25 ноября 1994 года. Аттестация должна проводится до начала обработки информации, подлежащей защите. Это необходимо в целях официального подтверждения эффективности используемых мер и средств по защите этой информации на конкретном объекте информатизации.

Аттестация является обязательной в следующих случаях:

- государственная тайна;
- при защите государственного информационного ресурса;
- управление экологически опасными объектами;
- ведение секретных переговоров.

Во всех остальных случаях аттестация носит добровольный характер, то есть может осуществляться по желанию заказчика или владельца объекта информатизации.

Аттестация предполагает комплексную проверку (аттестационные испытания) объекта информатизации в реальных условиях эксплуатации. Целью является проверка соответствия применяемых средств и мер защиты требуемому уровню безопасности. К проверяемым требованиям относится:

- защита от НСД, в том числе компьютерных вирусов;
- защита от утечки через ПЭМИН;
- защита от утечки или воздействия на информацию за счет специальных устройств, встроенных в объект информатизации.

Аттестация проводится органом по аттестации в соответствии со схемой, выбираемой этим органом, и состоит из следующего перечня работ:

- анализ исходных данных по аттестуемому объекту информатизации;
- предварительное ознакомление с аттестуемым объектом информатизации;
- проведение экспертного обследования объекта информатизации и анализ разработанной документации по защите информации на этом объекте с точки зрения ее соответствия требованиям нормативной и методической документации;

- проведение испытаний отдельных средств и систем защиты информации на аттестуемом объекте информатизации с помощью специальной контрольной аппаратуры и тестовых средств;
- проведение испытаний отдельных средств и систем защиты информации в испытательных центрах (лабораториях) по сертификации средств защиты информации по требованиям безопасности информации;
- проведение комплексных аттестационных испытаний объекта информатизации в реальных условиях эксплуатации;
- анализ результатов экспертного обследования и комплексных аттестационных испытаний объекта информатизации и утверждение заключения по результатам аттестации.

Органы по аттестации должны проходить аккредитацию ФСТЭК в соответствии с "Положением об аккредитации испытательных лабораторий и органов по сертификации средств защиты информации по требованиям безопасности информации". Все расходы по проведению аттестации возлагаются на заказчика, как в случае добровольной, так и обязательной аттестации. Органы по аттестации несут ответственность за выполнение своих функций, за сохранение в секрете информации, полученной в ходе аттестации, а также за соблюдение авторских прав заказчика.

В структуру системы аттестации входят:

- федеральный орган по сертификации средств защиты информации и аттестации объектов информатизации по требованиям безопасностиинформации – ФСТЭК России;
- органы по аттестации объектов информатизации по требованиям безопасности информации;
- испытательные центры (лаборатории) по сертификации продукции по требованиям безопасности информации;
- заявители (заказчики, владельцы, разработчики аттестуемых объектов информатизации).

В качестве заявителей могут выступать заказчики, владельцы или разработчики аттестуемых объектов информатизации. В качестве органов по аттестации могут выступать отраслевые и региональные учреждения, предприятия и организации по защите информации, специальные центры ФСТЭК России, которые прошли соответствующую аккредитацию.

Органы по аттестации:

- аттестуют объекты информатизации и выдают "Аттестаты соответствия";
- осуществляют контроль за безопасностью информации, циркулирующей на аттестованных объектах информатизации, и за их эксплуатацией;
- отменяют и приостанавливают действие выданных этим органом "Аттестатов соответствия";
- формируют фонд нормативной и методической документации, необходимой для аттестации конкретных типов объектов информатизации, участвуют в их разработке;
- ведут информационную базу аттестованных этим органом объектов информатизации;
- осуществляют взаимодействие с ФСТЭК России и ежеквартально информируют его о своей деятельности в области аттестации.

ФСТЭК осуществляет следующие функции в рамках системы аттестации:

1. организует обязательную аттестацию объектов информатизации;
2. создает системы аттестации объектов информатизации и устанавливает правила для проведения аттестации в этих системах;
3. устанавливает правила аккредитации и выдачи лицензий на проведение работ по обязательной аттестации;

4. организует, финансирует разработку и утверждает нормативные и методические документы по аттестации объектов информатизации;
5. аккредитует органы по аттестации объектов информатизации и выдает им лицензии на проведение определенных видов работ;
6. осуществляет государственный контроль и надзор за соблюдением правил аттестации и эксплуатацией аттестованных объектов информатизации;
7. рассматривает апелляции, возникающие в процессе аттестации объектов информатизации, и контроля за эксплуатацией аттестованных объектов информатизации;
8. организует периодическую публикацию информации по функционированию системы аттестации объектов информатизации по требованиям безопасности информации.

