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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению практических и самостоятельных работ

по дисциплине «Элементы высшей математики»

по специальностям СПО:

09.02.01 Компьютерные системы и комплексы

09.02.03 Программирование в компьютерных системах

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий естественнонаучных дисциплин
Протокол от «15» 01 2020 г. № 6
Председатель цикловой комиссии Е.А.Рейм

Раздел 1 Элементы линейной алгебры

ТЕМА «***МАТРИЦЫ И ОПРЕДЕЛИТЕЛИ***»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

I. Что необходимо знать по теме:

1. Матрицы: а) определение; б) обозначение; в) элементы матрицы; г) виды матрицы; д) обратная матрица и ее свойства; е) элементарные преобразования матрицы; ж) действия над матрицами и их свойства; з) ранг матрицы; и) схемы исследования систем линейных уравнений с использованием ранга матрицы.
2. Определители: а) определение; б) свойства; в) минор и алгебраическое дополнение элемента.

II. Что необходимо знать по теме:

1. Вычислять определители:

   а) второго порядка: $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} * a_{22} - a_{12} * a_{21}$

   б) третьего порядка:

   I способ: $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} * a_{22} * a_{33} + a_{12} * a_{23} * a_{31} + a_{21} * a_{32} * a_{13} - a_{13} * a_{22} * a_{31} - a_{12} * a_{21} * a_{33} - a_{23} * a_{32} * a_{11}$

   II способ: $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} * (a_{22} * a_{33} - a_{23} * a_{32}) - a_{12} * (a_{21} * a_{33} - a_{23} * a_{31}) + a_{13} * (a_{21} * a_{32} - a_{22} * a_{31})$

   в) определители n-го порядков с помощью свойств 7 и 8 приводить к третьему порядку, а затем вычислять.
2. Вычислять обратную матрицу по схеме:

   Дана матрица А. Найти A^{-1}.

   а) $|A| \neq 0 \Longrightarrow$ вычислить определитель матрицы

   б) $A^{*} \Longrightarrow$ транспонировать матрицу

   в) $\widetilde{A} = \begin{vmatrix} A'_{11} & A'_{12} \cdots & A_{1n} \\ \vdots & \vdots \quad \cdots & \vdots \\ A_{m1} & A'_{m2} \cdots & A_{mn} \end{vmatrix} \Longrightarrow$ Составить матрицу представляющую собой алгебраические дополнения каждого элемента транспонированной матрицы.

   г) $A^{-1} = \begin{vmatrix} \frac{A_{11}}{|A|} & \frac{A'_{12}}{|A|} & \cdots & \frac{A_{1n}}{|A|} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{A_{m1}}{|A|} & \frac{A_{m2}}{|A|} & \cdots & \frac{A_{mn}}{|A|} \end{vmatrix} \Longrightarrow$ Чтобы получить A^{-1} (обратную матрицу), необходимо матрицу $\widetilde{A}$ (Астильдой) поделить на определитель матрицы А.
3. Умножить матрицы,

   помнить, что:

   а) EA=AE=A, если А квадратная

   б) $A*A^{-1}=A^{-1}*A=E$

   в) $A*B \neq B*A$
4. Уметь вычислять R (ранг) матрицы
5. Уметь решать систему:

   а) по правилу Крамера (с помощью определителей)

   б) методом Гаусса (лестничный метод)

   в) матричным методом (с помощью обратной матрицы)

6. Исследовать систему с помощью ранга матрицы

III. Примеры решения и исследования систем:

1. Решить систему по правилу Крамера

$$\begin{cases} 2x - y + 2z = -3 \\ x + 2y - z = 4 \\ 3x + y + 3z = 3 \end{cases}$$

$$\Delta = \begin{vmatrix} 2 & -1 & 2 \\ 1 & 2 & -1 \\ 3 & 1 & 3 \end{vmatrix} = 2 * 2 * 3 + (-1) * (-1) * 3 + 1 * 1 * 2 - 2 * 2 * 3 - (-1) * 1 * 3 - (-1) * 1 * 2 = 12 + 3 + 2 - 12 + 3 + 2 = 10$$

$$\Delta_x = \begin{vmatrix} -3 & -1 & 2 \\ 4 & 2 & -1 \\ 3 & 1 & 3 \end{vmatrix} = -10; \quad \Delta_y = \begin{vmatrix} 2 & -3 & 2 \\ 1 & 4 & -1 \\ 3 & 3 & 3 \end{vmatrix} = 30; \quad \Delta_z = \begin{vmatrix} 2 & -1 & -3 \\ 1 & 2 & 4 \\ 3 & 1 & 3 \end{vmatrix} = 10;$$

$$x = \frac{\Delta_x}{\Delta} = \frac{-10}{10} = -1; \qquad y = \frac{\Delta_y}{\Delta} = \frac{30}{10} = 3; \qquad z = \frac{\Delta_z}{\Delta} = \frac{10}{10} = 1;$$

Ответ: (-1; 3; 1)

2. Решить систему по методом Гаусса

$$\begin{cases} 2x - y + 2z = -3 \\ x + 2y - z = 4 \\ 3x + y + 3z = 3 \end{cases} \begin{matrix} | \\ | \\ | \end{matrix} \begin{matrix} 1 \\ -2 \\ 1 \end{matrix} \begin{matrix} | \\ | \\ | \end{matrix} \begin{matrix} -3 \\ \\ \end{matrix}$$

$$\begin{array}{r} 2x-y+2z=-3 \\ -x-4y+2x=-8 \\ \hline -5y+4z=-11 \end{array} \qquad \begin{array}{r} -3x-6y+3z=-12 \\ 3x+y+3z=3 \\ \hline -5y+6z=-9 \end{array}$$

$$\begin{cases} x + 2y - z = 4 \\ -5y + 4z = -11 \\ -5y + 6z = -9 \end{cases} \begin{matrix} \\ | -1 \\ | 1 \end{matrix} \qquad \begin{array}{r} 5y-4z=11 \\ -5y+6z=-9 \\ \hline 2z=2(:2) \\ z=1 \end{array} \qquad \begin{cases} x + 2y - z = 4 \\ -5y + 4z = -11 \\ z + 1 \end{cases}$$

$z = 1 \Rightarrow -5y + 4 * 1 = -11 \qquad y = 3 \Rightarrow x + 2 * 3 - 1 = 4 \qquad x = -1$

Ответ: (-1; 3; 1)

3. Решить систему по методом матриц

$$\begin{cases} 2x - y + 2z = -3 \\ x + 2y - z = 4 \\ 3x + y + 3z = 3 \end{cases} \qquad \underset{A}{\begin{pmatrix} 2 & -1 & 2 \\ 1 & 2 & -1 \\ 3 & 1 & 3 \end{pmatrix}} * \underset{x}{\begin{pmatrix} x \\ y \\ z \end{pmatrix}} = \underset{B}{\begin{pmatrix} -3 \\ 4 \\ 3 \end{pmatrix}}$$

1) $A = \begin{pmatrix} 2 & -1 & 2 \\ 1 & 2 & -1 \\ 3 & 1 & 3 \end{pmatrix}$ найдем A^{-1} по схеме $A' = \begin{pmatrix} 2 & 1 & 3 \\ -1 & 2 & 1 \\ 2 & -1 & 3 \end{pmatrix}$

$$|A| = \begin{vmatrix} 2 & -1 & 2 \\ 1 & 2 & -1 \\ 3 & 1 & 3 \end{vmatrix} = 10$$

$$\bar{A} = \begin{pmatrix} + \begin{vmatrix} -1 & 3 \end{vmatrix} & - \begin{vmatrix} 2 & 3 \end{vmatrix} & + \begin{vmatrix} 2 & -1 \end{vmatrix} \\ - \begin{vmatrix} 1 & 3 \\ -1 & 3 \end{vmatrix} & + \begin{vmatrix} 2 & 3 \\ 2 & 3 \end{vmatrix} & - \begin{vmatrix} 2 & 1 \\ 2 & -1 \end{vmatrix} \\ + \begin{vmatrix} 1 & 3 \\ 2 & 1 \end{vmatrix} & - \begin{vmatrix} 2 & 3 \\ -1 & 1 \end{vmatrix} & + \begin{vmatrix} 2 & 1 \\ -1 & 2 \end{vmatrix} \end{pmatrix} = \begin{pmatrix} 7 & 5 & -3 \\ -6 & 0 & 4 \\ -5 & -5 & 5 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} {}/_{10} & {}/_{10} & {}/_{10} \\ -6/_{10} & 0 & 4/_{10} \\ -5/_{10} & -5/_{10} & 5/_{10} \end{pmatrix}$$

$$2)\ A^{-1} * B = \begin{pmatrix} 7/10 & 5/10 & -3/10 \\ -6/10 & 0 & 4/10 \\ -5/10 & -5/10 & 5/10 \end{pmatrix} * \begin{pmatrix} -3 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} \frac{7*(-3)+5*4+(-3)*3}{10} \\ \frac{-6*(-3)+0*4+4*3}{10} \\ \frac{-5*(-3)+(-5)*4+5*3}{10} \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Ответ: (-1; 3; 1)

4. Исследовать систему по рангу

m – число уравнений

n – число переменных

$$\begin{cases} x + 2y + z = 4 \\ 3x - 5y + 3z = 1 \\ 2x + 7y - z = 8 \\ 2x - 7y + 2z = -3 \end{cases} \quad m=4;\ n=3$$

1. Вычислим ранг A и B

$$\begin{matrix} -2 \\ \\ \\ \Delta \end{matrix} \left(\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 3 & -5 & 3 & 1 \\ 2 & 7 & -1 & 8 \\ 2 & -7 & 2 & -3 \end{array}\right) \begin{matrix} -3 \\ \\ \\ -2 \end{matrix} \sim \left(\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & -11 & 0 & -11 \\ 0 & 3 & -3 & 0 \\ 0 & -11 & 0 & -11 \end{array}\right) \begin{matrix} \\ :(-11) \\ :(3) \\ :(-11) \end{matrix} \sim \left(\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{array}\right) -1$$

A B

$$\sim \left(\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right) -1 \sim \left(\begin{array}{ccc|c} 1 & 2 & 1 & 4 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right) R_A=3 \quad R_B=3$$

A B

2. Делаем вывод для системы:

1) $R_A=R_B=3$ – система совместна

2) R=3=n – система имеет единственное решение

3) m=4 R=3 m-R=1 – одно уравнение, причем четвертое которое превратилась в нули зависисмое, его вычеркиваем

4) Найдем решение систем, записав ее в преобразованном виде из последней формы, где вычисляли ранг

$$\begin{cases} x + 2y + z = 4 \\ y - z = 0 \\ z = 1 \end{cases} \quad 1)\ z = 1 \Rightarrow y - 1 = 0, y = 1 \Rightarrow x + 2 * 1 + 1 = 4, x = 1$$

Ответ: (1; 1; 1)

5. Вычислим определитель:

$$\begin{vmatrix} 1 & 3 & 2 & 1 \\ 4 & -2 & 3 & -2 \\ 3 & 2 & 1 & 2 \\ -1 & 3 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 2 & 0 & 2 & 1 \\ 2 & 4 & 3 & -2 \\ 5 & -4 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{vmatrix} = +1 * \begin{vmatrix} 2 & 0 & 2 \\ 2 & 4 & 3 \\ 5 & -4 & 1 \end{vmatrix} = -24$$

6. Перемножить матрицы.

$$\begin{pmatrix}1 & 2 & 1\\3 & 1 & 0\\4 & 5 & 7\end{pmatrix}_{3x3} * \begin{pmatrix}1 & 2\\3 & 4\\5 & 1\end{pmatrix}_{3x2} = \begin{pmatrix}1*1+2*3+1*5 & 1*2+2*4+1*1\\3*1+3*3+0*5 & 3*2+1*4+0*1\\4*1+5*3+7*5 & 1*2+4*4+7*1\end{pmatrix}_{3x2}$$
$$= \begin{pmatrix}12 & 11\\6 & 10\\54 & 50\end{pmatrix}_{3x2}$$

ПРАКТИЧЕСКАЯ РАБОТА №1

ВЫПОЛНЕНИЕ ОПЕРАЦИЙ НАД МАТРИЦАМИ.

Вариант 1

1) $A = \begin{pmatrix}4 & 5 & -2\\3 & -1 & 0\\4 & 2 & 7\end{pmatrix}; B = \begin{pmatrix}2 & 1 & -1\\0 & 1 & 3\\5 & 7 & 3\end{pmatrix}$

Найти: 3A – (A + 2B)

2) Найти произведение матриц:

$$A = \begin{pmatrix}1 & 0 & 2\\0 & -1 & 3\\1 & 4 & 0\end{pmatrix}; B = \begin{pmatrix}2 & 1 & 1\\1 & 4 & 1\\3 & 2 & 0\end{pmatrix}$$

Вариант 2

1) $A = \begin{pmatrix}4 & 5 & 6\\-1 & 0 & 3\\-1 & 2 & -1\end{pmatrix}; B = \begin{pmatrix}0 & -1 & 2\\1 & 0 & -2\\3 & 1 & 2\end{pmatrix}$

Найти: (3B - A) + 2A

2) Найти произведение матриц:

$$A = \begin{pmatrix}2 & 3 & 1\\4 & 1 & -1\\5 & 2 & 0\end{pmatrix}; B = \begin{pmatrix}1 & 0 & 2\\0 & 3 & 0\\2 & 1 & 0\end{pmatrix}$$

ПРАКТИЧЕСКАЯ РАБОТА №2

ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЕЙ.

Вариант 1

Вычислить определитель матрицы:

$$A = \begin{pmatrix}2 & 1 & -4\\-1 & 0 & 1\\0 & 1 & 3\end{pmatrix}$$

Вариант 2

Вычислить определитель матрицы:

$$A = \begin{pmatrix}-4 & 1 & -1\\14 & -2 & 2\\7 & 2 & 1\end{pmatrix}$$

ПРАКТИЧЕСКАЯ РАБОТА №3

ОБРАТНАЯ МАТРИЦА. ВЫЧИСЛЕНИЕ ОБРАТНОЙ МАТРИЦЫ. НАХОЖДЕНИЕ РАНГА МАТРИЦЫ. РЕШЕНИЕ МАТРИЧНЫХ УРАВНЕНИЙ.

Вариант 1

1)Для заданной матрицы вычислить обратную:

$$A = \begin{pmatrix} 2 & 0 & 4 \\ -1 & 1 & -2 \\ 0 & 3 & 1 \end{pmatrix}$$

2) Найти ранг матрицы

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 3 & -5 & 3 \\ 2 & 7 & -1 \\ 2 & -7 & 2 \end{pmatrix}$$

3) Решить матричное уравнение: A * X = B

$$A = \begin{pmatrix} 1 & 1 & -1 \\ -2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}; B = \begin{pmatrix} 2 \\ 3 \\ 6 \end{pmatrix}$$

Вариант 2

1)Для заданной матрицы вычислить обратную:

$$A = \begin{pmatrix} 2 & 1 & -1 \\ -1 & -2 & 2 \\ 4 & 2 & 1 \end{pmatrix}$$

2) Найти ранг матрицы

$$A = \begin{pmatrix} 2 & 3 & -5 \\ 6 & 8 & -14 \\ 2 & -2 & -1 \\ 5 & 11 & -16 \end{pmatrix}$$

3) Решить матричное уравнение: A * X = B

$$A = \begin{pmatrix} 2 & 3 & 2 \\ 1 & 2 & -3 \\ 3 & 4 & 1 \end{pmatrix}; B = \begin{pmatrix} 9 \\ 14 \\ 6 \end{pmatrix}$$

ТЕМА «*СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ*»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Система линейных уравнений — это объединение из n линейных уравнений, каждое из которых содержит k переменных. Записывается это так:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2k}x_k = b_2 \\ \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nk}x_k = b_n \end{cases}$$

Многие, впервые сталкиваясь с высшей алгеброй, ошибочно полагают, что число уравнений обязательно должно совпадать с числом переменных. В школьной алгебре так обычно и бывает, однако для высшей алгебры это, вообще говоря, неверно.

Решение системы уравнений — это последовательность чисел $(k_1, k_2, \ldots, k_n)$, которая является решением каждого уравнения системы, т.е. при подстановке в это уравнение вместо переменных $x_1, x_2, \ldots, x_n$ дает верное числовое равенство.

Соответственно, решить систему уравнений — значит найти множество всех ее решений или доказать, что это множество пусто. Поскольку число уравнений и число неизвестных может не совпадать, возможны три случая:

1. Система несовместна, т.е. множество всех решений пусто. Достаточно редкий случай, который легко обнаруживается независимо от того, каким методом решать систему.

2. Система совместна и определена, т.е. имеет ровно одно решение. Классический вариант, хорошо известный еще со школьной скамьи.
3. Система совместна и не определена, т.е. имеет бесконечно много решений. Это самый жесткий вариант. Недостаточно указать, что «система имеет бесконечное множество решений» — надо описать, как устроено это множество.

Переменная x_i называется *разрешенной*, если она входит только в одно уравнение системы, причем с коэффициентом 1. Другими словами, в остальных уравнениях коэффициент при переменной x_i должен быть равен нулю.

Если в каждом уравнении выбрать по одной разрешенной переменной, получим набор разрешенных переменных для всей системы уравнений. Сама система, записанная в таком виде, тоже будет называться разрешенной. Вообще говоря, одну и ту же исходную систему можно свести к разным разрешенным, однако сейчас нас это не волнует. Вот примеры разрешенных систем:

$$1)\ \begin{cases} x_1 + 7x_2 + x_5 = 3 \\ x_3 = -4 \\ x_4 - x_6 = 11 \end{cases} \qquad 2)\ \begin{cases} x_1 = 7 - 2x_2 \\ x_3 = 3 + x_2 + 6x_5 \\ x_4 = x_5 \end{cases}$$

Обе системы являются разрешенными относительно переменных x_1, x_3 и x_4. Впрочем, с тем же успехом можно утверждать, что вторая система — разрешенная относительно x_1, x_3 и x_5. Достаточно переписать самое последнее уравнение в виде $x_5 = x_4$.

Теперь рассмотрим более общий случай. Пусть всего у нас k переменных, из которых r являются разрешенными. Тогда возможны два случая:

1. Число разрешенных переменных r равно общему числу переменных k: $r = k$. Получаем систему из k уравнений, в которых $r = k$ разрешенных переменных. Такая система является совместной и определенной, т.к. $x_1 = b_1, x_2 = b_2, \ldots, x_k = b_k$;
2. Число разрешенных переменных r меньше общего числа переменных k: $r < k$. Остальные $(k - r)$ переменных называются свободными — они могут принимать любые значения, из которых легко вычисляются разрешенные переменные.

Так, в приведенных выше системах переменные x_2, x_5, x_6 (для первой системы) и x_2, x_5 (для второй) являются свободными. Случай, когда есть свободные переменные, лучше сформулировать в виде теоремы:

Обратите внимание: это очень важный момент! В зависимости от того, как вы запишете итоговую систему, одна и та же переменная может быть как разрешенной, так и свободной. Большинство репетиторов по высшей математике рекомендуют выписывать переменные в лексикографическом порядке, т.е. по возрастанию индекса. Однако вы совершенно не обязаны следовать этому совету.