Испытательные лаборатории проводят испытания несертифицированной продукции, используемой на аттестуемом объекте информатизации. Со списком органов по аттестации и испытательных лабораторий, прошедших аккредитацию, можно ознакомиться на официальном сайте ФСТЭК России в разделе "Сведения о Системе сертификации средств защиты информации по требованиям безопасности информации".

Заявители:

- проводят подготовку объекта информатизации для аттестации путем реализации необходимых организационно-технических мероприятий по защите информации;
- привлекают органы по аттестации для организации и проведения аттестации объекта информатизации;
- предоставляют органам по аттестации необходимые документы и условия для проведения аттестации;
- привлекают, в необходимых случаях, для проведения испытаний несертифицированных средств защиты информации, используемых на аттестуемом объекте информатизации, испытательные центры (лаборатории) по сертификации;
- осуществляют эксплуатацию объекта информатизации в соответствии с условиями и требованиями, установленными в "Аттестате соответствия";
- извещают орган по аттестации, выдавший "Аттестат соответствия", о всех изменениях в информационных технологиях, составе и размещении средств и систем информатики, условиях их эксплуатации, которые могут повлиять на эффективность мер и средств защиты информации (перечень характеристик, определяющих безопасность информации, об изменениях которых требуется обязательно извещать орган по аттестации, приводится в "Аттестате соответствия");
- предоставляют необходимые документы и условия для осуществления контроля и надзора за эксплуатацией объекта информатизации, прошедшего обязательную аттестацию.

Для проведения испытаний заявитель предоставляет органу по аттестации следующие документы и данные:

- приемо-сдаточную документацию на объект информатизации;
- акты категорирования выделенных помещений и объектов информатизации;
- инструкции по эксплуатации средств защиты информации;
- технический паспорт на аттестуемый объект;
- документы на эксплуатацию (сертификаты соответствия требованиям безопасности информации) ТСОИ;
- сертификаты соответствия требованиям безопасности информации на ВТСС;
- сертификаты соответствия требованиям безопасности информации на технические средства защиты информации;
- акты на проведенные скрытые работы;

- протоколы измерения звукоизоляции выделенных помещений и эффективности экранирования сооружений и кабин (если они проводились);
- протоколы измерения величины сопротивления заземления;
- протоколы измерения реального затухания информационных сигналов до мест возможного размещения средств разведки;
- данные по уровню подготовки кадров, обеспечивающих защиту информации;
- данные о техническом обеспечении средствами контроля эффективности защиты информации и их метрологической поверке;
- нормативную и методическую документацию по защите информации и контролю эффективности защиты.

Приведенный общий объем исходных данных и документации может уточняться заявителем в зависимости от особенностей аттестуемого объекта информатизации по согласованию с аттестационной комиссией.