Теорема. Если в системе из n уравнений переменные $x_1, x_2, \ldots, x_r$ — разрешенные, а $x_{r+1}, x_{r+2}, \ldots, x_k$ — свободные, то:

1. Если задать значения свободным переменным ($x_{r+1} = t_{r+1}, x_{r+2} = t_{r+2}, \ldots, x_k = t_k$), а затем найти значения $x_1, x_2, \ldots, x_r$, получим одно из решений.
2. Если в двух решениях значения свободных переменных совпадают, то значения разрешенных переменных тоже совпадают, т.е. решения равны.

В чем смысл этой теоремы? Чтобы получить все решения разрешенной системы уравнений, достаточно выделить свободные переменные. Затем, присваивая свободным переменным разные значения, будем получать готовые решения. Вот и все — таким образом можно получить все решения системы. Других решений не существует.

Вывод: разрешенная система уравнений всегда совместна. Если число уравнений в разрешенной системе равно числу переменных, система будет определенной, если меньше — неопределенной.

И все бы хорошо, но возникает вопрос: как из исходной системы уравнений получить разрешенную? Для этого существует метод Гаусса.

ПРАКТИЧЕСКАЯ РАБОТА №4

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДОМ ОБРАТНОЙ МАТРИЦЫ, ПО ФОРМУЛАМ КРАМЕРА.

Вариант 1

1. $\begin{cases} 3x+2y+z=14 \\ 2x+y+4z=12 \\ x+3y+2z=11 \end{cases}$; 2. $\begin{cases} x+2y+3z=0 \\ 2x+5y+6z=1 \\ 2x+7y+7z=2 \end{cases}$; 3. $\begin{cases} x-2y+3z=3 \\ 3x+y-6z=-7 \\ 9x-2y-z=3 \end{cases}$

4. $\begin{cases} 2x+y-z=1 \\ x+y-z=-2 \\ 4x-3y+z=1 \end{cases}$;

Вариант 2

5. $\begin{cases} 2x-3y+z=5 \\ x+y-z=1 \\ 3x-5z=1 \end{cases}$; 6. $\begin{cases} x-2y-7z=-3 \\ 10x+y+4z=1 \\ 2x+y+5z=0 \end{cases}$; 7. $\begin{cases} x+2y-z=4 \\ 3x-y+z=4 \\ 2x+y+2z=16 \end{cases}$

8. $\begin{cases} x-2y+4z=6 \\ 2x-y+3z=11 \\ 4x+y-5z=9 \end{cases}$

ПРАКТИЧЕСКАЯ РАБОТА №5

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДОМ ГАУССА.

Вариант 1

9. $\begin{cases} 4x-y-5z=1 \\ x+y-2z=6 \\ 3x-2y-6z=-2 \end{cases}$; 10. $\begin{cases} 2x+y-z=1 \\ x+y+z=6 \\ 3x-y+z=4 \end{cases}$; 11. $\begin{cases} 3x-2y+z=-3 \\ 5x+y-2z=11 \\ x+y+z=1 \end{cases}$

Вариант 2

12. $\begin{cases} 3x+4y+2z=5 \\ 5x-6y-4z=-3 \\ -4x+5y+3z=31 \end{cases}$; 13. $\begin{cases} 5x-3y+2z=19 \\ 4x+5y-3z=31 \\ 3x+7y-4z=31 \end{cases}$; 14. $\begin{cases} x-2y+3z=3 \\ 3x+y-6z=-7 \\ 9x-2y-z=3 \end{cases}$;

ПРАКТИЧЕСКАЯ РАБОТА №6

РЕШЕНИЕ СИСТЕМ М ЛИНЕЙНЫХ УРАВНЕНИЙ С N – ПЕРЕМЕННЫМИ.

Вариант 1

16. $\begin{cases} 2x-y-3z+2t=12 \\ x+y-4z-t=7 \\ 3x+5y-2z+4t=15 \\ 3x-2y-3z+2t=13 \end{cases}$; 17. $\begin{cases} x+y-3z+2t=6 \\ x-2y-t=-6 \\ y+z+3t=16 \\ 2x-3y+2z=6 \end{cases}$; 18. $\begin{cases} 2x+3y+5z+4t=13 \\ x-2y+3z+3t=13 \\ 3x+2y-5t=-11 \\ 3y-2z+5t=12 \end{cases}$;

19. $\begin{cases} 2x-y+3z-2t=1 \\ x+y-z+t=4 \\ x-z+2t=6 \\ 3x-y+z-t=0 \end{cases}$

Вариант 2

20. $\begin{cases} 2x+y-4z+3t=-4 \\ x-4y+3z-2t=-1 \\ 3x+2y-2z+t=3 \\ 2x+4y-2z-3t=6 \end{cases}$; 21. $\begin{cases} 3x+4y-z-t=3 \\ 2x-y-2z+2t=9 \\ x+3y+5z-4t=2 \\ 4x-8y-3z+3t=10 \end{cases}$; 22. $\begin{cases} 2x+y+z+t=5 \\ x+2y+z+t=4 \\ x+y+2z+t=7 \\ x+y+z+2t=4 \end{cases}$;

23. $\begin{cases} x+y+2z+3t=1 \\ 3x-y-z-2t=-4 \\ 2x+3y-z-t=-6 \\ x+2y+3z-t=-4 \end{cases}$

Контрольная работа №1

Элементы линейной алгебры.

Вариант 1

1. Вычислить определитель:

$$\begin{vmatrix} 4 & 2 & 2 \\ 4 & 4 & 4 \\ 7 & 8 & 10 \end{vmatrix}$$

2. Решить систему уравнений методом Крамера:

$$\begin{cases} 2x_1+x_2+x_3=8 \\ 3x_1-2x_2-3x_3=-5 \\ -4x_1+3x_2+5x_3=10 \end{cases}$$

Вариант 2

1. Вычислить определитель:

$$\begin{vmatrix} 6 & 0 & 3 \\ 4 & 2 & 1 \\ 0 & 4 & 2 \end{vmatrix}$$

2. Решить систему уравнений методом Крамера:

$$\begin{cases} -3x_1+x_2+5x_3=4 \\ 6x_1-x_2-x_3=0 \\ 1{,}5x_1-2x_2+4x_3=-14 \end{cases}$$

Раздел 2. Элементы аналитической геометрии.

ТЕМА *«ВЕКТОРЫ И КООРДИНАТЫ НА ПЛОСКОСТИ»*

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Определение Упорядоченную совокупность $(x_1, x_2, \ldots, x_n)$ n вещественных чисел называют *n-мерным вектором*, а числа x_i $(i = 1,\ldots, n)$ - *компонентами*, или *координатами*, вектора.

Пример. Если, например, некоторый автомобильный завод должен выпустить в смену 50 легковых автомобилей, 100 грузовых, 10 автобусов, 50 комплектов запчастей для легковых автомобилей и 150 комплектов для грузовых автомобилей и автобусов, то производственную программу этого завода можно записать в виде вектора (50, 100, 10, 50, 150), имеющего пять компонент.

Обозначения. Векторы обозначают жирными строчными буквами или буквами с чертой или стрелкой наверху, например, $\bar{a}$ или $\vec{a}$. Два вектора называются *равными*, если они имеют одинаковое число компонент и их соответствующие компоненты равны.

Компоненты вектора нельзя менять местами, например, (3, 2, 5, 0, 1) и (2, 3, 5, 0, 1) разные вектора.

Операции над векторами. *Произведением* вектора $\mathbf{x} = (x_1, x_2, ..., xn)$ на действительное число λ называется вектор $\lambda\,\mathbf{x} = (\lambda\,x_1, \lambda\,x_2, ..., \lambda\,x_n)$.

Суммой векторов $\mathbf{x} = (x_1, x_2, ..., xn)$ и $\mathbf{y} = (y_1, y_2, ..., yn)$ называется вектор $\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, ..., x_n + + y_n)$.

Пространство векторов. *N-мерное векторное пространство* $\mathbf{R}^n$определяется как множество всех n-мерных векторов, для которых определены операции умножения на действительные числа и сложение.

Экономическая иллюстрация. Экономическая иллюстрация n-мерного векторного пространства: *пространство благ* (*товаров*). Под *товаром* мы будем понимать некоторое благо или услугу, поступившие в продажу в определенное время в определенном месте. Предположим, что существует конечное число наличных товаров n; количества каждого из них, приобретенные потребителем, характеризуются набором товаров

1. $\mathbf{x} = (x_1, x_2, ..., x_n)$,

где через x_i обозначается количество i-го блага, приобретенного потребителем. Будем считать, что все товары обладают свойством произвольной делимости, так что может быть куплено любое неотрицательное количество каждого из них. Тогда все возможные наборы товаров являются векторами пространства товаров $C = \{x = (x_1, x_2, ..., x_n)\ x_i \geq 0,\ i = 1,...,n\}$.

Линейная независимость. Система $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m$ n-мерных векторов называется *линейно зависимой*, если найдутся такие числа $\lambda_1, \lambda_2, ..., \lambda_m$, из которых хотя бы одно отлично от нуля, что выполняется равенство $\lambda_1\,\mathbf{e}_1 + \lambda_m\,\mathbf{e}_m = 0$; в противном случае данная система векторов называется *линейно независимой*, то есть указанное равенство возможно лишь в случае, когда все $\lambda_1 = \lambda_2 = ... = \lambda_m = 0$. Геометрический смысл линейной зависимости векторов в $\mathbf{R}^3$, интерпретируемых как направленные отрезки, поясняют следующие теоремы.

Теорема 1. Система, состоящая из одного вектора, линейно зависима тогда и только тогда, когда этот вектор нулевой.

Теорема 2. Для того, чтобы два вектора были линейно зависимы, необходимо и достаточно, чтобы они были коллинеарны (параллельны).

Теорема 3. Для того, чтобы три вектора были линейно зависимы, необходимо и достаточно, чтобы они были компланарны (лежали в одной плоскости).

Левая и правая тройки векторов. Тройка некомпланарных векторов **a, b, c** называется *правой*, если наблюдателю из их общего начала обход концов векторов **a, b, c** в указанном порядке кажется совершающимся по часовой стрелке. В противном случае, **b, c** - *левая тройка*. Все правые (или левые) тройки векторов называются *одинаково ориентированными*.

Базис и координаты. Тройка $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ некомпланарных векторов в $\mathbf{R}^3$ называется *базисом*, а сами векторы $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ - *базисными*. Любой вектор **a** может быть единственным образом разложен по базисным векторам, то есть представлен в виде

$$\mathbf{a} = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2 + x_3 \mathbf{e}_3, \qquad (1.1)$$

числа x_1, x_2, x_3 в разложении (1.1) называются *координатами* вектора **a** в базисе $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ и обозначаются **a** (x_1, x_2, x_3).

Ортонормированный базис. Если векторы $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ попарно перпендикулярны и длина каждого из них равна единице, то базис называется *ортонормированным*, а координаты x_1, x_2, x_3 - *прямоугольными*. Базисные векторы ортонормированного базиса будем обозначать **i, j, k.**

Будем предполагать, что в пространстве $\mathbf{R}^3$ выбрана правая система декартовых прямоугольных координат {0, **i, j, k**}.

Векторное произведение. *Векторным произведением* вектора **a** на вектор **b** называется вектор **c**, который определяется следующими тремя условиями:

1. Длина вектора **c** численно равна площади параллелограмма, построенного на векторах **a** и **b**, т. е.
$\mathbf{c} = |\mathbf{a}||\mathbf{b}| \sin(\mathbf{a}\hat{}\mathbf{b})$.

2. Вектор **c** перпендикулярен к каждому из векторов **a** и **b.**

3. Векторы **a, b** и **c**, взятые в указанном порядке, образуют правую тройку.

Для векторного произведения **c** вводится обозначение $\mathbf{c} = [\mathbf{ab}]$ или
$\mathbf{c} = \mathbf{a} \times \mathbf{b}$.

Если векторы **a** и **b** коллинеарны, то $\sin(\mathbf{a}\hat{}\mathbf{b}) = 0$ и $[\mathbf{ab}] = 0$, в частности, $[\mathbf{aa}] = 0$. Векторные произведения ортов: $[\mathbf{ij}] = \mathbf{k}$, $[\mathbf{jk}] = \mathbf{i}$, $[\mathbf{ki}] = \mathbf{j}$.

Если векторы **a** и **b** заданы в базисе **i, j, k** координатами **a** (a_1, a_2, a_3), **b** (b_1, b_2, b_3), то

$$\left[\vec{a}\vec{b}\right] = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = = \vec{i}(a_2 b_3 - a_3 b_2) - \vec{j}(a_1 b_3 - a_3 b_1) + \vec{k}(a_1 b_2 - a_2 b_1).$$

Смешанное произведение. Если векторное произведение двух векторов **a** и **b** скалярно умножается на третий вектор **c**, то такое произведение трех векторов называется *смешанным произведением* и обозначается символом **a b c.**

Если векторы **a, b** и **c** в базисе **i, j, k** заданы своими координатами
a (a_1, a_2, a_3), **b** (b_1, b_2, b_3), **c** (c_1, c_2, c_3), то

$$\vec{a}\vec{b}\vec{c} = \begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix}$$

Смешанное произведение имеет простое геометрическое толкование — это скаляр, по абсолютной величине равный объему параллелепипеда, построенного на трех данных векторах.

Если векторы образуют правую тройку, то их смешанное произведение есть число положительное, равное указанному объему; если же тройка **a, b, c** - левая, то **a b c** <0 и V = **- a b c**, следовательно V = |**a b c**|.

Координаты векторов, встречающиеся в задачах первой главы, предполагаются заданными относительно правого ортонормированного базиса. Единичный вектор, сонаправленный вектору **a,** обозначается символом $\mathbf{a}^0$. Символом **r=OM**обозначается радиус-вектор точки M, символами a, AB или **|a|, |AB|** обозначаются модули векторов **a** и **AB.**

Пример 1.2. Найдите угол между векторами **a** = 2**m**+4**n** и **b** = **m-n**, где **m** и **n** - единичные векторы и угол между **m** и **n** равен 120^0.

Решение. Имеем: cos φ = **ab**/ab, **ab** = (2**m**+4**n**) (**m-n**) = 2 $\mathbf{m}^2$ - 4$\mathbf{n}^2$+2**mn** =

= 2 - 4+2cos120^0 = - 2 + 2(-0.5) = -3; a = $\sqrt{\mathbf{a}^2}$; $\mathbf{a}^2$ = (2**m**+4**n**) (2**m**+4**n**) =

= 4 $\mathbf{m}^2$ +16**mn**+16 $\mathbf{n}^2$ = 4+16(-0.5)+16=12, значит a = $\sqrt{12}$, b = $\sqrt{\mathbf{b}^2}$; $\mathbf{b}^2$ =

= (**m-n**)(**m-n**) = $\mathbf{m}^2$ -2**mn**+ $\mathbf{n}^2$ = 1-2(-0.5)+1 = 3, значит b = $\sqrt{3}$. Окончательно имеем: cos φ

= $\frac{-3}{\sqrt{12 \cdot 3}}$ = -1/2, φ = 120^0.

Пример 1.3. Зная векторы **AB** (-3,-2,6) и **BC** (-2,4,4), вычислите длину высоты AD треугольника ABC.

Решение. Обозначая площадь треугольника ABC через S, получим:

S = 1/2 BC AD. Тогда AD=2S/BC, BC= $\sqrt{\overrightarrow{BC}^2}$ = $\sqrt{(-2)^2+4^2+4^2}$ = 6,

S = 1/2| **AB** × **AC**|. **AC=AB+BC**, значит, вектор **AC** имеет координаты

$\vec{AC}(-5,2,10)$

$$\vec{AB}\times\vec{AC}=\begin{vmatrix}\vec{i} & \vec{j} & \vec{k}\\ -3 & -2 & 6\\ -5 & 2 & 10\end{vmatrix}=\vec{i}(-20-12)-\vec{j}(30-30)+\vec{k}(-6-10)=$$

= -16(2 $\vec{i}$ + $\vec{k}$). |**AB×AC**| = $\sqrt{16^2(2^2+1)}$ = $16\sqrt{5}$; S = $8\sqrt{5}$,откуда

AD = $\frac{16\sqrt{5}}{6}$ = $\frac{8\sqrt{5}}{3}$.

Пример 1.4. Даны два вектора **a** (11,10,2) и **b** (4,0,3). Найдите единичный вектор **c,** ортогональный векторам **a** и **b** и направленный так, чтобы упорядоченная тройка векторов **a, b, c** была правой.

Решение. Обозначим координаты вектора **c** относительно данного правого ортонормированного базиса через x, y, z.

Поскольку **c** ⊥ **a, c** ⊥ **b**, то **ca** = 0, **cb** = 0. По условию задачи требуется, чтобы c = 1 и **a b c** >0.

Имеем систему уравнений для нахождения x, y, z: 11x +10y + 2z = 0, 4x+3z=0, $x^2+y^2+z^2=0$.

Из первого и второго уравнений системы получим z = -4/3 x, y = -5/6 x. Подставляя y и z в третье уравнение, будем иметь: $x^2 = 36/125$, откуда

$x = \pm \frac{6}{5\sqrt{5}}$. Используя условие **a b c** >0, получим неравенство

$$\begin{vmatrix} 11 & 10 & 2 \\ 4 & 0 & 3 \\ x & y & z \end{vmatrix} > 0 \quad \text{или} \quad 5(6x - 5y - 8z) > 0$$

С учетом выражений для z и y перепишем полученное неравенство в виде: 625/6 x > 0, откуда следует, что x>0. Итак, $x = \frac{6}{5\sqrt{5}}$, $y = -\frac{1}{\sqrt{5}}$, $z = -\frac{8}{5\sqrt{5}}$.

ПРАКТИЧЕСКАЯ РАБОТА №7

ВЫПОЛНЕНИЕ ДЕЙСТВИЙ НАД ВЕКТОРАМИ. РЕШЕНИЕ ПРОСТЕЙШИХ ЗАДАЧ АНАЛИТИЧЕСКОЙ ГЕОМЕТРИИ НА ПЛОСКОСТИ.

Вариант 1

1. Дан отрезок AB: A (-6; 3), B (3; 4). Найти на отрезке точку M (x; y), чтобы AM = 0,4AB. (Сделать чертёж.)
2. Найти периметр треугольника, вершинами которого являются точки (3; 4), (-2; 4), (2; 2).
3. Найти координаты точки, делящей отрезок между точками A (-2; 3) и B (4; 6) в соотношении 2:3. (Сделать чертёж.)
4. Вершины треугольника: A (5; 0), B (3; -8), C (1; -4). Найти длину медиан AD и BE.
5. Дана точка C (3;5), которая делит отрезок AB в соотношении AC: CB = 3:4. Найти координаты начала отрезка A, если концом служит точка B (-1; 1). (Сделать чертёж.)
6. Точка M (1; 1) делит пополам отрезок, заключённый между точками A (x; 5) и B (-2; y). Найти эти точки.
7. Дан отрезок AB: A (-2; 3), B (2; -3). Точка M (x; y) делит отрезок в соотношении 2:3. Найти координаты точки деления и длину отрезка AB.
8. Дан отрезок AB: A (-1; 2), B (4; -4). До какой точки C (x; y) надо его продолжить в направлении от A к B, чтобы его длина увеличилась в 1,5 раза? (Сделать чертёж.)
9. Найти длины медиан треугольника ABC, если A (2; 1), C (0; 3), B (-2; 3). (Сделать чертёж.)
10. Дан треугольник ABC: A (-1; $3-3\sqrt{3}$), B (2; 3), C (-4; 3). Найти длину стороны AB и медианы AD. (Сделать чертёж.)

Вариант 2

11. На оси Oy найти точку, равноудалённую от точек M (6; 12) и N (-8; 10). (Сделать чертёж.)
12. Дан отрезок AB, где A (2; 5) и B (4; 8). Точкой C (x; y) отрезок делится в отношении 2:3. Найти координаты точки C.
13. Отрезок, соединяющий начало координат с точкой M (2; -6) разделить в отношении 3:7. Найти координаты точки деления.

14. Расстояние между точками A (2; y) и (x; 3) делится в точке (5;4) пополам. Найти ординату первой точки и абсциссу второй.
15. На протяжении отрезка AB с координатами A (-5; 5) и B (1; -4). Найти точку с абсциссой, равной 9. (Сделать чертёж.)
16. Отрезок AB разделён на три равные части. Найти координаты точек деления, если A (3; 2) и B (15; 6).
17. Отрезок AB = 10; A (x; 4), B (1; -2). Найти абсциссу точки A.
18. Отрезок AB разделён на 5 равных частей. Найти координаты точек деления, если A (3; 2) и B (-6; -15).
19. Дан отрезок AB: A (-2;1); B (8; 6). Найти точку C (x; y), которая делит отрезок в отношении 1:4. (Сделать чертёж.)
20. Отрезок AB разделён на три равные части точками C (3; -1) и D (1; 4). Найти длину отрезка AB и координаты его концов.

ТЕМА «*УРАВНЕНИЕ ЛИНИИ НА ПЛОСКОСТИ*»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1. 1. Пусть x и y – две произвольные переменные.

Определение: Соотношение вида $F(x, y) = 0$ называется *уравнением*, если оно справедливо не для всяких пар чисел x и y.

Пример: $2x + 7y - 1 = 0$, $x^2 + y^2 - 25 = 0$.

Если равенство $F(x, y) = 0$ выполняется для любых x, y, то, следовательно, $F(x, y) = 0$ – тождество.

Пример: $(x + y)^2 - x^2 - 2xy - y^2 = 0$

Говорят, что числа x_0 и y_0 *удовлетворяют уравнению*, если при их подстановке в это уравнение оно обращается в верное равенство.

Важнейшим понятием аналитической геометрии является понятие уравнения линии.

Определение: Уравнением данной линии называется уравнение $F(x, y) = 0$, которому удовлетворяют координаты всех точек, лежащих на этой линии, и не удовлетворяют координаты никакой из точек, не лежащих на этой линии.

Линия, определяемая уравнением $y = f(x)$, называется графиком функции $f(x)$. Переменные x и y – называются текущими координатами, т. к. являются координатами переменной точки.

Несколько примеров определения линий.

1) $x - y = 0 \Rightarrow x = y$. Это уравнение определяет прямую;

2) $x^2 - y^2 = 0 \Rightarrow (x-y)(x+y) = 0 \Rightarrow$ точки должны удовлетворять либо уравнению $x - y = 0$, либо уравнению $x + y = 0$, что соответствует на плоскости паре пересекающихся прямых, являющихся биссектрисами координатных углов;

3) $x^2 + y^2 = 0$. Этому уравнению удовлетворяет только одна точка O (0,0).

2. Определение: Любая прямая на плоскости может быть задана уравнением первого порядка

$Ax + By + C = 0$,

причем постоянные A, B не равны нулю одновременно, т.е. $A^2 + B^2 \neq 0$. Это уравнение первого порядка называют общим уравнением прямой.

В зависимости от значений постоянных A, B и C возможны следующие частные случаи:

- C = 0, A ¹ 0, B ¹ 0 – прямая проходит через начало координат

- A = 0, B ¹ 0, C ¹ 0 {By + C = 0} - прямая параллельна оси Ox

- B = 0, A ¹ 0, C ¹ 0 {Ax + C = 0} – прямая параллельна оси Oy

- B = C = 0, A ¹ 0 – прямая совпадает с осью Oy

- A = C = 0, B ¹ 0 – прямая совпадает с осью Ox

Уравнение прямой может быть представлено в различном виде в зависимости от каких–либо заданных начальных условий.

Уравнение прямой с угловым коэффициентом.

Если общее уравнение прямой Ax + By + C = 0 привести к виду:

$$y = -\frac{A}{B} * -\frac{C}{B}$$

и обозначить $-\frac{A}{B} = k, -\frac{C}{B} = b$; т. е. $y = kx + b$, то полученное уравнение называется уравнением прямой с угловым коэффициентом k.

Уравнение прямой в отрезках.

Если в общем уравнении прямой $Ax + By + C = 0\ C \neq 0$, то, разделив на –C, получим: $-\frac{A}{C}x - \frac{B}{C}y = 1$ или $\frac{x}{a} \div \frac{y}{b} = 1$, где $a = -\frac{C}{A}; v = -\frac{C}{B}$

Геометрический смысл коэффициентов в том, что коэффициент a является координатой точки пересечения прямой с осью Qx, а b – координатой точки пересечения прямой с осью Oy.

Нормальное уравнение прямой.

Если обе части уравнения Ax + By + C = 0 разделить на число $\mu = \pm\frac{1}{\sqrt{A^2+B^2}}$, которое называется нормирующем множителем, то получим

xcosj + ysinj - p = 0 –нормальное уравнение прямой.

Знак ± нормирующего множителя надо выбирать так, чтобы m×C <0.

p – длина перпендикуляра, опущенного из начала координат на прямую, а j - угол, образованный этим перпендикуляром с положительным направлением оси Ox.

3. Уравнение прямой по точке и угловому коэффициенту.

Пусть угловой коэффициент прямой равен k, прямая проходит через точку M (x_0, y_0). Тогда уравнение прямой находится по формуле: $y - y_0 = k(x - x_0)$

Уравнение прямой, проходящей через две точки.

Пусть в пространстве заданы две точки $M_1(x_1, y_1, z_1)$ и $M_2(x_2, y_2, z_2)$, тогда уравнение прямой, проходящей через эти точки: $\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1} = \frac{z-z_1}{z_2-z_1}$

Если какой- либо из знаменателей равен нулю, следует приравнять нулю соответствующий числитель.

На плоскости записанное выше уравнение прямой упрощается:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

если $x_1 \neq x_2$ и $x = x_1$, если $x_1 = x_2$.

Дробь $\frac{y_2 - y_1}{x_2 - x_1} = k$ называется угловым коэффициентом прямой.

2. $Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$.

Существует система координат (не обязательно декартова прямоугольная), в которой данное уравнение может быть представлено в одном из видов, приведенных ниже.

1) $\frac{x^2}{a^2} + \frac{y^2}{e^2} = 1$. - уравнение эллипса.

2) $\frac{x^2}{a^2} + \frac{y^2}{b^2} = -1$ - уравнение "мнимого" эллипса.

3) $\frac{x^2}{a^2} - \frac{y^2}{e^2} = 1$. - уравнение гиперболы.

4) $a^2x^2 - c^2y^2 = 0$ – уравнение двух пересекающихся прямых.

5) $y^2 = 2px$ – уравнение параболы.

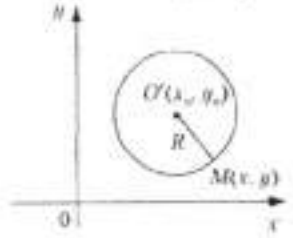
6) $y^2 - a^2 = 0$ – уравнение двух параллельных прямых.

7) $y^2 + a^2 = 0$ – уравнение двух "мнимых" параллельных прямых.

8) $y^2 = 0$ – пара совпадающих прямых.

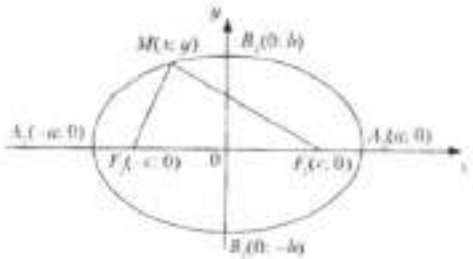
9) $(x - a)^2 + (y - b)^2 = R^2$ – уравнение окружности.

Окружность – *множество всех точек плоскости, равноудаленных от заданной точки*, называемой *центром*.



Нормальные уравнения окружности радиуса R с центром в точках $C(x_0, y_0)$ и $O(0,0)$ соответственно $(x - x_0)^2 + (y - y_0)^2 = R^2$, (2) $x^2 + y^2 = R^2$.Таким образом, центр окружности в точке $O(0,-8)$, а радиус $R = \sqrt{73}$.

Эллипс – множество всех точек плоскости, для каждой из которых сумма расстояний до двух заданных фиксированных точек, называемых *фокусами* (F_1 , F_2), есть величина постоянная.

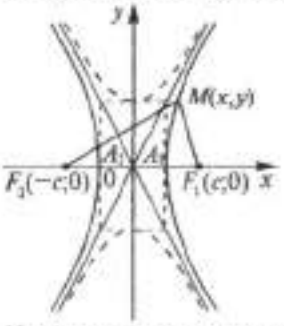


Каноническое уравнение эллипса (координатные оси совпадают с осями эллипса)

$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$. *Некоторые обозначения и характеристики эллипса: a, b – полуоси* эллипса,

$F_1(c;0),\ F_2(-c;0)$ – *фокусы* эллипса, если $a>b$ ($F_1(0;c),\ F_2(0;-c)$, если $a<b$), $c^2=a^2-b^2$ ($a>b$), ($c^2=b^2-a^2$ ($a<b$)), $\varepsilon=\frac{c}{a}<1$ – *эксцентриситет* эллипса, $r_1=a-\varepsilon x;\ \ r_2=a+\varepsilon x$ – расстояния от точки $M(x,y)$ эллипса до его фокусов *(фокальные радиусы).*

Гипербола – это множество всех точек плоскости, для каждой из которых модуль разности расстояний до двух заданных фиксированных точек, называемых *фокусами*, есть величина постоянная.



Каноническое уравнение гиперболы (координатные оси совпадают с осями гиперболы)

$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$, фокусы на OX; $-\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$, фокусы на OY.

Некоторые обозначения и характеристики гиперболы:

a, b – соответственно *действительная* и *мнимая полуоси* гиперболы (наоборот, *мнимая* и *действительная*, для гиперболы $-\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$);

$F_1(c;0),\ F_2(-c;0)$ – *фокусы* гиперболы ($F_1(0;c),\ F_2(0;-c)$ для гиперболы $-\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$);

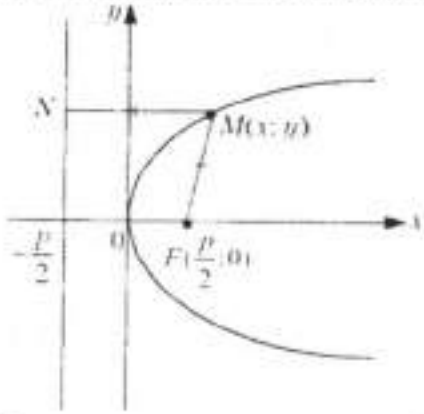
$c^2=a^2+b^2$, $\varepsilon=\frac{c}{a}>1$ – *эксцентриситет* гиперболы,

$r_1=|\varepsilon x-a|;\ \ r_2=|\varepsilon x+a|$ – расстояния от точки $M(x, y)$ эллипса до его фокусов *(фокальные радиусы).*

$y = \pm\frac{b}{a}x$ – уравнение обеих *асимптот* гиперболы;

$y = \frac{m}{x}$ – *обратная пропорциональная зависимость*.

Парабола – множество всех точек плоскости, равноудаленных от заданной точки, называемой *фокусом*, и заданной прямой, называемой *директрисой*.



Каноническое уравнение параболы с вершиной в начале координат

$y^2 = 2px$, если она симметрична относительно оси *Ox*,

$x^2 = 2py$, если она симметрична относительно оси *Oy*.

Некоторые обозначения и характеристики параболы:

$r = x + \frac{p}{2}$ – расстояние от произвольной точки *M (x,y)* параболы до фокуса $F(\frac{p}{2};0)$ параболы *(фокальный радиус)*, расстояние от произвольной точки *M (x,y)* параболы до директрисы, $x = -\frac{p}{2}$ – уравнение *директрисы* параболы.

ПРАКТИЧЕСКАЯ РАБОТА №8

СОСТАВЛЕНИЕ УРАВНЕНИЯ ПРЯМОЙ НА ПЛОСКОСТИ.

Вариант 1

21. Написать уравнение прямой, проходящей через точку пересечения прямых $\frac{x}{3} - \frac{y}{2} - 1 = 0$ и $2x + y = 3$ перпендикулярно к прямой 2x-y+1+0. (Сделать чертёж.)
22. Дан треугольник ABC: A (5; 0), B (1; 2); C (-3; -2). Найти длину медианы CE и написать её уравнение. (Сделать чертёж.)
23. Дано уравнение прямой 3x-2y=0 и на ней точки A (x; 6) и B (-2; y). найти абсциссу первой точки и ординату второй. (Сделать чертёж.)
24. Найти длину отрезка прямой 12x-5y+60=0, заключенного между осями координат.
25. Через точку пересечения прямых 2x+5y-14=0 и 4x-3y+11=0 проведена прямая, осекающая на оси Ox отрезок, равный 4. Написать её уравнение. (Сделать чертёж.)
26. В точке на прямой 4x-3y+11=0, абсцисса которой равна 1 восстановлен к ней перпендикуляр. Написать его.
27. Дан треугольник ABC: A (-3; 0), B (3; 3); C (1; -4). Написать уравнение медианы BD и высоты AE и найти острый угол между ними. (Сделать чертёж)

Вариант 2

28. Через точку пересечения прямых $\frac{x}{4}+\frac{y}{3}=1$ и $3x-6y=22$ проведена прямая перпендикулярно прямой 3y=x+6. Написать ее уравнение. (Сделать чертёж)
29. Написать уравнение прямой, проходящей через точку A (1; 2) перпендикулярно к прямой проходящей через точки (4; 3) и (-2; 1). (Сделать чертёж)
30. Найти величину внутренних углов треугольника с вершинами A (6; 0), B (3; 4); C (0; -2). (Сделать чертёж)
31. Дан треугольник ABC: A (-2; 2), B (5; 3); C (2; -4). Написать уравнение медианы AE, высоты BD и найти острый угол между ними. (Сделать чертёж)
32. Дан треугольник ABC: A (5; 3), B (3; -4); C (-4; 0). Написать уравнение медианы AD, найти её длину и угол A.
33. Дан треугольник ABC: A (-1; 2), B (5; 7); C (1; -3). Вычислить угол между стороной AB и медианой, проведенной из вершины C.
34. Написать уравнение прямой, проходящей через середину отрезка, соединяющей точки M (-1; 2) и N (4; -1).

ПРАКТИЧЕСКАЯ РАБОТА №9

ВЗАИМНОЕ РАСПОЛОЖЕНИЕ ПРЯМЫХ НА ПЛОСКОСТИ.

Вариант 1

1) Прямая EF, не лежащая в плоскости ABC, параллельна стороне AB параллелограмма ABCD. Выясните взаимное расположение прямых EF и CD.

2) В ромбе ABCD диагонали пересекаются в точке O, точка F не лежит в плоскости ABC. Можно ли провести плоскость через прямую FC и точки A и C? Ответ обоснуйте.

Вариант 2

1) Прямая MT, не лежащая в плоскости ABC, параллельна стороне BC параллелограмма ABCD. Выясните взаимное расположение прямых MT и CD.

2) Даны параллелограмм ABCD и точка E, не лежащая в плоскости ABC. Как расположены прямая AC и плоскость BDE? Ответ обоснуйте.

ПРАКТИЧЕСКАЯ РАБОТА №10

СОСТАВЛЕНИЕ И ИССЛЕДОВАНИЕ УРАВНЕНИЙ ОКРУЖНОСТИ, ЭЛЛИПСА.

Вариант 1

41. Написать уравнение окружности с центром в точке (4; 2), если она проходит через точку M (3; 2).
42. Написать уравнение эллипса, если его фокусы имеют координаты $(0; \pm 5)$, а эксцентриситет равен $\frac{2}{3}$.
43. Построить окружность по ее уравнению: $4x^2+4y^2-28x-95=0$.
44. Дано уравнение окружности: $x^2+y^2+6x-10y+18=0$. Какой вид примет уравнение этой окружности, если начало координат перенести в центр? Найти радиус. (Сделать чертёж)
45. Написать уравнение окружности, если она проходит через точки A (8; 2) и B (10;0), а радиус равен 10 единиц длинны.
46. Написать уравнение окружности, которая проходит через точку (5; 3), а ее центр лежит в точке пересечения прямых 5x-3y-13=0 и x+4y+2=0.

47. Написать уравнение эллипса, если абсцисса одного из фокусов равна 3, а эксцентриситет равен 0,6.

Вариант 2

48. Дана окружность $x^2+y^2+6x-8y-7=0$. Найти координаты центра и радиуса окружности. (Сделать чертёж)
49. Написать уравнение окружности, проходящей через начало координат, если ее центр лежит в вершине параболы $y^2 = 2x^2-4x+1$. (Сделать чертёж)
50. Найти расстояние от центра окружности $x^2+y^2+6x-4y-12=0$ до точки М (-3; 5). (Сделать чертёж)
51. Определить координаты центра и радиуса окружности $x^2+y^2-8x+6y+21=0$.
52. Дано уравнение эллипса:$2x^2+3y^2-6=0$. Найти длины осей, координат фокусов и эксцентриситет.
53. Найти точки пересечения окружности, $x^2+y^2-169=0$ с прямой $5x-12y=0$.
54. Дано уравнение окружности $x^2+y^2+6x+10y-15=0$. Написать уравнение концентрической окружности, радиус которой вдвое меньше радиуса данной окружности. (Сделать чертёж)

ПРАКТИЧЕСКАЯ РАБОТА №11

СОСТАВЛЕНИЕ И ИССЛЕДОВАНИЕ УРАВНЕНИЙ ГИПЕРБОЛЫ И ПАРАБОЛЫ.

Вариант 1

55. Написать уравнение гиперболы и ее асимптот, если мнимая ось 6, а эксцентриситет $\frac{5}{4}$.
56. Написать уравнение параболы, симметричной относительно оси Оу и проходящей через точки (-2; 4) и $(\sqrt{3}; 0)$. (Сделать чертёж)
57. Написать уравнение гиперболы, если разность между действительной и мнимой полностью равна 1, а эксцентриситет равен $\frac{5}{4}$.

Вариант 2

58. Дано уравнение параболы: $y = -x^2+6x-5$. Найти координаты ее вершины и точек пересечения с осями координат. (Сделать чертёж)
59. Написать уравнение гиперболы и ее асимптот, если действительная ось равна 8, а эксцентриситет равен 1,25. (Сделать чертёж)
60. Дано уравнение параболы: $y = x^2-4x+1$. Найти координаты вершины и фокуса параболы и написать уравнение директрисы. (Сделать чертёж)

Контрольная работа №2

Элементы аналитической геометрии.

1 вариант

1. Даны точки А (1;1) и В (4;2). Составить уравнение прямой, проходящей через эти точки.
2. Составить уравнение плоскости, проходящей через три точки А (-3;2; -1), В (-1;2;4), С (3;3; -1), не лежащие на одной прямой. Найти координаты вектора нормали.
3. Составить каноническое уравнение прямой по точке К (-2;0;3) и направляющему вектору.
4. Найти координаты центра и радиус окружности, если её уравнение задано в виде:

$x^2+y^2+10x-6y-15=0$.