- пояснительную записку, содержащую информационную характеристику и организационную структуру объекта защиты, сведения об организационных и технических мероприятиях по защите информации от утечки по техническим каналам;
- перечень объектов информатизации, подлежащих защите, с указанием мест их расположения и установленной категории защиты;
- перечень выделенных помещений, подлежащих защите, с указанием мест их расположения и установленной категории защиты;
- перечень устанавливаемых ТСОИ с указанием наличия сертификата (предписания на эксплуатацию) и мест их установки;
- перечень устанавливаемых ВТСС с указанием наличия сертификата и мест их установки;
- перечень устанавливаемых технических средств защиты информации с указанием наличия сертификата и мест их установки;
- схему (в масштабе) с указанием плана здания, в котором расположены защищаемые объекты, границы контролируемой зоны, трансформаторной подстанции, заземляющего устройства, трасс прокладки инженерных коммуникаций, линий электропитания, связи, пожарной и охранной сигнализации, мест установки разделительных устройств и т.п.;
- технологические поэтажные планы здания с указанием мест расположения объектов информатизации и выделенных помещений и характеристиками их стен, перекрытий, материалов отделки, типов дверей и окон;
- планы объектов информатизации с указанием мест установки ТСОИ, ВТСС и прокладки их соединительных линий, а также трасс прокладки инженерных коммуникаций и посторонних проводников;
- план-схему инженерных коммуникаций всего здания, включая систему вентиляции;
- план-схему системы заземления объекта с указанием места расположения заземлителя;
- план-схему системы электропитания здания с указанием места расположения разделительного трансформатора (подстанции), всех щитов и разводных коробок;
- план-схему прокладки телефонных линий связи с указанием мест расположения распределительных коробок и установки телефонных аппаратов;
- план-схему систем охранной и пожарной сигнализации с указанием мест установки и типов датчиков, а также распределительных коробок;
- схемы систем активной защиты (если они предусмотрены).
- Порядок проведения аттестации объектов информатизации по требованиям безопасности информации включает следующие действия:

- подача и рассмотрение заявки на аттестацию. Заявка имеет установленную форму, с которой можно ознакомиться в "Положении об аттестации объектов информатизации по требованиям безопасности". Заявитель направляет заявку в орган по аттестации, который в месячный срок рассматривает заявку, выбирает схему аттестации и согласовывает ее с заявителем.
- предварительное ознакомление с аттестуемым объектом – производится в случае недостаточности предоставленных заявителем данных до начала аттестационных испытаний;
- испытание в испытательных лабораториях несертифицированных средств и систем защиты информации, используемых на аттестуемом объекте.
- разработка программы и методики аттестационных испытаний. Этот шаг является результатом рассмотрения исходных данных и предварительного ознакомления с аттестуемым объектом. Орган по аттестации определяет перечень работ и их продолжительность, методику испытаний, состав аттестационной комиссии, необходимость использования контрольной аппаратуры и тестовых средств или участия испытательных лабораторий. Программа аттестационных испытаний согласовывается с заявителем.
- заключение договоров на аттестацию. Результатом предыдущих четырех этапов становится заключение договора между заявителем и органом по аттестации, заключением договоров между органом по аттестации и привлекаемыми экспертами и оформлением предписания о допуске аттестационной комиссии к проведению аттестации.
- проведение аттестационных испытаний объекта информатизации. В ходе аттестационных испытаний выполняется следующее:
- анализ организационной структуры объекта информатизации, информационных потоков, состава и структуры комплекса технических средств и программного обеспечения, системы защиты информации на объекте, разработанной документации и ее соответствия требованиям нормативной документации по защите информации;
- определяется правильность категорирования объектов ЭВТ и классификации АС (при аттестации автоматизированных систем), выбора и применения сертифицированных и несертифицированных средств и систем защиты информации;
- проводятся испытания несертифицированных средств и систем защиты информации на аттестуемом объекте или анализ результатов их испытаний в испытательных центрах (лабораториях) по сертификации;
- проверяется уровень подготовки кадров и распределение ответственности персонала за обеспечение выполнения требований по безопасности информации;
- проводятся комплексные аттестационные испытания объекта информатизации в реальных условиях эксплуатации путем проверки фактического выполнения установленных требований на различных этапах технологического процесса обработки защищаемой информации;
- оформляются протоколы испытаний и заключение по результатам аттестации с конкретными рекомендациями по устранению допущенных нарушений, приведению системы защиты объекта информатизации в соответствие с установленными требованиями и совершенствованию этой системы, а также рекомендациями по контролю за функционированием объекта информатизации.

К заключению прилагаются протоколы испытаний, подтверждающие полученные при испытаниях результаты и обосновывающие приведенный в заключении вывод.

Протокол аттестационных испытаний должен включать:

- вид испытаний;

- объект испытаний;
- дату и время проведения испытаний;
- место проведения испытаний;
- перечень использованной в ходе испытаний аппаратуры (наименование, тип, заводской номер, номер свидетельства о поверке и срок его действия);
- перечень нормативно-методических документов, в соответствии с которыми проводились испытания;
- методику проведения испытания (краткое описание);
- результаты измерений;
- результаты расчетов;
- выводы по результатам испытаний

Протоколы испытаний подписываются экспертами – членами аттестационной комиссии, проводившими испытания, с указанием должности, фамилии и инициалов.