5. Составьте уравнение эллипса, если расстояние между фокусами равно 6 (фокусы лежат на оси ох) и большая ось равна 10.

6. Найти полуоси, координаты фокусов для эллипса и построить его.

2 вариант

1. Даны точки А (-1;2) и В (2,1). Составить уравнение прямой, проходящей через эти точки.

2. Составить уравнение плоскости, проходящей через три точки А(1;-2;0), В(2;0;-1), С(0;-1;2), не лежащие на одной прямой. Найти координаты вектора нормали.

3. Составить каноническое уравнение прямой по точке М (-2;0;3) и направляющему вектору.

4. Найти координаты центра и радиус окружности, если её уравнение задано в виде: $x^2+y^2+8x-4y+40=0$.

5. Составьте уравнение эллипса, если расстояние между фокусами равно 8 (фокусы лежат на оси ох) и малая ось равна 6.

6. Найти полуоси, координаты фокусов эллипса и построить его.

РАЗДЕЛ 3. ОСНОВЫ МАТЕМАТИЧЕСКОГО АНАЛИЗА

ТЕМА «*МНОЖЕСТВА*»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1. Основные понятия. Понятие *множества* является первичным и поэтому формально не может быть определено. Обычно множество объясняют, следуя основателю теории множеств Г. Кантору, как совокупность объектов произвольной природы, рассматриваемую, как единое целое.

Объекты, составляющие множество, называют его *элементами*. Множества обозначают прописными буквами латинского алфавита (X, Y, $\dots$), элементы множеств – строчными буквами (x, y, $\dots$). Утверждение «элемент x принадлежит множеству X» записывается следующим образом: $x \in X$ ($\in$ – символ принадлежности). В противном случае – $x \notin X$ (или $x \overline{\in} X$).

Факт принадлежности элемента x множеству X можно указать с помощью *характеристической функции множества*:

$$\mu_X(x) = \begin{cases} 1, & \text{если } x \in X, \\ 0, & \text{если } x \notin X. \end{cases}$$

На рис. 1 показана принадлежность точек отрезка $[a, b]$ множеству X. В более общем случае, предложенном Л. Заде, функция $\mu_X(x)$ отражает частичную принадлежность элемента x множеству X, принимая значения из интервала $(0, 1]$.

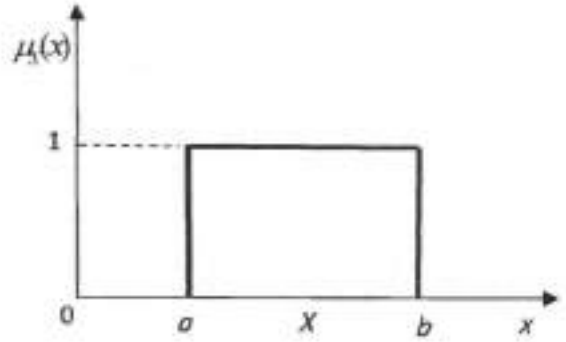


Рис. 1. Характеристическая функция множества X на отрезке $[a, b]$

Если каждый элемент множества X входит в множество Y, то X на-

зывается *подмножеством* Y. При этом пишут: $X \subseteq Y$ или $Y \supseteq X$. Если $X \subseteq Y$ и $X \neq Y$, то пишут $X \subset Y$ или $Y \supset X$. Здесь $\subseteq, \subset, \supseteq, \supset$ – символы включения. Множества X и Y равны, если они состоят из одних и тех же элементов, иначе говоря, если $X \subseteq Y$ и $Y \subseteq X$.

Множество, не содержащее ни одного элемента, называется *пустым* (обозначается $\varnothing$). Множество называется *истинным*, если оно не пустое.

Из определения подмножества следует, что каждое множество является подмножеством самого себя: $X \subseteq X$. Кроме того, считают, что пустое множество есть подмножество любого множества X: $\varnothing \subset X$.

Различают два вида подмножеств множества X: само X и $\varnothing$ называют *несобственными* подмножествами; все остальные подмножества, если они существуют, – *собственными*.

В теории множеств для удобства и краткости записей используют специальные обозначения:

$\forall$ – квантор общности (означающий «любой», «для всех», «каков бы ни был»);

$\exists$ – квантор существования (означающий «существует», «найдется», «можно найти»);

$\Rightarrow$ – импликация (символ следствия, означающий «влечет за собой).

С помощью этих символов условие $X \subset Y$ запишется так:

$$\forall x: x \in X \Rightarrow x \in Y.$$

Множество X называется *эквивалентным* множеству Y (обозначается $X \sim Y$), если между элементами X и Y можно установить взаимно однозначное соответствие. Различают *конечные* и *бесконечные* множества. Число элементов множества X называется его *мощностью* или *кардинальным числом* и обозначается $|X|$ или $\operatorname{card} X$ (англ. *cardinality* – мощность). Таким образом, конечное множество, содержащее n элементов, имеет мощность $|X| = n$. Если X эквивалентно множеству натуральных чисел N, то его называют *счетным* и его мощность обозначается через N_0. Множество X, эквивалентное множеству действительных чисел R, называется *континуальным*, а его кардинальное число *c – мощностью континуума*.

2. Способы задания. Для задания множества существуют различные способы. Множество *считают заданным, если о каждом элементе можно сказать, принадлежит он данному множеству* или нет.

Использование общепринятых обозначений. Для числовых множеств имеем: N – множество натуральных чисел; Z – множество целых чисел; P – множество простых чисел, Q – множество рациональных чисел; R – множество действительных чисел; R^* – множество действительных положительных чисел; чисел; R_0 – множество действительных неотрицательных чисел; C – множество комплексных чисел.

Перечислительный способ. Множество задается *перечислением* его элементов:

$$X = \{x_1, x_2, \ldots, x_n\}.$$

Этот способ применим к заданию конечных множеств. Согласно принципу равенства множеств по их объему (содержимому) порядок элементов множества несущественен.

Пример 1. $X_1 = \{a, b, c\}$, $X_2 = \{b, c, a\}$, $X_1 = X_2$. □

Множество, допускающее включение одного и того же элемента несколько раз, называется *мультимножеством*. Число повторений элемента называется его *кратностью*. В сокращенной записи мультимножества вместе с элементом x_i указывается его кратность m_i. Кратность $m_i = 1$ может опускаться.

Пример2. Представить мультимножеством простые множители числа 720. Этими множителями являются: $720 = 2^4 \cdot 3^2 \cdot 5$. Они образуют мультимножество $X = \{2, 2, 2, 2, 3, 3, 5\}$, которое в сокращенной записи имеет вид

$X = \{(2,4), (3,2), 5\}$

или в другой записи:

$X = \{(4/2), (2/3), 5\}$,

где $m_1 = 4$, $m_2 = 2$, $m_3 = 1$. □

Описательный способ. Этот способ применим для конечных и бесконечных множеств. Он состоит в задании характеристического свойства $P(x)$ (одноместного предиката), которым обладают все элементы x множества X и только они. Тогда имеет место запись $X = \{x \mid P(x)\}$ или $X = \{x : P(x)\}$.

Пример 3. Отрезок [0, 1] действительной прямой можно определить следующим образом

$[0, 1] = \{x \mid x \in R, \ 0 \le x \le 1\}$. □

Пример 4. $X = \{x \mid x \in N, \ x \mod 2 = 0\}$ – множество всех натуральных четных чисел. □

Рекурсивный способ.

1°. Указываются некоторые исходные элементы, входящие в множество.

2°. Описывается механизм получения новых элементов из исходных.

3°. Объявляется, что в множестве нет никаких других элементов кроме тех, которые можно получить из исходных, применяя описанный в п. 2° механизм.

Пример 5. Множество $X = \{1, 2, 2^2, 2^3, \ldots\} \subset N$ можно задать рекурсивно:

1°. $x = 1 \in X$.

2°. $\forall x \in X \Rightarrow 2 \cdot x \in X$.

3°. X – наименьшее подмножество натурального ряда, удовлетворяющее условиям 1° и 2°.

Условие 3° определяет X как пересечение всех множеств, удовлетворяющих условиям 1° и 2°. □

В дальнейшем при рекурсивном задании множеств последний пункт, как правило, не указывается, но всякий раз это требование подразумевается.

Пример 1.6. Рассмотрим совокупность всех слов в алфавите $X = \{x_1, x_2, \ldots, x_m\}$, элементы которого будем называть буквами, а элементы определяемого множества X^* – словами.

1°. Каждая буква является словом: $x_i \in X^*$, $i = 1, 2, \ldots, m$.

2°. Результат приписывания к слову любой буквы является словом:

$$\forall x \in X^* \Rightarrow xx_1 \in X^*,\ xx_2 \in X^*,\ \ldots,\ xx_m \in X^*.$$

Например, если $X = \{0, 1\}$, то X^* содержит следующие элементы:

0, 1 (согласно 1°);

00, 01, 10, 11 (согласно 2°);

000, 001, 010, 011, 100, 101, 110, 111 (согласно 2°) и так далее. □

Теория множеств, рассматриваемая без ограничений на способы задания множеств, называется *наивной теорией множеств*. В этой теории еще при жизни ее создателя Г. Кантора были обнаружены многочисленные парадоксы. Приведем один из известных парадоксов Б. Рассела, открытый им в 1903 г.

Пусть M – множество всех множеств, которые не содержат себя в качестве своего элемента. Содержит ли множество M само себя в качестве элемента? Если да, то, по определению M, оно не должно быть элементом M – противоречие. Если нет – то, по определению M, оно должно быть элементом M – вновь противоречие.

Для преодоления противоречий в наивной теории множеств было предложено несколько возможных ее аксиоматизаций, в рамках которых утверждение о существовании *множества всех множеств* было бы не выводимым.

3. Операции над множествами. Во многих случаях удается избежать противоречий наивной теории множеств, если выбрать некоторое так называемое *универсальное множество* U (*универсум*) и ограничиться рассмотрением только его подмножеств. Если некоторые множества взять в качестве исходных, то из них можно получить новые с помощью следующих операций.

Объединением множеств X и Y (обозначение $X \bigcup Y$) называется множество, состоящее из всех тех элементов, которые принадлежат хотя бы одному из множеств X или Y:

$$X \bigcup Y = \{x \mid x \in X \vee x \in Y\}.$$

Вместо символа объединения $\bigcup$ используется также символ +.

Пересечением множеств X и Y (обозначение $X \bigcap Y$) называется множество, состоящее из элементов, принадлежащих каждому из множеств X и Y:

$$X \bigcap Y = \{x \mid x \in X \wedge x \in Y\}.$$

Для операции пересечения используются также другие обозначения:

$$X \bigcap Y = X \cdot Y = XY.$$

Аналогично определяются объединение и пересечение произвольной совокупности множеств X_α, $\alpha \in A$. Здесь A – множество индексов. Если A – множество n первых натуральных чисел, то употребляются обозначения $\bigcup_{i=1}^{n} X_i$ и $\bigcap_{i=1}^{n} X_i$, а в случае, если $A = N$, то будем писать $\bigcup_{i=1}^{\infty} X_i$ и $\bigcap_{i=1}^{\infty} X_i$.

Разностью множеств X и Y (обозначение $X \setminus Y$) называется множество, состоящее из всех элементов X, не принадлежащих Y:

$$X \setminus Y = \{x \mid x \in X \wedge x \notin Y\}.$$

В отличие от двух предыдущих операций разность некоммутативна: $X \setminus Y \neq Y \setminus X$. Если $X \setminus Y = \varnothing$, то $X \subseteq Y$.

Симметрической разностью множеств X и Y (обозначение $X \oplus Y$) называется множество элементов X и Y, которые содержатся только в одном из этих множеств:

$$X \Delta Y = \{x \mid x \in (X \setminus Y) \cup (Y \setminus X)\}.$$

Дополнением к множеству X (обозначение $\bar{X}$) относительно универсального множества U называется множество:

$$\bar{X} = U \setminus X \quad \text{или} \quad \bar{X} = \{x \mid x \notin X\}.$$

Операции объединения, пересечения и дополнения часто называют *булевыми операциями* над множествами. Так как операция разности не обладает свойством ассоциативности, то ее выражают через другие операции, например, операции дополнения и пересечения:

$$A \setminus B = A \cdot \bar{B}.$$

Универсальное множество позволяет геометрически изображать множества и операции над ними с помощью *диаграмм Венна* (рис. 2).

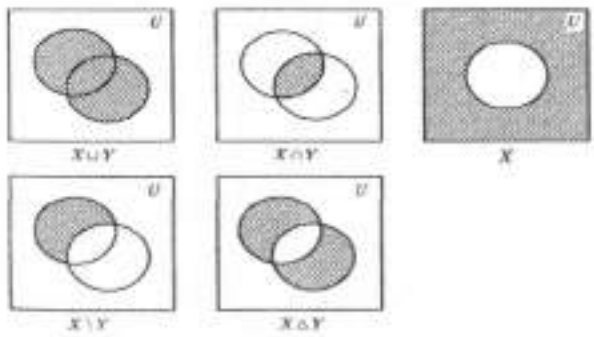


Рис. 2. Диаграммы Венна

ПРАКТИЧЕСКАЯ РАБОТА №12

ВЫПОЛНЕНИЕ ОПЕРАЦИЙ НАД МНОЖЕСТВАМИ.

Вариант 1

Найдите объединение, пересечение, разность, прямое произведение, декартов квадрат каждого множества. Составьте все подмножества множеств A и B, если

$A = \{3; 4; 7; 0\}, B = \{4; 8; -2; 0\}$

Вариант 2

Найдите объединение, пересечение, разность, прямое произведение, декартов квадрат каждого множества. Составьте все подмножества множеств A и B, если

$A = \{f; y; 9; 0; g\}, B = \{5; 8; g; y; 9\}$

ТЕМА «*ТЕОРИЯ ПРЕДЕЛОВ. НЕПРЕРЫВНОСТЬ.*»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

I. Что необходимо знать по теме:

1. Определение предела функции: (два вида)
   1.1. $\left.\begin{matrix} x \to a \\ y = f(x) \to A \end{matrix}\right\} \Longrightarrow \lim\limits_{x \to a} f(x) = A$
   «Число А называется пределом функции y=f(x) при $x \to a$, если при приближении x к a как слева, так и справа значение функции приближённо равно A(f(x)≈A). Существование придела не зависит от того принадлежит ли "a" О.О.Ф.»
   1.2. $\left.\begin{matrix} x \to \infty \\ y = f(x) \to A \end{matrix}\right\} \Longrightarrow \lim\limits_{x \to \infty} f(x) = A$
2. Теоремы о пределах:
   2.1. $\lim\left(f_1(x) \pm f_2(x)\right) = \lim f_1(x) \pm \lim f_2(x)$
   2.2. $\lim C = C$, где $C = const$ (т.е. число)
   2.3. $\lim Cf(x) = C \lim f(x)$
   2.4. $\lim f_1(x) * f_2(x) = \lim f_1(x) * \lim f_2(x)$
   2.5. $\lim \frac{f_1(x)}{f_2(x)} = \frac{\lim f_1(x)}{\lim f_2(x)}$, если $\lim f_2(x) \neq 0$
   2.6. $\lim\left(f(x)\right)^n = (\lim f(x))^n$
   2.7. $\lim \sqrt[n]{f(x)} = \sqrt[n]{\lim f(x)}$
   2.8. $\lim\limits_{x \to a} x = a$
3. Предел непрерывной функции.
   3.1. $\lim\limits_{x \to a} f(x) = f(a)$, если $f(x)$ – непрерывна в точке $x = a$, т.е. для непрерывной функции предел вычисляется подстановкой
   3.2. $f(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$ – называется степенным многочленом, он всегда непрерывен $\Longrightarrow$ Предел многочлена вычисляется подстановкой
4. Бесконечно малые и бесконечно большие функции:
   4.1. $\lim f(x) = 0 \Longrightarrow f(x)$ – называется бесконечно малой (б.м.)
   4.2. $\lim f(x) = \infty \Longrightarrow f(x)$ – называется бесконечно большой (б.б.)
   4.3. Если $f(x)$ – б.м., то $\frac{1}{f(x)}$ – б.б. и наоборот
5. Правило Лопиталя:

$$\lim \frac{f(x)}{\varphi(x)} \left(\frac{0}{0}\right) = \lim \frac{f'(x)}{\varphi'(x)}$$

II. Схема решения примеров на вычисление пределов

1. Предел многочлена:
   1.1. при $x \to a \Longrightarrow$ вычисляется подстановкой
   1.2. при $x \to \infty \Longrightarrow$ равен ∞
2. Предел дроби: (раскрытие неопределённости)
   2.1. $\frac{\text{число}}{\text{число}} \Longrightarrow \left(\frac{\quad}{\text{зн} \neq 0}\right) \Longrightarrow$ вычисляется подстановкой
   2.2. $\frac{\text{число}}{0} \Longrightarrow$ равен ∞
   2.3. $\frac{0}{0} \Longrightarrow$ 1) разложить на множители (см. примечание №1 и №2)
   2) сократить дробь
   3) подставить и вычислить значение предела
   2.4. $\frac{\text{число}}{\infty} \Longrightarrow$ равен 0
   2.5. $\frac{\infty}{\infty} \Longrightarrow$ (см. примечание №3)
   1) если самая большая степень числителя меньше знаменателя $\Longrightarrow$ равен 0
   2) если самая большая степень числителя больше знаменателя $\Longrightarrow$ равен ∞
   3) если самая большая степень числителя равна знаменателю $\Longrightarrow$ равен отношению коэффициентов при этих степенях.

Примечания:

№1 Для разложения на множители надо знать формулы:

1) $a^2 - b^2 = (a + b) * (a - b)$ – разность квадратов
2) $a^3 + b^3 = (a + b) * (a^2 - a * b + b^2)$ – сумма кубов
3) $a^3 - b^3 = (a - b) * (a^2 + a * b + b^2)$ – разность кубов
4) $ax^2 + bx + c = a(x - x_1)(x - x_2)$ – где x_1 и x_2 корни квадратного трехчлена

№2 Если в записи числителя или знаменателя присутствует $\sqrt{\ }$, то вместо разложения на множители необходимо числитель и знаменатель домножить на одно и тоже выражение, чтобы получить формулу a^2-b^2 – преобразовать её до тех пор, пока не произойдет сокращение.

№3 В числителе и знаменателе дроби должны быть только степенные многочлены (если есть скобки их необходимо раскрыть).

3. Замечательные пределы:
   3.1. $\lim\limits_{x \to \infty} \left(1 + \frac{1}{x}\right)^x = e, \ e \approx 2{,}72$
   3.2. $\lim\limits_{x \to 0} (1 + x)^{\frac{1}{x}} = e$
   3.3. $\lim\limits_{x \to 0} \frac{\sin x}{x} = 1$

III. Примеры на вычисление пределов

При вычислении предела дроби проверку необходимо начинать со знаменателя. Результат подсчета ставится в скобках рядом с дробью.

1) $\lim\limits_{x \to 2}(3x^3 - 4x^2 - x - 4) = 3 * 2^3 - 4 * 2^2 - 2 + 5 = 11$
2) $\lim\limits_{x \to \infty}(3x^3 - 4x^2 - x - 4) = \infty$
3) $\lim\limits_{x \to 2} \frac{x^2-4}{x^2-x+1} = \frac{2^2-4}{2^2-2+1} = \frac{0}{4-2+1} = 0$
4) $\lim\limits_{x \to 3} \frac{x^2-2}{x-3} \left(\frac{3^3-4=\text{число}}{3-3=0}\right) = \infty$

5) $\lim\limits_{x\to 2}\frac{x^2-4}{x^2-x-2}\left(\frac{2^2-4=0}{2^2-2-2=0}\right)=\lim\frac{(x-2)*(x+2)}{(x+1)*(x-2)}=\frac{2+2}{2+1}=\frac{4}{3}$

1. $x^2-4=(x-2)*(x+2)$ - разность квадратов
2. $x^2-x-2=0$ – квадратный трехчлен

$$\left.\begin{array}{c} x^2-x-2=0 \\ D=1+4*2*1=9 \\ x_1=\dfrac{1-3}{2}=-1 \\ x_2=\dfrac{1+3}{2}=2 \end{array}\right\} x^2-x-2=1*(x+1)*(x-2)$$

6) $\lim\limits_{x\to 0}\frac{x}{1-\sqrt{x+1}}\left(\frac{0}{0}\right)=\lim\limits_{x\to 0}\frac{x*(1+\sqrt{x+1})}{(1-\sqrt{x+1})*(1+\sqrt{x+1})}=\lim\limits_{x\to 0}\frac{x*\left(1+\sqrt{x+1}\right)}{1^2-\left(\sqrt{x+1}\right)^2}=$

$\lim\limits_{x\to 0}\frac{x*\left(1+\sqrt{x+1}\right)}{1-(x+1)}=\lim\limits_{x\to 0}\frac{x*\left(1+\sqrt{x+1}\right)}{1-x-1}=\lim\limits_{x\to 0}\frac{x*\left(1+\sqrt{x+1}\right)}{-x}=\frac{1+\sqrt{x+1}}{-1}=-2$

7) $\lim\limits_{x\to\infty}\frac{4}{x^2-4x+5}\left(\frac{\text{число}}{\infty}\right)=0$

8) $\lim\limits_{x\to\infty}\frac{(2x-1)*(5x+1)}{3-x-12x^2}\left(\frac{\infty}{\infty}\right)=\lim\frac{10x^2+2x-5x-1}{2-x-12x^2}=\frac{10}{-12}=-\frac{5}{6}$

9) $\lim\limits_{x\to\infty}\frac{3x^3-4x^2+5}{7x^2-8x-1}\left(\frac{\infty}{\infty}\right)=\infty$ (степень числителя больше степени знаменателя)

10) $\lim\limits_{x\to\infty}\frac{2x^2(x-1)}{x^4-2x-5}=\lim\limits_{x\to\infty}\frac{2x^3-2x^2}{x^4-2x-5}=0$ (степень числителя меньше степени знаменателя)

11) $\lim\limits_{2x\to\infty}\left(1+\frac{1}{2x}\right)^{2x}=e$

12) $\lim\limits_{x\to\infty}\left(1+\frac{1}{x}\right)^{3x}=\left(\lim\limits_{x\to\infty}(1+\frac{1}{x})^{x}\right)^3=e^3$

13) $\lim\limits_{x\to\infty}\left(1+\frac{1}{2x}\right)^{x}=\lim\limits_{2x\to\infty}(1+\frac{1}{2x})^{2x*\frac{1}{2}}=\left(\lim\limits_{2x\to\infty}\left(1+\frac{1}{2x}\right)^{2x}\right)^{\frac{1}{2}}=e^{\frac{1}{2}}$

14) $\lim\limits_{x\to\infty}\left(1+\frac{3}{2x}\right)^{x}=\left(\lim\limits_{\frac{2}{3}x\to\infty}\left(1+\frac{1}{\frac{2}{3}x}\right)^{\frac{2}{3}x}\right)^{\frac{3}{2}}=e^{\frac{2}{3}}$

15) $\lim\limits_{x\to 0}(1+4x)^{\frac{1}{x}}=\lim\limits_{4x\to 0}(1+4x)^{\frac{1}{4x}*\frac{4}{1}}=\left(\lim\limits_{4x\to 0}(1+4x)^{\frac{1}{4x}}\right)^4=e^4$

16) $\lim\limits_{x\to 0}\left(1+\frac{x}{2}\right)^{\frac{1}{x}}=\left(\lim\limits_{\frac{1}{2}x\to 0}\left(1+\frac{1}{2}x\right)^{\frac{1}{\frac{1}{2}x}}\right)^{\frac{1}{2}}=e^{\frac{1}{2}}$

17) $\lim\limits_{x\to 0}\frac{\sin 3x}{x}=\lim\limits_{3x\to 0}\frac{3\sin 3x}{3x}=3\lim\limits_{x\to 0}\frac{\sin 3x}{3x}=3*1=3$

ПРАКТИЧЕСКАЯ РАБОТА №13

ОПРЕДЕЛЕНИЕ ОСНОВНЫХ СВОЙСТВ ФУНКЦИИ. ВЫЧИСЛЕНИЕ ПРЕДЕЛОВ ФУНКЦИИ.

I	II	III
1) $\lim\limits_{x\to1}\frac{x^2-x+1}{x^2-1}$	1) $\lim\limits_{x\to3}\frac{3x+x^2}{x^2-9}$	1) $\lim\limits_{x\to1}\frac{x^2-4}{x-1}$
2) $\lim\limits_{x\to2}\frac{x^2-x+1}{x^2-1}$	2) $\lim\limits_{x\to2}\frac{4x^2-1}{x^2-9}$	2) $\lim\limits_{x\to2}\frac{x^2-4}{x-1}$
3) $\lim\limits_{x\to1}\frac{x^2-x}{x^2-1}$	3) $\lim\limits_{x\to3}\frac{4x^2-1}{x^2-9}$	3) $\lim\limits_{x\to1}\frac{x^2-x}{x^2+x-2}$
4) $\lim\limits_{x\to1}\frac{x^3-1}{x^2+1}$	4) $\lim\limits_{x\to-2}\frac{x+1}{x^3+8}$	4) $\lim\limits_{x\to4}\frac{x^3-64}{x^2-16}$
5) $\lim\limits_{x\to1}\frac{x^3-1}{x^2-1}$	5) $\lim\limits_{x\to-2}\frac{x^2+2x}{x^3+8}$	5) $\lim\limits_{x\to-4}\frac{x^3-64}{x^2-16}$
6) $\lim\limits_{x\to2}\frac{2x+x^2}{x^2+5x+6}$	6) $\lim\limits_{x\to2}\frac{x^2-4}{3x^2-5x-2}$	6) $\lim\limits_{x\to3}\frac{x-1}{4x^2-11x-3}$
7) $\lim\limits_{x\to-2}\frac{2x+x^2}{x^2+5x+6}$	7) $\lim\limits_{x\to-2}\frac{x^2-4}{3x^2-5x-2}$	7) $\lim\limits_{x\to3}\frac{x-3}{4x^2-11x-3}$
8) $\lim\limits_{x\to3}\frac{4x^2-11x-3}{3x^2-8x-3}$	8) $\lim\limits_{x\to5}\frac{3x^2-17x+10}{3x^2-16x+5}$	8) $\lim\limits_{x\to2}\frac{4x^2-7x-2}{5x^2-9x-2}$
9) $\lim\limits_{x\to2}\frac{x-2}{\sqrt{x+2}+3}$	9) $\lim\limits_{x\to6}\frac{5-x}{3-\sqrt{x+3}}$	9) $\lim\limits_{x\to5}\frac{3-\sqrt{2x-1}}{5-x}$
10) $\lim\limits_{x\to2}\frac{x+2}{\sqrt{x+2}-2}$	10) $\lim\limits_{x\to6}\frac{5-x}{3+\sqrt{x+3}}$	10) $\lim\limits_{x\to5}\frac{3+\sqrt{2x-1}}{5-x}$
11) $\lim\limits_{x\to2}\frac{x-2}{\sqrt{x+2}-2}$	11) $\lim\limits_{x\to6}\frac{6-x}{3-\sqrt{x+3}}$	11) $\lim\limits_{x\to5}\frac{3+\sqrt{2x-1}}{4-x}$
12) $\lim\limits_{x\to\infty}\frac{3}{1-x^2}$	12) $\lim\limits_{x\to\infty}\frac{x^2-4x-1}{1-x^2}$	12) $\lim\limits_{x\to\infty}\frac{x^2-4x+1}{x(1-x^2)}$
13) $\lim\limits_{x\to\infty}\frac{x^3-4x^2+5x-1}{x(1-x^2)}$	13) $\lim\limits_{x\to\infty}\frac{(x-1)*(2x+1)}{x^3-2x-1}$	13) $\lim\limits_{x\to\infty}\frac{x^3-4x^2-5x+2}{1+3x-10x^3}$
14) $\lim\limits_{x\to\infty}\frac{2x^2-4x-7}{x^3+4x^2-1}$	14) $\lim\limits_{x\to\infty}\frac{10}{2x-1}$	14) $\lim\limits_{x\to\infty}\frac{(2x-1)*(x+2)}{3x-5}$
15) $\lim\limits_{x\to\infty}\frac{x(x^2-x-1)}{x^2+5x-4}$	15) $\lim\limits_{x\to\infty}\frac{x^3+2x^2-x-1}{x(2x-1)}$	15) $\lim\limits_{x\to\infty}\frac{2,5}{4x^2-x+5}$
16) $\lim\limits_{x\to\infty}\left(1+\frac{2}{x}\right)^x$	16) $\lim\limits_{x\to0}(1+5x)^{\frac{1}{x}}$	16) $\lim\limits_{x\to\infty}\left(1+\frac{4}{x}\right)^x$
17) $\lim\limits_{x\to0}\left(1+\frac{x}{3}\right)^{\frac{1}{x}}$	17) $\lim\limits_{x\to\infty}\left(1+\frac{1}{3x}\right)^x$	17) $\lim\limits_{x\to0}\left(1+\frac{x}{5}\right)^{\frac{1}{x}}$
18) $\lim\limits_{x\to\infty}\left(1+\frac{3}{x}\right)^{5x}$	18) $\lim\limits_{x\to0}(1+x)^{\frac{2}{5x}}$	18) $\lim\limits_{x\to\infty}\left(1+\frac{1}{3x}\right)^{5x}$
19) $\lim\limits_{x\to0}(1+3x)^{\frac{1}{2x}}$	19) $\lim\limits_{x\to\infty}\left(1+\frac{3}{x}\right)^{2x}$	19) $\lim\limits_{x\to0}(1+2x)^{\frac{2}{x}}$
20) $\lim\limits_{x\to0}\frac{sin\frac{x}{2}}{5}$	20) $\lim\limits_{x\to0}\frac{\sin3x}{2x}$	20) $\lim\limits_{x\to0}\frac{\sin\frac{x}{3}}{x}$

IV	V	VI
1) $\lim_{x\to 4}\frac{x^2-5x}{x^2-25}$	1) $\lim_{x\to 4}\frac{x^2-16}{x^2-4x}$	1) $\lim_{x\to 6}\frac{x+6}{x^2-36}$
2) $\lim_{x\to 5}\frac{x^2-5x}{x^2-25}$	2) $\lim_{x\to 4}\frac{x^2+16}{x^2-4x}$	2) $\lim_{x\to 6}\frac{x^2-6x}{x^2-36}$
3) $\lim_{x\to 5}\frac{x^2-25}{x^2+25}$	3) $\lim_{x\to 1}\frac{x^2+16}{x^2-4x}$	3) $\lim_{x\to 3}\frac{x+6}{x^2-36}$
4) $\lim_{x\to 1}\frac{x^2-1}{x^3+1}$	4) $\lim_{x\to 2}\frac{x^3-8}{x^2-2x}$	4) $\lim_{x\to 1}\frac{x^3+1}{x^2+x}$
5) $\lim_{x\to -1}\frac{x^2-1}{x^3-1}$	5) $\lim_{x\to 1}\frac{x^3-8}{x^2-2x}$	5) $\lim_{x\to -1}\frac{x^3+1}{x^2+x}$
6) $\lim_{x\to 3}\frac{4x^2-11x-3}{x-3}$	6) $\lim_{x\to 2}\frac{x^2-5}{x^2+x-6}$	6) $\lim_{x\to -1}\frac{x^2-x+2}{x+3}$
7) $\lim_{x\to 3}\frac{4x^2-11x+3}{x-3}$	7) $\lim_{x\to 2}\frac{x-2}{x^2+x-6}$	7) $\lim_{x\to 1}\frac{4x^2-7x+3}{3x^2-2x-1}$
8) $\lim_{x\to 4}\frac{2x^2-7x-4}{3x^2-13x+4}$	8) $\lim_{x\to 3}\frac{4x^2-11x-3}{5x^2-16x+3}$	8) $\lim_{x\to -3}\frac{2x^2+x+15}{3x^2+7x-6}$
9) $\lim_{x\to 7}\frac{\sqrt{x+2}+3}{x^2-49}$	9) $\lim_{x\to 5}\frac{x-5}{2-\sqrt{x-1}}$	9) $\lim_{x\to 4}\frac{4+\sqrt{3x+4}}{x-4}$
10) $\lim_{x\to 7}\frac{\sqrt{x+2}-3}{x^2-49}$	10) $\lim_{x\to 5}\frac{x-4}{2+\sqrt{x-1}}$	10) $\lim_{x\to 4}\frac{4-\sqrt{3x+4}}{x-4}$
11) $\lim_{x\to 7}\frac{\sqrt{x+2}+3}{x^2+49}$	11) $\lim_{x\to 5}\frac{x+5}{2-\sqrt{x-1}}$	11) $\lim_{x\to 4}\frac{4+\sqrt{3x+4}}{x+4}$
12) $\lim_{x\to\infty}\frac{3}{x^2-4x+1}$	12) $\lim_{x\to\infty}\frac{1-5x^2}{3x^2+4x-1}$	12) $\lim_{x\to\infty}\frac{(4x-1)*(2-x)}{5x-1}$
13) $\lim_{x\to\infty}\frac{x(1-x^2)}{4x^3+2x^2-x-1}$	13) $\lim_{x\to\infty}\frac{23}{4x(x^2-x-7)}$	13) $\lim_{x\to\infty}\frac{7x^2+2x-1}{1-2x+8x^2}$
14) $\lim_{x\to\infty}\frac{x^3-2x-1}{1-x^2}$	14) $\lim_{x\to\infty}\frac{(2x-1)*(1-x)}{3x^3+7x^2-1}$	14) $\lim_{x\to\infty}\frac{1}{x(4x^2-2x+1)}$
15) $\lim_{x\to\infty}\frac{x^3+2x^2-x-1}{x(1-x^3)}$	15) $\lim_{x\to\infty}\frac{4-x^2-x^3}{2x^2-6x-1}$	15) $\lim_{x\to\infty}\frac{2x^3+4x^2+8}{1-x^4}$
16) $\lim_{x\to\infty}\left(1+\frac{2}{3x}\right)^{x}$	16) $\lim_{x\to 0}(1+5x)^{\frac{1}{x}}$	16) $\lim_{x\to\infty}\left(1+\frac{3}{x}\right)^{2x}$
17) $\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^{7x}$	17) $\lim_{x\to\infty}\left(1+\frac{7}{x}\right)^{x}$	17) $\lim_{x\to 0}\left(1+\frac{x}{5}\right)^{\frac{1}{x}}$
18) $\lim_{x\to 0}(1+2x)^{\frac{1}{3x}}$	18) $\lim_{x\to\infty}\left(1+\frac{1}{3x}\right)^{2x}$	18) $\lim_{x\to\infty}\left(1+\frac{1}{4x}\right)^{x}$
19) $\lim_{x\to 0}(1+x)^{\frac{5}{x}}$	19) $\lim_{x\to 0}(1+x)^{\frac{1}{3x}}$	19) $\lim_{x\to 0}(1+x)^{\frac{2}{3x}}$
20) $\lim_{x\to 0}\frac{sin\frac{3x}{2}}{x}$	20) $\lim_{x\to 0}\frac{\sin 2x}{3x}$	20) $\lim_{x\to 0}\frac{\sin\frac{x}{5}}{x}$

ПРАКТИЧЕСКАЯ РАБОТА №14

«Исследование функций на непрерывность».

Вариант 1

1)Исследовать функцию на непрерывность:

$y=x^2$-2 при x=3

2) Найти точки разрыва и исследовать их характер

$y=\frac{3}{x^2-2x+1}$

Вариант 2

1)Исследовать функцию на непрерывность:

$y = x^2 + 4x + 3$ в точке x=2

1)Исследовать функцию на непрерывность:

$$y = \frac{x-1}{x^2-3x-10}$$

Контрольная работа №3

Теория пределов. Непрерывность.

Вариант 1

1)Вычислить пределы функций:

1. $\lim\limits_{x\to\infty}\frac{5-2x-3x^2}{x^2+x+3}$;
2. $\lim\limits_{x\to-1}\frac{4x^2+7x+3}{2x^2+x-1}$;
3. $\lim\limits_{x\to0}\frac{\sqrt{4+x}-\sqrt{4-x}}{3x^2+x}$;
4. $\lim\limits_{x\to0}\frac{8x^2\cos 3x}{\sin^2 5x}$;

2)

Вычислить пределы функции:

$$1)\ \lim_{x\to5}\frac{x^2-8x+15}{x^2-25};\ 2)\ \lim_{x\to0}\frac{x}{\sqrt{x+4}-2};\ 3)\ \lim_{x\to\infty}\frac{10x^4-8x^2+3}{5x^4+3x^3+5};$$

$$4)\ \lim_{x\to0}\frac{\sin 5x}{\sin 7x};\ 5)\ \lim_{x\to\infty}\left(1-\frac{2}{x}\right)^x$$

Вариант 2

1) Вычислить пределы функций:

1. $\lim\limits_{x\to\infty}\dfrac{3x^2+5x+4}{2x^2-x+1}$;
2. $\lim\limits_{x\to 3}\dfrac{2x^2-9x+9}{x^2-5x+6}$;
3. $\lim\limits_{x\to 0}\dfrac{2x}{\sqrt{10+x}-\sqrt{10-x}}$;
4. $\lim\limits_{x\to 0}\dfrac{7x^2}{1-\cos 4x}$;

2)Вычислить пределы функции

1) $\lim\limits_{x\to 1}\dfrac{x^3-1}{x-1}$; 2) $\lim\limits_{x\to 0}\dfrac{\sqrt{x+1}-1}{x}$; 3) $\lim\limits_{x\to\infty}\dfrac{3x^3-4x+8}{5x^3+27x^2+x}$; 4) $\lim\limits_{x\to 0}\dfrac{\sin 5x}{\sin 8x}$;

5) $\lim\limits_{x\to\infty}\left(1+\dfrac{1}{5x}\right)^x$

ТЕМА «***ДИФФЕРЕНЦИАЛЬНОЕ ИСЧИСЛЕНИЕ***»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

I. Что необходимо знать по теме:

1. Определение: $\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = y'$ «′» - штрих или «′» производная $y' = \frac{dy}{dx} \Rightarrow dy = y'dx$, dy – называют дифференциалом
2. Физический смысл производной:
   2.1. Производная – скорость изменения функции.
   2.2. $S' = V$
   2.3. $S'' = V' = a$
3. Геометрический смысл производной
   3.1. M_0T – касательная в точке M_0 к графику функции y=f(x)
   3.2. $y'_{x0} = k = tg\ a$ (k – угловой коэффициент M_0T)
   3.3. $y = y'_{x0}(x - x_0) + y_{x0}$ – уравнение касательной M_0T

4. Таблица производных

Для простых функций	Для сложных функций, где U – обозначение алгебраического выражения зависящего от x
4.1. $(U \pm V)' = (U)' \pm (V)'$	
4.2. $(C)' = 0$, C -	
4.3. $(UV)' = (U)'V + U(V)'$	
4.4. $(CU)' = C(U)'$	
4.5. $\left(\frac{U}{V}\right)' = \frac{(U)'V - U(V)'}{V^2}$	
4.6. $(x)' = 1$	
4.7. $(x^n)' = nx^{n-1}$	$(U^n)' = nU^{n-1} * (U)'$ - степенная
4.8. $(a^x)' = a^x \ln a$	4.8. $(a^U)' = a^U * (U)' * \ln a$ } - показательная
4.9. $(e^x)' = e^x$	4.9. $(e^U)' = e^U * (U)'$ }
4.10. $(\log_a x)' = \frac{1}{x \ln a}$	4.10. $(\log_a U)' = \frac{(U)'}{U \ln a}$
4.11. $(\ln x)' = \frac{1}{x}$	4.11. $(\ln U)' = \frac{(U)'}{U}$
4.12. $(\sin x)' = \cos x$	4.12. $(\sin U)' = \cos U * (U)'$
4.13. $(\cos x)' = -\sin x$	4.13. $(\cos U)' = -\sin U * (U)'$
4.14. $(tg\ x)' = \frac{1}{cos^2 x}$	4.14. $(tg\ U)' = \frac{(U)'}{cos^2\ U}$
4.15. $\left(\frac{U}{C}\right)' = \frac{(U)'}{C}$	
4.16. $(ctg\ x)' = -\frac{1}{sin^2 x}$	4.16. $(ctg\ U)' = -\frac{(U)'}{sin^2 U}$
4.17. $(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$	4.17. $(\arcsin U)' = \frac{(U)'}{\sqrt{1-U^2}}$
4.18. $(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$	4.18. $(\arccos U)' = -\frac{(U)'}{\sqrt{1-U^2}}$
4.19. $(\operatorname{arctg} x)' = \frac{1}{1+x^2}$	4.19. $(\operatorname{arctg} U)' = \frac{(U)'}{1+U^2}$
4.20. $(\operatorname{arcctg} x)' = -\frac{1}{1+x^2}$	4.20. $(\operatorname{arcctg} U)' = -\frac{(U)'}{1+U^2}$

4.19. $(\operatorname{arctg} x)' = \frac{1}{1+x^2}$	4.19. $(\operatorname{arctg} U)' = \frac{(U)'}{1+U^2}$
4.20. $(\operatorname{arcctg} x)' = -\frac{1}{1+x^2}$	4.20. $(\operatorname{arcctg} U)' = -\frac{(U)'}{1+U^2}$

5. Исследование функции на монотонность (возрастание, убывание) и экстремум (минимум, максимум)

   y = f(x) – заданная функция

   5.1. Найти О.О.Ф.