Заключение по результатам аттестации подписывается членами аттестационной комиссии, утверждается руководителем органа аттестации и представляется заявителю [2].

Заключение и протоколы испытаний подлежат утверждению органом по аттестации.

- оформление, регистрация и выдача "Аттестата соответствия" (если заключение по результатам аттестации утверждено).
- осуществление государственного контроля и надзора, инспекционного контроля за проведением аттестации и эксплуатацией аттестованных объектов информатизации;
- рассмотрение апелляций. В случае, если заявитель не согласен с отказом в выдаче "Аттестата соответствия", он может подать апелляцию в вышестоящий орган по аттестации или в ФСТЭК. Апелляция рассматривается в срок, не превышающий один месяц с привлечением заинтересованных сторон.

Аттестат соответствия должен содержать:

- регистрационный номер;
- дату выдачи;
- срок действия;
- наименование, адрес и местоположение объекта информатизации;
- категорию объекта информатизации;
- класс защищенности автоматизированной системы;
- гриф секретности (конфиденциальности) информации, обрабатываемой на объекте информатизации;
- организационную структуру объекта информатизации и вывод об уровне подготовки специалистов по защите информации;
- номера и даты утверждения программы и методики, в соответствии с которыми проводились аттестационные испытания;
- перечень руководящих документов, в соответствии с которыми проводилась аттестация;
- номер и дата утверждения заключения по результатам аттестационных испытаний;
- состав комплекса технических средств обработки информации ограниченного доступа, перечень вспомогательных технических средств и систем, перечень технических средств защиты информации, а также схемы их размещения в помещениях и относительно границ контролируемой зоны, перечень используемых программных средств;
- организационные мероприятия, при проведении которых разрешается обработка информации ограниченного доступа;
- перечень действий, которые запрещаются при эксплуатации объекта информатизации;

– список лиц, на которых возлагается обеспечение требований по защите информации и контроль за эффективностью реализованных мер и средств защиты информации.

Аттестат соответствия подписывается руководителем аттестационной комиссии и утверждается руководителем органа по аттестации. Аттестат соответствия выдается на период, в течение которого обеспечивается неизменность условий функционирования объекта информатизации и технологии обработки защищаемой информации, могущих повлиять на характеристики, определяющие безопасность информации (состав и структура технических средств, условия размещения, используемое программное обеспечение, режимы обработки информации, средства и меры защиты), но не более чем на 3 года.

Практические задания

1. Разработать протокол аттестационных испытаний для системы ведение секретных переговоров;
2. Разработать аттестат соответствия для системы защиты телекоммуникационных систем;
3. Соберите информацию и составьте справку об испытательных центах (лабораториях) по сертификации продукции по требованиям безопасности информации;
4. Подберите состав комплекса технических средств обработки информации ограниченного доступа, перечень вспомогательных технических средств и систем, перечень технических средств защиты информации, а также схемы их размещения в помещениях и относительно границ контролируемой зоны, перечень используемых программных средств для информационной системы.

Контрольные вопросы

1. Дайте определение объекта информатизации?
2. В каких случаях аттестация является обязательной?
3. Назовите перечень работ при аттестации?
4. Что входит в структуру системы аттестации?
5. Функции ФСТЭК в рамках системы аттестации?
6. Где представлен список органов по аттестации?
7. Какие документы и данные заявитель должен предоставить органу по аттестации?
8. Что должен включать протокол аттестационных испытаний?
9. Что должен содержать аттестат соответствия?

Практическое занятие № 7

Составление трудового договора сотрудника службы информационной безопасности

Цель занятия: Закрепить знания об основах трудового права, регулировании вопросов обеспечения сохранности информации с ограниченным доступом и правовом регулировании взаимоотношений администрации и персонала в области защиты информации.

Краткие теоретические сведения

Источники трудового права

Источники трудового права представляют собой акты, содержащие правовые нормы, посредством которых регулируются трудовые отношения. Источники трудового права подразделяются на федеральные и локальные.

К федеральным источникам относятся:

- Конституция Российской Федерации, которая является юридической базой трудового законодательства;
- Федеральные законы, содержащие нормы трудового права;
- Трудовой кодекс Российской Федерации от 30 декабря 2001 г. (ФЗ РФ №197-ФЗ);
- нормативные указы Президента РФ, направленные на регуляцию трудовых отношений;
- постановления Правительства РФ, регулирующие отношения, составляющие предмет трудового права;
- нормативные акты министерств, ведомств и комитетов Российской Федерации;
- постановления пленума Верховного Суда Российской Федерации по спорным вопросам трудовых отношений
- К локальным источникам трудового права относятся:
- нормативно-правовые акты субъектов РФ;
- правотворчество органов местного самоуправления;
- правила внутреннего трудового распорядка, установленные на предприятии;
- коллективные договоры и соглашения;
- трудовые договоры;
- приказы и распоряжения руководителей предприятий и учреждений.

Трудовой договор (контракт) как особый вид договорных отношений. Соглашение двух или нескольких лиц об устранении, изменении или прекращении гражданских прав и обязанностей признается договором (см. п. 1 ст. 420 ГК РФ).

Под термином "договор" понимают также гражданское правоотношение, возникшее из договора, и собственно документ, в котором изложено содержание договора, заключенного в письменной форме.

По гражданскому законодательству, участниками договора могут быть граждане, юридические лица, Российская Федерация, субъекты Российской Федерации и муниципальные образования (см. ст. 124 ГК РФ).

Путем заключения трудового договора трудящиеся реализуют не только свой принцип свободы труда в соответствии со ст. 37 Конституции РФ, но и право на выбор профессии, занятий, а также выбор места работы.

По трудовому договору работник обязуется выполнять работу по определенной специальности, квалификации или должности с подчинением внутреннему трудовому распорядку, а работодатель (физическое или юридическое лицо) обязуется выплачивать трудящемуся заработную плату и обеспечивать условия труда, предусмотренные законодательством о труде, коллективным договором и соглашением сторон (см. ст. 15 КЗоТ РФ).

Важнейшая черта института трудового договора, пронизывающая как нормы права о приеме, так и нормы о переводах и увольнении, - это свобода трудового договора, отражающая принцип свободы труда в обществе, закрепленный в ст. 37 Конституции РФ. Свобода трудового

договора означает, что граждане: а) свободно по своему желанию выбирают место и род трудовой деятельности, работы; б) свободно, добровольно решают вопрос о работе, заключают трудовой договор и могут его расторгнуть в любое время в установленном порядке; в) имеют, как правило, устойчивые трудовые договоры. Трудовой договор отражает договорный принцип привлечения к труду в качестве работника. Без трудового договора не может работать ни один работник, кем бы, где бы и сколько бы он ни работал в качестве такового.

Трудовой договор может быть расторгнут: по соглашению сторон, по инициативе работника, по инициативе работодателя, по инициативе других компетентных органов, в других предусмотренных законом случаях.

Предмет и стороны контракта. Из определения трудового договора, данного законодателем в ст. 15 КЗоТ РФ (в редакции Федерального закона от 6 мая 1998 года № 69-ФЗ) как соглашение между работником и работодателем (физическим или юридическим лицом), видимо, что это двустороннее соглашение. Одной его стороной является гражданин, заключивший договор в качестве работника, а другой – работодатель (физическое или юридическое лицо независимо от организационно – правовой формы собственности), принявший его на работу.

Но чтобы трудовое отношение возникло и соответствовало требованиям закона, каждая из сторон должна обладать трудовой праводееспособностью (правосубъективностью), то есть способствовать своими действиями осуществить субъективные права и обязанности.

Работником в качестве субъекта трудового договора может быть любой гражданин, достигший определенного возраста (15 - 16 лет) и обладающий реальной способностью трудиться.

Предметом трудового права являются трудовые отношения, возникающие при применении работником своей способности к труду в процессе трудовой деятельности, а также другие общественные отношения, связанные с трудовыми.