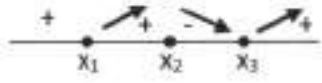
   5.2. Найти y' = f '(x)

   5.3. y' = 0 – решить уравнение, найденные корни называются критическими точками.

   5.4. На числовую прямую нанести О.О.Ф. и критические точки и по методу интервалов расставить на ней знаки производной.

   5.5. Записать ответ, при этом необходимо помнить:

   1. «+» → функция возрастает (↗)
   2. «-» → функция убывает (↘)
   3. если «+» «-» → x_0 – точка max
   4. если «-» «+» → x_0 – точка min

   Например 

   x_1 – не является экстремумом;

   x_2 – точка max; x_3 – точка min.

6. При решении примеров на вычисление производной необходимо вспомнить правила:

   6.1. Степеней:

   6.1.а. $a^n = \underbrace{a * a * a * \ldots * a}_{\text{n... раз}}$

   6.1.б. $a^0 = 1$

   6.1.в. $a^{-n} = \frac{1}{a^n}$

   6.1.г. $\sqrt[n]{a^m} = a^{\frac{m}{n}}$

   6.1.д. $a^n * a^m = a^{n+m}$

   6.1.е. $\frac{a^n}{a^m} = a^{n-m}$

   6.1.ж. $(a^n)^m = a^{n*m}$

   6.2. Логарифмов:

   6.2.а. $\log_a MN = \log_a M + \log_a N$

   6.2.б. $\log_a \frac{M}{N} = \log_a M - \log_a N$

   6.2.в. $\log_a M^n = n \log_a M$

   6.3. Для того, чтобы рационально вычислить производную необходимо упростить функцию, что означает необходимость избавиться от дробей, корней, произведений, упростить логарифм.

   6.4. Прежде чем воспользоваться таблицей производных, необходимо в записи функции найти последнее действие и по его названию отыскать формулу в таблице.

   Примечание: Необходимо заметить, что начиная с формулы 4.7 из таблицы, начинаются сложные функции, поэтому сначала надо выяснить какая функция: простая или сложная.

Например: x^5 – степенная простая (левая колонка формул)
$(2x-1)^5$ или $(\sin x)^5$ - степенные сложные (правая колонка формул)

7. Решение примеров на вычисление производных

7.1. $y = \underbrace{x^2}_{u} * \underbrace{\sin 5x}_{v}$ – упрощать нечего, последнее действие умножение

$$y' = (x^2 \sin 5x) = (x^2)' * \sin 5x + x^2 * (\sin 5x)' = 2x * \sin 5x + x^2 * \cos 5x * 5$$

степенная простая — синус сложный

7.2. $y = \frac{x^2}{7} = \frac{1}{7}x^2 \Rightarrow y' = \left(\frac{1}{7}x^2\right)' = \frac{1}{7}(x^2)' = \frac{1}{7} * 2x = \frac{2}{7}x$

7.3. $y = \frac{5}{x^2} = 5x^{-2} \Rightarrow y' = (5x^{-2})' = 5(x^{-2})' = 5(2x^{-3}) = -\frac{10}{x^3}$

7.4. $y = 2x\sqrt{x} = 2x * x^{\frac{1}{2}} = 2x^{\frac{3}{2}} \Rightarrow y' = \left(2x^{\frac{3}{2}}\right)' = 2\left(x^{\frac{3}{2}}\right)' = 2 * \frac{3}{2}x^{\frac{1}{2}} = 3\sqrt{x}$

7.5. $y = \frac{3x^2}{4\sqrt[4]{x^2}} = \frac{3}{4} * \frac{x^2}{x^{\frac{2}{4}}} = \frac{3}{4}x^{\frac{4}{3}} \Rightarrow y' = \left(\frac{3}{4}x^{\frac{4}{3}}\right)' = \frac{3}{4}\left(x^{\frac{4}{3}}\right)' = \frac{3}{4} * \frac{4}{3}x^{\frac{1}{3}} = \sqrt[3]{x}$

7.6. $y = \frac{3}{x} + \frac{2x}{3} - 1 = 3x^{-1} + \frac{2}{3}x - 1 \Rightarrow y' = \left(3x^{-1} + \frac{2}{3}x - 1\right)' = (3x^{-1})' + \left(\frac{2}{3}x\right)' - (1)' = 3(-1x^{-2}) + \frac{2}{3} * 1 - 0 = -\frac{3}{x^2} + \frac{2}{3}$

7.7. $y = \frac{4x^{-5x+1}}{7} = \frac{1}{7}(4x^2 - 5x + 1) \Rightarrow y' = \left(\frac{1}{7}(4x^2 - 5x + 1)\right)' = \frac{1}{7}(4 * 2x - 5 * 1) = \frac{8x-5}{7}$

7.8. $y = \sqrt{e^{3x-1}} = e^{\frac{3x-1}{2}} = e^{\frac{1}{2}(3x-1)} = e^{\frac{3}{2}x-\frac{1}{2}} \Rightarrow y^n = \left(e^{\frac{3}{2}x-\frac{1}{2}}\right)' = e^{\frac{3}{2}x-\frac{1}{2}} * \left(\frac{3}{2}x - \frac{1}{2}\right) = \frac{3}{2}e^{\frac{3}{2}x-\frac{1}{2}}$

7.9. $y = \ln\frac{3x}{5-x^2} = \ln(3x-1) - \ln(5-x^2) \Rightarrow y' = (\ln(3x-1) - \ln(5-x^2))' = (\ln(3x-1))' - (\ln(5-x^2))' = \frac{(3x-1)'}{3x-1} - \frac{(5-x)}{5-x^2} = \frac{3}{3x-1} - \frac{-2x}{5-x^2} = \frac{3(5-x^2)+2x(3x-1)}{(3x-1)(5-x^2)} = \frac{15-3x^2+6x^2-2x}{(3x-1)*(5-x^2)} = \frac{3x^2-2x+15}{(3x-1)*(5-x^2)}$

7.10. $y = 3x\sqrt[3]{x} - e^{-x} + tg\,\frac{1}{2}x = 3x^{\frac{4}{3}} - e^{-x} + tg\,\frac{1}{2}x \Rightarrow y'\left(3x^{\frac{4}{3}} - e^{-x} + tg\frac{1}{2}x\right)' = \left(3x^{\frac{4}{3}}\right)' - (e^{-x})' + \left(tg\,\frac{1}{2}x\right)' = 3 * \frac{4}{3}x^{\frac{1}{3}} - e^{-x}(-x)' + \frac{\left(\frac{1}{2}x\right)'}{cos^2\frac{1}{2}x} = 4\sqrt[3]{x} + e^{-x} + \frac{1}{2cos^2\frac{1}{2}x}$

7.11.

$S = \frac{1}{3}t^3 - 2t^2 + 3$

$a = ?$

$V = (\frac{1}{3}t^3 - 2t^2 + 3)'$

$a = (t^2 - 4t)' = 2t - 4$

$a = 2 * 3 - 4 = 2$ м/с^2

7.12.

$S = \frac{1}{3}t^3 - \frac{1}{2}t^2 - 2t + 5$

$t = ?$

$V = (\frac{1}{3}t^3 - \frac{1}{2}t^2 - 2t + 5)'$

$V = t^2 - t - 2;\ D = (-1)^2 - 4 * 1 * (-2)$

$o = t^2 - t - 2;$

$D = 1 + 8 + 9$

$t_1 = \frac{1-3}{2} = -1$ – *не может быть*

$1 + 3$

7.13. Исследовать на монотонность и экстремум функцию $y = x^3 - 3x$

1) О.О.Ф. $x \in R$ или $(-\infty; +\infty)$

2) $y' = (x^3 - 3x)' = 3x^2 - 3$

3) $y' = 0 \quad 3x^2 - 3 = 0 \quad x^2 - 1 = 0 \quad x_1 = 1 \quad x_2 = 2$ – критические точки

4)

при $x \in (-\infty; -1) \cup (1; +\infty) \Rightarrow$ функция возрастает

при $x \in (-1; 1) \Rightarrow$ функция убывает

x = -1 – точка максимума, причем $y_{max} = (-1)^3 - 3 * (-1) = -1 + 4 = 2$

x=1 – точка минимума, причем $y_{min} = 1^3 - 3 * 1 = 1 - 3 = -2$

ПРАКТИЧЕСКАЯ РАБОТА №15

«Нахождение производной функции».

I	II	III
1. $y=\frac{3}{x^3}-\frac{x}{4}+5$	1. $y=\frac{5x+\sqrt{x}}{2}$	1. $y=2x\sqrt[3]{x}+e^{-x}$
2. $y=\frac{3x}{5\sqrt[3]{x}}$	2. $y=x^2-\frac{1}{x}+\frac{x^3}{4}-1$	2. $y=x\ln x$
3. $y=(4x-1)(5-x)$	3. $y=\frac{7x}{2\sqrt[3]{x^2}}$	3. $y=\frac{3}{x^3}-\frac{x}{4}+7$
4. $y=4x^2\sqrt{x}$	4. $y=2x\operatorname{tg}5x$	4. $y=\frac{2x}{7\sqrt[3]{x}}-1$
5. $y=\frac{e^{3x}-e^{-x}}{4}$	5. $y=\frac{7x^2+7x-1}{6}$	5. $y=\frac{e^{-2x}+e^{3x}}{7}$
6. $y=\sqrt[3]{e^{2x+5}}$	6. $y=\frac{5x+2}{1-x^2}$	6. $y=\sqrt[4]{e^{1-x}}$
7. $y=\frac{2x-1}{x^2}$	7. $y=\sqrt{e^{x-1}}$	7. $y=(4x-1)(x^2-4)$
8. $y=x\sqrt[3]{x}+sin5x$	8. $y=\frac{x}{\sqrt{x}}-e^{-x}$	8. $y=e^{\sin 5x}-1$
9. $y=\ln\frac{1-x^2}{2x-1}$	9. $y=\cos^3 2x$	9. $y=\ln\sqrt[3]{sin3x}$
10. $y=\sin^2 5x$	10. $y=\ln\frac{x^2}{2x-1}$	10. $y=\frac{e^{-2x}+e^{x}}{5}$
11. $y=x^3\cos 2x$	11. $y=\frac{e^{x}-1}{5}$	11. $y=\ln\frac{3x-4}{1-x^2}$
12. $y=\ln\sqrt{tg3x}$	12. $y=(8x-1)^7$	12. $y=4\arccos 2x$
13. $y=\sqrt{2x-1}$	13. $y=\ln\sin^2 x$	13. $y=\sqrt[3]{2x^2+5}$
14. $y=\frac{2}{x^2}-e^{-x}+lnx$	14. $y=2x^2-\ln^2 x+e^{2x}$	14. $y=xe^{x}+5\sin 3x$
15. $y=\frac{e^{x}}{x}+arctg2x$	15. $y=5\arcsin 5x$	15. $y=4x^2\sqrt{x}-tg2x$

IV	V	VI
1. $y=\frac{2x}{7\sqrt{x}}+3e^{-x}$	1. $y=\frac{2}{3x}-\frac{x^2}{8}+5$	1. $y=3x\sqrt[3]{x^2}+5$
2. $y=\frac{4}{x^2}-\frac{\sqrt{x}}{5}-6x+1$	2. $y=2x^2\sqrt{x}-e^{4x}$	2. $y=\frac{2x^2}{5\sqrt{x}}$
3. $y=x^3\cos 7x$	3. $y=\frac{e^{-x}+5e^{3x}}{4}$	3. $y=\frac{2}{x^2}-\frac{x^3}{5}+5$
4. $y=\sqrt{e^{3x+5}}$	4. $y=\ln\frac{4-x}{x^3}$	4. $y=x^3\operatorname{tg}2x$
5. $y=5x\sqrt[3]{x^2}-sin2x$	5. $y=\sqrt[3]{e^{5x-1}}$	5. $y=\frac{2x^2-4x+5}{9}$

6. $y=5\arcsin 5x-\frac{1}{x}$	6. $y=(4x-1)(2x-1)$	6. $y=\sqrt{e^{1+2x^2}}$
7. $y=\frac{5x^2-6x+7}{2}$	7. $y=e^{tg2x}-1$	7. $y=*=(8x-1)^9$
8. $y=\ln\frac{5x+6}{2x^2-4x+1}$	8. $y=\sqrt[3]{4x^2}-5$	8. $y=\ln\frac{3x-1}{x^2}$
9. $y=\ln tg^2 5x$	9. $y=x^3\sin\frac{x}{3}$	9. $y=\ln\cos^2 2x$
10. $y=(4x^2-1)^7$	10. $y=\ln\frac{8x}{5x+1}$	10. $y=x^2e^{-x}$
11. $y=\frac{2x^2}{5\sqrt{x}}$	11. $y=\ln\sqrt[3]{sin2x}$	11. $y=\frac{2x^2-1}{4x+2}$
12. $y=\frac{4x^2-1}{x^3}$	12. $y=e^{2x}-1$	12. $y=6\operatorname{arctg}5x$
13. $y=xe^{5x}$	13. $y=\frac{2x-1}{4-x}$	13. $y=\frac{x\sqrt{x}}{5}+cos2x$
14. $y=\sqrt{cos2x}$	14. $y=4-\operatorname{arctg}\sqrt{x}$	14. $y=e^{\sin 3x}$
15. $y=2x-e^{\cos 4x}$	15. $y=\sin 2x-5$	15. $y=e^{\sin 3x}$

ПРАКТИЧЕСКАЯ РАБОТА №16

«Нахождение дифференциалов функции».

Вариант 1

1) Найти дифференциал первого порядка функций:

а) $y = a^x * e^x$; б) $y = \frac{e^x + 1}{e^x}$

2) Найти дифференциал второго порядка функции:

$y = a^{3x}$

Вариант 2

1) Найти дифференциал первого порядка функций:

а) $y = e^x * \cos x$; б) $y = (x^3 - 2)^4$

2) Найти дифференциал второго порядка функции:

$y = \ln tg\, 2x$

ПРАКТИЧЕСКАЯ РАБОТА №17

«Исследование функции с помощью производной. Полная схема исследования функции».

I	II	III
1. $y=\frac{1}{3}x^3-\frac{5}{2}x^2+6x$	1. $y=x^3-3x^2+4$	1. $y=-x^3+3x^2-2$
2. $y=x^4-4x^2+4$	2. $y=x^4-8x^2+16$	2. $y=-x^4+2x^2+3$

IV	V	VI
1. $y=\frac{1}{3}x^3-x^2-3x+\frac{1}{2}$	1. $y=x^3-6x^2+9x-3$	1. $y=x^3+9x^2+24x+12$
2. $y=x^4-x^2$	2. $y=x^4+8x^2+5$	2. $y=x^4-8x^2-9$

КОНТРОЛЬНАЯ РАБОТА №4

«Дифференциальное исчисление».