К ним относятся:

- отношения между работником и работодателем;
- организационно-управленческие отношения профсоюзного органа, представляющего интересы трудового коллектива, с администрацией предприятия по поводу улучшения условий труда, быта и отдыха работников;
- отношения по перераспределению рабочей силы;
- отношения по поводу занятости и трудоустройства;
- отношения, связанные с возмещением материального ущерба;
- процессуальные отношения, возникающие при разрешении трудовых споров

Регулирование взаимоотношений администрации и персонала в области защиты информации. На любом предприятии, как правило, весь рабочий персонал условно может быть разделен на три группы:

1. Руководители предприятия:

А) Руководители высшего звена (владелец предприятия, учредители, президент, начальник службы безопасности)

Б) Руководители среднего звена (начальники отделов и служб, менеджеры и т.п.)

2. Специалисты (наиболее многочисленная группа, сюда входят работники, обладающие специальными знаниями и непосредственно выполняющие основной массив работы Вашего предприятия)

3. Обслуживающий и технический персонал (сюда относятся грузчики, разнорабочие, уборщицы и т. п.)

Очевидно, что все эти группы должны обладать разными объемами информации. Руководители предприятия в целом, должны иметь доступ к любой информации, касающейся деятельности предприятия; руководители подразделений - информация, касающаяся работы вверенного подразделения; специалисты должны иметь доступ к такому объему информации, который необходим им для профессионального и добросовестного выполнения ими своих профессиональных обязанностей; обслуживающий и технический персонал должен иметь минимум доступа к конфиденциальной информации. При допуске рабочего персонала к

конфиденциальной информации необходимо помнить правило: «Чем меньше человек о чем-то знает, тем меньше он может об этом рассказать».

Важной составляющей правового регулирования в области информационных технологий является установление ответственности граждан за противоправные действия при работе с конфиденциальными информационными ресурсами.

Работники должны инструктироваться о необходимости держать в секрете все, что они узнают на работе. Причем инструктаж проводится как при приеме на работу, так и в ходе выполнения служебных обязанностей. Ну и, конечно, обязательно напомнят увольняющемуся работнику о том, что он обязан молчать обо всем, что он знает. И такие действия не ограничиваются устными напоминаниями. В обязательном порядке все работники подписывают обязательство о сохранении конфиденциальной информации.

Информация, предназначенная для руководства ни в коем случае не должна попасть к среднему звену и уж, тем более, не должна быть известна обслуживающему персоналу. Информация должна быть четко разделена на уровни и каждый имеет доступ в соответствии со своей должностью

Должностные лица и граждане, виновные в нарушении законодательства Российской Федерации о конфиденциальной информации, несут уголовную, административную, гражданско-правовую или дисциплинарную ответственность в соответствии с действующим законодательством.

Взаимные обязательства администрации и оформляемого лица отражаются в трудовом договоре (контракте), а также в дополнительных документах, таких как приложения, подписки и др.

Практические задания

1. Разработать приложение к заявлению об увольнении для лица, работавшего с документами, составляющими коммерческую тайну предприятия.
2. Разработать памятку работнику о сохранении коммерческой тайны предприятия. Памятка должна содержать следующие пункты: ссылку на перечень сведений, составляющих коммерческую тайну предприятия; об утрате или недостаче документов; при увольнении, уходе в отпуск; о помещении сведений в государственные секреты; порядок предоставления информации службе безопасности и другие.
3. Разработать документ (памятку), регулирующий обязанности лиц, допущенных к сведениям, составляющих коммерческую тайну предприятия.
4. Разработать подписку работника о сохранении коммерческой тайны предприятия.
5. Разработать текст трудового договора (контракта) для конкретного предприятия на основе существующего законодательства РФ.
6. Разработать текст заявления для суда на лицо, нарушившее подписку о неразглашении коммерческой тайны для привлечения к административной ответственности (выплата компенсации за материальный ущерб).

Контрольные вопросы

1. Перечислите источники трудового права
2. Что такое трудовой контракт (договор)?
3. Расскажите о предмете и сторонах трудового контракта (договора).
4. Как должны инструктироваться работники о необходимости сохранять в тайне конфиденциальную информацию предприятия?
5. Какую ответственность несут должностные лица и граждане, виновные в нарушении законодательства Российской Федерации о конфиденциальной информации?
6. Как условно можно разделить персонал на предприятии и как должны регулироваться информационные потоки между условными группами?
7. Перечислите основные документы, регулирующие взаимоотношения администрации и персонала в области защиты информации.