Вариант 1

Найти производную функции:

$$1)\ y=\frac{4}{x^2}-\frac{\sqrt{x}}{5}-6x+1;2)\ y=x^3\cdot\cos 7x;3)y=\ln\frac{1-x^2}{2x-1}$$

Вариант 2

Найти производную функции:

$$1)\ y=\frac{2}{x^2}-\frac{\sqrt[3]{x}}{5}+5;2)\ y=x^3\cdot\sin\frac{x}{3};3)y=\ln\frac{8x}{5x+1}$$

ТЕМА: ***«ИНТЕГРАЛЬНОЕ ИСЧИСЛЕНИЕ»***

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

I. **Что необходимо знать по теме:**

a. Неопределённый интеграл

b. Определение:

1) Если F`(x)=f(x), то F(x) называют первообразной функции f(x).
2) F(x)+C, где C – число, все первообразные f(x) или неопределённый интеграл. Это записывают в виде формулы: $\int f(x)dx = F(x) + C$, $\int$ - знак неопределённого интеграла.

c. Таблица неопределённых интегралов:

Для простых функций	Для сложнолинейных функций
1. $\int(f_1(x) \pm f_2(x))dx = \int f_1(x)dx \pm \int f_2(x)dx$	-
2. $\int rf(x)dx = r\int f(x)dx$	-
3. $\int rdx = rx + c$	-
4. $\int x^n dx = \frac{x^{n+1}}{n+1}, n \neq -1$	4. $\int(rx+b)^n dx = \frac{(rx+b)^{n+1}}{r(n+1)} + c, n \neq -1$
5. $\int x^{-1}dx = \ln\|x\| + c$	5. $\int(rx+b)^{-1}dx = \frac{\ln\|rx+b\|}{r} + c$
6. $\int a^x dx = \frac{a^x}{\ln a} + c$	6. $\int a^{rx+b}dx = \frac{a^{rx+b}}{r\ln a} + c$
7. $\int e^x dx = e^x + c$	7. $\int e^{rx+b}dx = \frac{e^{rx+b}}{r} + c$
8. $\int \cos x\, dx = \sin x + c$	8. $\int \cos(rx+b)\, dx = \frac{\sin(rx+b)}{r} + c$
9. $\int \sin x\, dx = -\cos x + c$	9. $\int \sin(rx+b)\, dx = -\frac{\cos(rx+b)}{r} + c$
10. $\int \frac{dx}{cos^2 x} = tg\, x + c$	10. $\int \frac{dx}{cos^2(rx+b)} = \frac{tg(rx+b)}{r} + c$
11. $\int \frac{dx}{sin^2 x} = -ctg\, x + c$	11. $\int \frac{dx}{sin^2(rx+b)} = -\frac{ctg(rx+b)}{r} + c$
12. $\int \frac{dx}{\sqrt{1-x^2}} = \arcsin x + c$	12. $\int \frac{dx}{\sqrt{a^2-x^2}} = arcsin\frac{x}{a} + c$
13. $\int \frac{dx}{1+x^2} = arctg\, x + c$	13. $\int \frac{dx}{a^2+x^2} = \frac{1}{a} arcsin\frac{x}{a} + c$

1. **Определённый интеграл**
2. $\int_a^b f(x)dx = [F(b)] - [F(a)]$ – формула Ньютона-Лейбница.
3. Применение интеграла при вычислении площадей плоских фигур:

3.1. y=f(x)≥ 0, x=a, x=b, y=0 – криволинейная трапеция (геометрический смысл определённого интеграла)

$$S_{к.тр} = \int_a^b f(x)dx$$

3.2. y=f(x)≤ 0, x=a, x=b, y=0 – криволинейная трапеция

$$S_{к.тр.}=\int_a^b[-f(x)]\,dx$$

2.3. $y_1=f_1(x)$, $y_2=f_2(x)$ ($f_1(x)\geq f_2(x)$) – площадь между двумя кривыми линиями

$$S=\int_a^b[f_1(x)-f_2(x)]dx$$

Примечание: Формула не зависит от того, в каких частях расположен рисунок, главное от ($f_1(x)$) верхней линии отнять ($f_2(x)$) – нижнюю линию.

3. Применение интеграла при вычислении объёмов тел вращения

$$V_{т.в.}=\pi\int_a^b[f(x)]^2dx$$

4. Примечание: При построении графика функции $y=ax^2+bx+c$ (параболы) необходимо вычислить её вершину по формуле: $x_0=\frac{-b}{2a}$ и $y_0=ax_0^2+bx_0+c$.

II. Решение примеров и задач на вычисление интегралов

1. Так как в таблице отсутствуют формулы вычисления интегралов действий корней, умножения, деления, а также некоторых основных функций, например: tgx, ctgx, $\log_a x$ и т. д., и всех сложных функций (кроме сложнолинейных), то прежде, чем обратиться к таблице, необходимо (если это возможно) избавиться от этих действий и функций (т. е. подогнать к таблице).
2. Так как не всегда возможно подогнать к таблице, то существуют различные способы вычисления интегралов:
   a. непосредственное интегрирование;
   b. метод подстановки;
   c. интегрирование по частям и т. д.
3. Мы будем использовать только один способ: «Непосредственное интегрирование» (это означает: подогнать функцию под табличную).

Внимание! ***Схема решения:***

1) В записи функций не должно быть *корней, дробей* (кроме таких, как последние четыре из таблицы), *произведений*;
2) Прежде чем воспользоваться таблицей в записи функций найти *последнее действие* и по его названию найти формулу из таблицы.

4. Примеры:

1) $\int(2x-1)(4-x)dx = \int(8x-2x^2-4)dx = \int 8xdx - \int 2x^2dx - \int 4dx = \frac{9x^2}{2} - \frac{2x^3}{3} - 4x + C$

2) $\int \frac{(x-1)^2}{5}dx = \frac{1}{5}\int(x^2-2x+1)dx = \frac{1}{5}\left(\frac{x^3}{3} - \frac{2x^2}{2} + x\right) + C = \frac{x^3}{15} - \frac{x^2}{5} + \frac{x}{5} + C$

3) $\int \frac{x\sqrt{x}}{2}dx = \frac{1}{2}\int x * x^{\frac{1}{2}}dx = \frac{1}{2}\int x^{\frac{3}{2}}dx = \frac{x^{\frac{5}{2}}}{2*\frac{5}{2}} + C = \frac{x^{\frac{5}{2}}}{5} + C$

4) $\int \frac{4x^2dx}{3\sqrt[3]{x}} = \frac{4}{3}\int \frac{x^2dx}{x^{\frac{1}{3}}} = \frac{4}{3}\int x^{\frac{5}{3}}dx = \frac{4*x^{\frac{8}{3}}}{3*\frac{8}{3}} + C = \frac{x^{\frac{8}{3}}}{2} + C$

5) $\int \frac{x^2+4x-5}{x^2}dx = \int x^{-2}(x^2+4x-5)dx = \int 1 + 4x^{-1} - 5x^{-2})dx = x + 4\ln|x| - \frac{5x^{-1}}{-1} + C$

6) $\int \frac{x^3+8}{x+2}dx = \int \frac{(x+2)(x^2-2x+4)}{x+2}dx = \frac{x^3}{3} - \frac{2x^2}{2} + 4x + C$

7) $\int \sqrt[3]{e^{2x-1}}dx = \int e^{\frac{2x}{3}-\frac{1}{3}}dx = \frac{e^{\frac{2x}{3}-\frac{1}{3}}}{\frac{2}{3}} + C$

8) $\int\left(e^{-x} + sin5x + \frac{1}{cos^2 2x}\right)dx = \frac{e^{-x}}{-1} + \frac{-cos5x}{5} + \frac{tg2x}{2} + C$

9) $\int \frac{4dx}{\sqrt{1-x^2}} = 4\arcsin x + C$

10) $\int \frac{dx}{\sqrt{4-x}} = \int(4-x)^{-\frac{1}{2}}dx = \frac{(4-x)^{\frac{1}{2}}}{-1*\frac{1}{2}} + C$

11) $\frac{5dx}{5x+1} = \int(5x+1)^{-1}dx = \frac{\ln|5x+1|}{5} + C$

12) $\int(3x-1)^4\,dx = \frac{(3x-1)^5}{3*5} + C$

13) $\int \frac{3dx}{sin^2\frac{x}{3}} = -\frac{3ctg\frac{1}{3}x}{\frac{1}{3}} + C = -9ctg\frac{x}{3} + C$

14) $y = x^3$; x=1, x=2, y=0. Найти: S - ?

$S=S_1+S_2$

$S_1=\int_{-1}^{0}[-x^3]dx = \left[-\frac{x^4}{4}\right]_{-1}^{0} = 0 - \left[-\frac{(-1)^4}{4}\right] = \frac{1}{4}$ (ед^2)

$S_2=\int_0^2 x^3dx = \left[\frac{x^4}{4}\right]_0^2 = \left[\frac{16}{4}\right] - 0 = 4$ (ед^2)

$S=4+\frac{1}{4} = 4{,}25$ (ед^2)

15) $y = x^2 - 4$; y=0. Найти: S - ?

$x_0=-\frac{b}{2a}, x_0 = \frac{0}{2} = 0, y = 0 - 4 = -4$ (вершина)

a=1 – ветви вверх

x^2-4=0, x=±2 – корни

$$S=\int_{-2}^{2}[-(x^2-4)]dx=\left[4x-\frac{x^3}{3}\right]_{-2}^{2}=\left[8-\frac{8}{3}\right]-\left[-8+\frac{8}{3}\right]=8-\frac{8}{3}+8-\frac{8}{3}=2\left(8-\frac{8}{3}\right)=2\left(8-\frac{8}{3}\right)=\frac{2(24-8)}{3}=\frac{32}{3}\ (\text{ед}^2)$$

Примечание: Чтобы найти a и b, когда они не заданы, надо приравнять правые части заданных функций ($y=x^2-4$ и $y=0 \Rightarrow x^2-4=0$, $x=\pm 2$

16) $y=x^2-1$, $y=0$. Найти: V - ?

Нет a и b $\Rightarrow x^2-1=0$, $x=\pm 1$

$$V=\pi\int_{-1}^{1}[(x^2-1)]^2dx=\pi\int_{-1}^{1}(x^4-2x^2+1)dx=\left[\frac{x^5}{5}-\frac{2x^3}{3}+x\right]_{-1}^{1}\pi=\left[\frac{3x^5-10x^3+15}{15}\right]_{-1}^{1}\pi=\left[\frac{3-10+15}{15}\right]\pi-\left[\frac{-3+10-15}{15}\right]\pi=\frac{8}{15}\pi+\frac{8}{15}\pi=\frac{16\pi}{15}\ (\text{ед}^3)$$

ПРАКТИЧЕСКАЯ РАБОТА №18

«Интегрирование подстановкой и по частям».

I	II	III
1) $\int(3x-2)(x^2+5)dx$	1) $\int\frac{2x^2}{7\sqrt{x}}dx$	1) $\int\frac{3x\sqrt{x}}{5}dx$
2) $\int(\frac{2}{x}+\frac{x^2}{7}-2)dx$	2) $\int\frac{x^3-2x-x+7}{x}dx$	2) $\int\left(3-\frac{5}{x^2}+\frac{x}{2}+x\right)dx$
3) $\int\frac{2x^2-5x+4}{x^2}dx$	3) $\int x^2(3x-1)dx$	3) $\int(5x-1)(2-3x^2)dx$
4) $\int\frac{2x}{7\sqrt[3]{x}}dx$	4) $\int\frac{x^3-64}{x^2+4x+16}dx$	4) $\int\frac{6x^2-5x+7}{x^3}dx$
5) 5) $\int(4x^2*\sqrt{x}-e^{-x}dx$	5) $\int\left(\frac{1}{x^2}+\frac{2x}{5}+5x-1)dx\right)$	5) $\int\left(\frac{x^2}{4\sqrt[3]{x}}+e^{-}\right)dx$
6) 6) $\int\sqrt{e^{3x+5}}dx$	6) $\int\left(x\sqrt[3]{x^2}+sin\frac{x}{2}\right)dx$	6) $\int\sqrt{e^{1-3x}}dx$
7) $\int x(x-1)^2dx$	7) $\int\frac{(x+2)^2}{5}dx$	7) $\int 3x(2-x)^2dx$
8) $\int\left(\frac{1}{1+x^2}-\sin5x\right)dx$	8) $\int\sqrt[3]{e^{1-x}}dx$	8) $\int\left(cos\frac{x}{3}-\frac{2}{\sqrt{1-x^2}}\right)dx$
9) $\int\frac{x^3-1}{x^2+x+1}dx$	9) $\int\sqrt[3]{4x-1}\,dx$	9) $\int\frac{x^3-1}{x-1}dx$
10) $\int\frac{3dx}{\sqrt{1-2x}}$	10) $\int\frac{3dx}{1-2x}$	10) $\int\frac{dx}{\sqrt[3]{3x+7}}$
11) $\int(3x-1)^{10}dx$	11) $\int\left(\frac{e^{3x}-e^{-x}}{5}\right)dx$	11) $\int(1-2x)^7dx$
12) $\int\frac{dx}{3x+7}$	12) $\int cos(3x-\pi)dx$	12) $\int\frac{dx}{7-x}$
13) $\int\frac{dx}{sin^2\frac{x}{2}}$	13) $\int\left(\frac{1}{x^2}-\frac{2}{\sqrt{1-x^2}}\right)dx$	13) $\int\frac{dx}{sin^24x}$

IV	V	VI

1) $\int \frac{\sqrt{x}}{4x^2}dx$	1) $\int(5x-3)(1-4x)dx$	1) $\int \frac{3\sqrt[3]{x}}{x}dx$
2) $\int \frac{1-x+3x^2}{x}dx$	2) $\int\left(2-\frac{4}{x^3}+\frac{x}{2}\right)dx$	2) $\int \frac{2x^3+4x^2-5}{x^4}dx$
3) $\int x^3(4x^2-2x-1)dx$	3) $\int \frac{4x^3-2x^2+5x-1}{x}dx$	3) $\int x(4x^2-1)dx$
4) $\int \frac{x^3-64}{x+4}dx$	4) $\int \frac{x^2}{5\sqrt{x}}dx$	4) $\int \frac{x^3+8}{x^2-2x+4}dx$
5) $\int\left(x-\frac{2}{x^3}+\frac{x}{7}+5\right)dx$	5) $\int\left(\frac{x\sqrt[3]{x^2}}{2}+2e^{-x}\right)dx$	5) $\int\left(\frac{1}{x^2}-\frac{4x^3}{5}+6\right)dx$
6) $\int(2x^2\sqrt[3]{x^2}+cos3x)dx$	6) $\int(x^2\sqrt{x}-\sqrt{e^{x-1}})dx$	6) $\int\left(\frac{x\sqrt{x}}{5}+e^{2x}\right)dx$
7) $\int \frac{(3x-1)^2}{7}dx$	7) $\int x^2(x-1)^2dx$	7) $\int \frac{(4x+1)^2}{3}dx$
8) $\int \sqrt{e^{2x+5}}dx$	8) $\int\left(\frac{2}{\sqrt{1-x^2}}-cos\frac{x}{3}\right)dx$	8) $\int \sqrt{e^{2-3x}}dx$
9) $\int \sqrt{4x+7}dx$	9) $\int \frac{x^3+1}{x+1}dx$	9) $\int \sqrt[3]{x+3}dx$
10) $\int \frac{dx}{4x-1}$	10) $\int \frac{dx}{\sqrt[3]{2x-1}}$	10) $\int \frac{3dx}{1-5x}$
11) $\int \frac{2x^2-4x+7}{7}dx$	11) $\int \sqrt[4]{3-8x}dx$	11) $\int \frac{e^{-x}+e^{4x}}{4}dx$
12) $\int sin\left(3x-\frac{\pi}{3}\right)$	12) $\int \frac{4dx}{1-x}$	12) $\int \cos(x-\pi)dx$
13) $\int\left(e^{-x}+\frac{5}{\sqrt{1-x^2}}\right)dx$	13) $\int \frac{5dx}{cos^2 4x}$	13) $\int\left(cos\frac{x}{5}+\frac{1}{\sqrt{4-x^2}}\right)dx$

ПРАКТИЧЕСКАЯ РАБОТА №19

«Интегрирование простейших рациональных дробей. Интегрирование некоторых видов иррациональностей».

Вариант 1

Найти неопределенный интеграл

1) $\int (5x-3)(1-4x)dx$; $\int \frac{x^3+64}{x+4}\,dx$; $\int \sin(3x-\pi)dx$

2) $\int (1-2x)^7 dx$; $\int \frac{dx}{3x+7}$

3) $\int (1-x)\sin x\,dx$; $\int e^x \cos x\,dx$

Вариант 2

Найти неопределенный интеграл

1) $\int x^3(4x^2-2x-1)dx$; $\int \frac{4x^3-2x^2+5x-1}{x}\,dx$; $\int \frac{\mathrm{dx}}{\sin^2 2x}$

2) $\int \sqrt{4x+7}dx$; $\int \frac{3\,dx}{1-5x}$

3) $\int (2+x)\cos x\,dx$; $\int e^x \cdot \sin x\,dx$

ПРАКТИЧЕСКАЯ РАБОТА №20

«Вычисление определённых интегралов».

I	II	III
1. $\int_{-1}^{1}(3x-1)(1-x^2)dx$	1. $\int_0^1 \frac{dx}{1+x^2}$	1. $\int_1^2 (4x-1)(1-x)dx$
2. $\int_{\pi/2}^{\pi} \frac{dx}{3\cos^2\frac{x}{3}}$	2. $\int_1^2 \frac{x^2-x-1}{x}dx$	2. $\int_0^1 \frac{dx}{3x+1}$
3. $\int_0^1 \frac{dx}{\sqrt{1-x^2}}$	3. $\int_0^{\pi/4} \cos(2x-\frac{\pi}{4})dx$	3. $\int_0^{\sqrt{3}/3} \frac{dx}{1+x^2}$

IV	V	VI
1. $\int_{\pi/3}^{\pi/2} \frac{3dx}{2\cos^2\frac{x}{2}}$	1. $\int_{-1}^{2}(3x-5)(7x+1)dx$	1. $\int_{\pi/6}^{\pi/4} \frac{dx}{\sin^2 2x}$
2. $\int_{-1}^{1} \frac{2x^2-4x+5}{x^2}dx$	2. $\int_0^{\pi/2}\left(\cos\left(2x-\frac{\pi}{2}\right)\right)dx$	2. $\int_1^2 \frac{x^3+4x^2+5}{x^2}dx$
3. $\int_0^1 \frac{dx}{1+x^2}$	3. $\int_{\sqrt{3}/3}^{\sqrt{3}} \frac{dx}{1+x^2}$	3. $\int_0^1 \frac{dx}{\sqrt{1-x^2}}$

ПРАКТИЧЕСКАЯ РАБОТА №21

«Вычисление площадей плоских фигур. Вычисление объёмов тел вращения».

I	II	III
1. $y=x^2$, $x=0$, $x=2$, $y=0$ Найти: S - ?	1. $y=x^3$, $x=-2$, $x=-1$, $y=0$ Найти: S - ?	1. $y=-x^2$, $x=1$, $x=2$, $y=0$ Найти: S - ?
1. $y=x^2-4x-5$, $y=0$ Найти: S - ?	2. $y=-x^2+9$, $y=0$ Найти: S - ?	2. $y=-x^2-2x+8$, $y=0$ Найти: S - ?
3. $y=x^2-8x+16$, $y=6-x$ Найти: S - ?	3. $y=-x^2-7x+8$, $y=2x+16$ Найти: S - ?	3. $y=x^2-6x+9$, $y=3x-9$ Найти: S - ?
4. $y=x^2$, $y=2x+4$ Найти: S - ?	4. $y=x^2$, $y=3x$ Найти: S - ?	4. $y=x^2$, $y=x$ Найти: S - ?
5. $y=x^3$, $x=1$, $x=2$, $y=0$ Найти: V - ?	5. $y=x^2$, $x=1$, $x=2$, $y=0$ Найти: V - ?	5. $y=\sqrt{7}x^3$, $x=0$, $x=1$, $y=0$ Найти: V - ?
6. $y=x^2-2x$, $y=0$ Найти: V - ?	5. $y=x^2-1$, $y=0$ Найти: V - ?	6. $y=3x-x^2$, $y=0$ Найти: V - ?
7. $y=x^2-4$, $x=0$, $x=1$, $y=0$ Найти: V - ?	6. $y=x^2-3x$, $x=1$, $x=2$, $y=0$ Найти: V - ?	7. $y=x^2-9$, $x=-1$, $x=1$, $y=0$ Найти: V - ?

IV	V	VI
1. $y=x^3$, $x=0$, $x=2$, $y=0$ Найти: S - ?	1. $y=-x^2$, $x=1$, $x=2$, $y=0$ Найти: S - ?	1. $y=-x^3$, $x=1$, $x=2$, $y=0$ Найти: S - ?
2. $y=x^2-x-6$, $y=0$ Найти: S - ?	2. $y=-x^2+x+6$, $y=0$ Найти: S - ?	2. $y=-x^2+2x+3$, $y=0$ Найти: S - ?
3. $y=-x^2-10x-16$, $y=x+2$ Найти: S - ?	3. $y=x^2-8x+18$, $y=-2x+18$ Найти: S - ?	3. $y=3x-x^2$, $y=5x-8$ Найти: S - ?
4. $y=x^2$, $y=2x$ Найти: S - ?	4. $y=x^2$, $y=-x$ Найти: S - ?	4. $y=x^2$, $y=-3x$ Найти: S - ?
5. $y=\sqrt{5}x^2$, $x=0$, $x=1$, $y=0$ Найти: V - ?	5. $y=5x^2$, $x=1$, $x=2$, $y=0$ Найти: V - ?	5. $y=7x^3$, $x=0$, $x=1$, $y=0$ Найти: V - ?
6. $y=x^2-4$, $y=0$ Найти: V - ?	6. $y=x-x^2$, $y=0$ Найти: V - ?	6. $y=1-x^2$, $y=0$ Найти: V - ?

7. $y=4x-x^2$, $x=1$, $x=2$, $y=0$ Найти: V - ?	7. $y=x^2-16$, $x=-1$, $x=1$, $y=0$ Найти: V - ?	7. $y=3x-x^2$, $x=1$, $x=2$, $y=0$ Найти: V - ?

КОНТРОЛЬНАЯ РАБОТА №5
«Интегральное исчисление».

Вариант №1

1. Вычислите неопределенные интегралы:

$$\int (4x^3-6x^2-4x+3)dx$$

$$\int \frac{x^4-xe^x+6}{x}dx$$

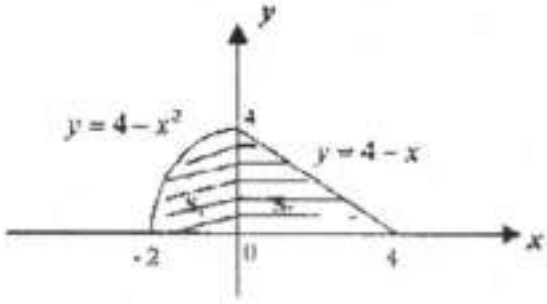
2. Вычислите определенные интегралы:

$$\int_{-1}^{0} (x^3+2x)dx$$

$$\int_{4}^{5} (4-x)^3 dx$$

3. Найдите площадь фигуры, ограниченной линиями $y=-3x^2$, $y=0$, $x=1$ и $x=2$.

4. Вычислите площадь фигуры, ограниченной линиями.



Вариант №2

1. Вычислите неопределенные интегралы:

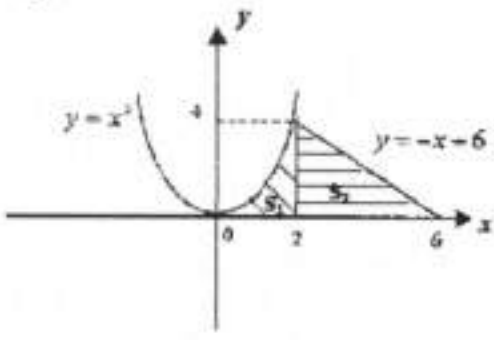
$$\int (x^{-4}-x^{-3}-3x^{-2}+1)dx$$

$$\int x^4(x-1)dx$$

2. Вычислите определенные интегралы:

$$\int_{-2}^{3} (4x^3-3x^2+2x+1)dx$$

$$\int_{-1}^{2} \left(\frac{4}{3}x^3-\frac{3}{4}x^2+5\right)dx$$



3. Найдите площадь фигуры, ограниченной линиями $y=x^2-x-2$, $y=0$

4. Вычислите площадь фигуры, ограниченной линиями

Вариант №3

1. Вычислите неопределенные интегралы:

$$\int\left(\frac{4}{3}x^3-\frac{3}{4}x^2+5\right)dx$$

$\int x^2(1+5x)dx$

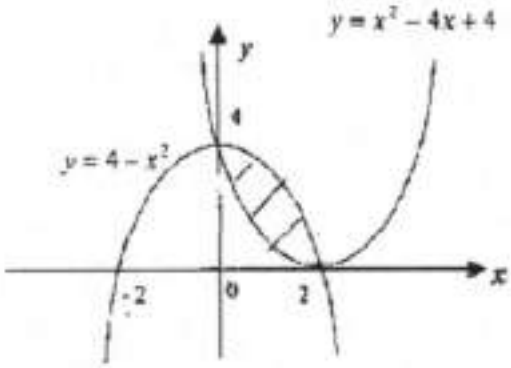
2. Вычислите определенные интегралы:

$\int_{-1}^{3}(4x^3-6x^2-4x+3)dx$

$\int_{-1}^{2}(x^2+2x+1)dx$

3. Найдите площадь фигуры, ограниченной линиями $y=x^2-8x+7,\ y=0$

4. Вычислите площадь фигуры, ограниченной линиями.



Вариант №4

1. Вычислите неопределенные интегралы:

$\int(2x-e^x-1)dx$

$\int 3(2x^2-1)^2dx$

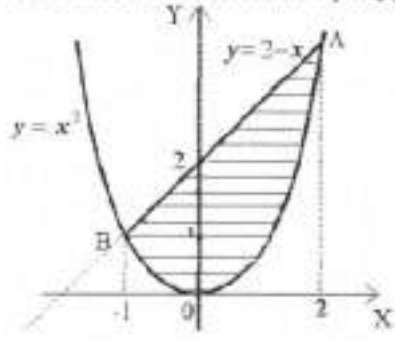
2. Вычислите определенные интегралы:

$\int_{0}^{3}(3+2x)^3dx$

$\int_{-2}^{2}(x^3+2x+5)dx$

3. Найдите площадь фигуры, ограниченной линиями $y=-2x^2,\ y=0,\ x=1,\ x=2$

4. Вычислите площадь фигуры, ограниченной линиями.



ТЕМА «*ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ*»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

I. Что необходимо знать по теме:

a. Определение: Дифференциальным уравнением называется уравнение, связывающее между собой независимую переменную x, искомую функцию y и её производные или дифференциалы.

Обозначение: $F(x, y, y')=0$, в общем виде $F(x, y, y', y'', \ldots, y^{(n)})=0$

b. Порядком дифференциального уравнения называется порядок старшей производной, входящей в уравнение.

c. Решением дифференциального уравнения называется функция, которая обращает это уравнение в тождество.

d. Дифференциальное уравнение всегда имеет бесчисленное множество решений, записанных в виде формулы, которая носит название общего решения.

Общее решение содержит столько независимых произвольных постоянных, каков порядок дифференциального уравнения.

e. Частным решением дифференциального уравнения называется решение, полученное из общего при различных числовых значениях произвольных постоянных. Значения произвольных постоянных находятся при определённых начальных значениях аргумента и функции.

f. Решение некоторых видов дифференциальных уравнений

СХЕМЫ РЕШЕНИЯ:

I порядка	II порядка
6.1. Простейшие Общий вид: $y'=f(x)$ Схема решения: 1. Проинтегрировать правую часть. $y=\int f(x)dx$	6.1. Простейшие Общий вид: $y''=f(x)$ Схема решения: 1. Проинтегрировать правую часть. $y'=\int f(x)dx$ 2. Проинтегрировать результат первого действия $y=\int(\int f(x)dx)dx$
6.2. *С разделяющимися переменными* Общий вид: $y'=f(x)*\varphi(y)$ Схема решения: 1. Записать через дифференциал $dy=f(x)\varphi(y)dx$ 2. Разделить переменные $\frac{dy}{\varphi(y)}=f(x)dx$ 3. Проинтегрировать левую и правую часть $\int \frac{dy}{\varphi(y)}=\int f(x)dx$	6.2. *Однородные с постоянными коэффициентами* Общий вид: $ay''+by''+cy=0$ $a, b, c \in R$. $a\neq 0$ Схема решения: 1. Составить характеристическое уравнение и решить его $at^2+bt+c=0$ 2. В зависимости от количества и вида корней записать ответ: 2.1. $D>0$, $t_1\neq t_2$ => Ответ: $y=C_1e^{t_1x}+y=C_2xe^{t_2x}$ 2.2. $D=0$, $t_0=0$ => Ответ: $y=C_1e^{t_0x}+y=C_2xe^{t_0x}$ 2.3. $D<0$, $t_1, t_2=\alpha\pm\beta i$ => Ответ: $Y=e^{ax}(C_1\cos\beta x+C_2\sin\beta x)$

	Примечание: $\alpha \pm \beta i$ – алгебраическая форма записи комплексного числа, где α и $\beta \in R$, а i – мнимая единица, её главное свойство $i^2=-1$.

III. Примеры решения дифференциальных уравнений

a) $xy'-1=x^2$

Приведём к общему виду:

$y'=\frac{x^2+1}{x}$ – простейшее первого порядка. Применяем схему

$$y=\int \frac{x^2+1}{x}dx = \int x^{-1}(x^2+1)dx = \int (x+x^{-1})dx = \frac{x^2}{2} + \ln|x| + C$$

Ответ: $y=\frac{x^2}{2} + \ln|x| + C$

b) $y''=x-1$ – интегрируем правую часть

$y'=\int (x-1)dx = \frac{x^2}{2} - x + C_1$ – интегрируем ещё раз

$$y=\int (\frac{x^2}{2} - x + C_1)dx = \frac{x^3}{6} - \frac{x^2}{2} + C_1 x + C_2$$

Ответ: $y= \frac{x^3}{6} - \frac{x^2}{2} + C_1 x + C_2$

c) $x^2y'-2xy=3y$

Общий вид: $y'=\frac{y(3+2x)}{x^2}$ - с разделяющимися переменными

$dy=\frac{y(3+2x)}{x^2}dx$

$$\frac{dy}{y} = \frac{3+2x}{x^2}dx$$

$$\int \frac{dy}{y} = \int \frac{3+2x}{x^2}dx$$

$$\int y^{-1}dy = \int x^{-2}(3+2x)dx => \ln|y| = \int (3x^{-2} + 2x^{-1})dx => \ln|y| = \frac{3x^{-1}}{-1} + 2\ln|x| + C$$

Функция найдена не явно. Ответ можно так и оставить.

d) $y''+2y'=0$
$t^2+2t=0$
$t=0, t=-2$
Ответ: $y=C_1e^{0x}+C_2e^{-2x}$, т. е. $y=C_1+C_2e^{-2x}$

e) $y''+6y'+9y=0$
$t^2+6t+9=0$
$t_0=-\frac{6}{2}=-3$

Ответ: $y=C_1e^{-3x}+C_2xe^{-3x}$

f) $y''+4y'+8y=0$

$t^2+4t+8=0$

$D=16-32=-16=16i^2$

$t_{1,2}=\frac{-4\pm\sqrt{16i^2}}{2}=\frac{-4\pm4i}{2}=\underset{\alpha}{-2}\pm\underset{\beta}{2i}$

Ответ: $y=e^{-2x}(C_1\cos2x+C_2\sin2x)$

g) $y''-y=0$

$t^2-1=0$

$t_{1,2}=\pm1$

Ответ: $y=C_1e^{x}+C_2e^{-x}$

h) $xdy-ydx=0 \Rightarrow xdy=ydx \Rightarrow \frac{dy}{y}=\frac{dx}{x} \Rightarrow \int\frac{dy}{y}=\int\frac{dx}{x}$

Ответ: $\ln|y|=\ln|x|+C$

ПРАКТИЧЕСКАЯ РАБОТА № 22

«Решение дифференциальных уравнений первого порядка с разделяющимися переменными, однородных дифференциальных уравнений первого порядка, линейных дифференциальных уравнений первого порядка».

ПРАКТИЧЕСКАЯ РАБОТА № 23

«Решение линейных однородных дифференциальных уравнений второго порядка с постоянными коэффициентами, линейных неоднородных дифференциальных уравнений второго порядка с постоянными коэффициентами».

I	II	III
1. $y'=2x-e^{-x}$	1. $\frac{y'}{x^2}+3x=x^2+1$	1. $2y'-x=e^{2x}$
2. $\frac{y''}{x}-1=x^2$	2. $(1+x^2)y'-1=0$	2. $\frac{y''}{x^2}+x=5$
3. $\sqrt{1-x^2}y'-5=0$	3. $xy''-x^4=3x^5$	3. $y'=5\sin\frac{x}{2}+x$
4. $3y''-\sin5x=3$	4. $y''-5=\cos4x$	4. $x^4y''-1=5x$
5. $\frac{2y'}{x^2}=4x-1$	5. $e^{-x}y'-4=0$	5. $(1+y^2)dx-x^3dy=0$
6. $x^2y''-1=x^4$	6. $2y''=e^{2x}+1$	6. $(1+x^2)y'-4=0$
7. $y'-y+2=0$	7. $(1+x)dx=(1-x)dy$	7. $3y''-x=e^{-x}$
8. $xy'-2xy^2=y^2$	8. $\frac{dx}{\sqrt{x}}=\frac{3dy}{\sqrt{y}}$	8. $3y'-7=2y$
9. $y^3dx-x^2dy=0$	9. $y'+2y-3=0$	9. $3\sqrt{x}dy=(2xy+3y)dx$
10. $x^2dy+(2y+1)dx=0$	10. $y'-5xy=x$	10. $xy'-1+2y=0$
11. $2(xy+y)dx=xdy$	11. $(xy^2+x)dy=(y-x^2y)dx$	11. $x^2dy=(2xy+3y)dx$
12. $y\sin xdx-3dy=0$	12. $xdx-\cos5xydy=0$	12. $(y-y\sin3x)dx+dy=0$
13. $y''+3y'+2y=0$, $x=0, y=-1, y'=3$	13. $y''-9y=0$, $x=0, y=0, y'=6$	13. $y''+2y'-8y=0$, $x=0, y=4, y'=-4$
14. $y''-10y'+25y=0$	14. $y''+3y'-4y=0$	14. $y''-5y'=0$

15. $y''-6y'+45y=0$	15. $y''+4y'=0$	15. $y''-16y=0$
16. $y''-3y'=0$	16. $y''-14y'+49y=0$	16. $y''-4y'+5y=0$
17. $y''-9y=0$	17. $y''+2y'+5y=0$	17. $y''-8y'+16y=0$
18. $y''+4y=0$	18. $y''+y=0$	18. $y''+9y=0$

IV	V	VI
1. $\frac{y'}{x}+1=x^3$	1. $2y'-4x=e^{2x}$	1. $\frac{y'}{x^3}-3x=5$
2. $xy''-x^3=5x$	2. $y''+x=\cos 5x$	2. $x^4y''-5=6x$
3. $\frac{y''}{\sin 3x}=1$	3. $\frac{y'}{x+1}=5x$	3. $y''-e^{-x}=2x-\sin 3x$
4. $\sqrt{1-x^2}y'=3$	4. $\frac{y''}{x}=6x-1$	4. $(1+x^2)y'=7$
5. $y''=e^{2x}-\cos x$	5. $\sqrt{4-x^2}y'=5$	5. $\sqrt[3]{x-1}y''=1$
6. $y'-3=\sqrt[3]{2-5x}$	6. $y''=(2x-1)^3-1$	6. $e^{-x}y'-5=0$
7. $y'+3y=xy$	7. $xy'-5xy=6y$	7. $5y'+2xy=4y$
8. $y'-3y=7$	8. $y'-y=7$	8. $y'-3=4y$
9. $(y+xy)dx+(x-xy)dy=0$	9. $(x^2+4x^2y)dy=(4x-1)dx$	9. $(2y+x^2y)dx=(3x-2xy)dy$
10. $x^2dy=\sqrt{1-y^2}dx$	10. $\sqrt[3]{x}dy-\sqrt{x}dx=0$	10. $x^3dy-(1+y^2)dx=0$
11. $\frac{dy}{\sqrt{x}}=\frac{dx}{\sqrt[3]{y^2}}$	11. $y\sin 2xdx-(2y-1)dy=0$	11. $\frac{dy}{x^2}=\frac{dx}{\sqrt{y}}$
12. $y\sin 3xdx-(1-y)dy=0$	12. $x^3dy=ydx$	12. $x\cos ydy+(2-3x)dx=0$

13. y''-3y'=0, x=0, y=1, y'=-1	13. y''-2y'-8y=0, x=0, y=4, y'=10	13. y''-3y'+2y=0, x=0, y=2, y'=3
14. y''+y'-6y=0	14. y''-6y'+9y=0	14. y''-64y=0
15. y''-2y'+50y=0	15. y''-4y'+8y=0	15. y''+11y'=0
16. y''+10y'+25y=0	16. y''-49y=0	16. y''+8y'+16y=0
17. y''-y=0	17. y''+5y'=0	17. y''+4y'+29y=0
18. y''+16y=0	18. y''+25y=0	18. y''+49y=0

КОНТРОЛЬНАЯ РАБОТА №6

«Дифференциальные уравнения».

Вариант 1

Решить дифференциальное уравнение:

$$1)(1+y)dx=(1-x)dy;\ 2)\frac{dy}{dx}=y+1;\ 3)\ \frac{d^2y}{dx^2}+\frac{dy}{dx}-6y=0$$

Вариант 2

Решить дифференциальное уравнение:

$$1)x^2dy+(2y+1)dx=0;\ 2)\frac{dy}{dx}+2y-3=0;\ 3)\ y''+5y'+6=0$$

ТЕМА ***«ТЕОРИЯ РЯДОВ»***

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Что необходимо знать по теме:

Определение: Пусть $u_1, u_2, u_3, \ldots u_n \ldots$, где $u_n = f(n)$ – бесконечная числовая последовательность. ***Выражение $u_1+u_2+u_3+\ldots+u_n+\ldots$*** называется ***бесконечным числовым рядом***, а числа u_1, u_2, u_3 – членами ряда; $u_n = f(n)$ – называется общим членом ряда.

Ряд часто записывают в виде: $\sum_{n=1}^{\infty} u_n = u_1+u_2+u_3+\ldots+u_n+\ldots$

Определение: Сумму первых n членов числового ряда обозначают через S_n и называют ***n-ой частичной суммой ряда***: $S_n=u_1+u_2+u_3+\ldots+u_n$

Определение: Ряд называется ***сходящимся***, если его n-ая частичная сумма S_n при неограниченном возрастании n стремится к конечному пределу, т.е. если $\lim_{n\to\infty} S_n = S$, где S- конечное действительное число.

Тогда число **S** называется ***суммой ряда:***

Определение: Ряд называется ***расходящимся***, если n-ая частичная сумма ряда при $n\to\infty$ не стремится к конечному пределу.

Определение: Ряд $a + aq + aq^2 + aq^{n-1} + \ldots \ \left(|q| < 1\right),$ составленный из членов любой убывающей геометрической прогрессии, является **сходящимся** и имеет сумму $S = \frac{a}{1-q}$ (геометрический ряд), где q- знаменатель геометрической прогрессии.

Свойства сходящихся рядов:

1. Если сходится ряд $u_1+u_2+u_3+\ldots$, то сходится и ряд $u_{m+1}+u_{m+2}+u_{m+3}+\ldots$, получаемый из данного ряда ***отбрасыванием первых m-членов*** (этот последний ряд называют **m-м остатком исходного ряда**); наоборот, из сходимости m-ого остатка ряда вытекает сходимость заданного ряда.

2. Если сходится ряд $u_1+u_2+u_3+\ldots$ и суммой его является число **S**, то сходится и ряд $au_1+au_2+au_3+\ldots$ (***полученный умножением данного ряда на число a***), причём сумма последнего ряда равна **a·S.**

3. Если ряды $u_1+u_2+u_3+\ldots+u_n+\ldots$ и $v_1+v_2+v_3+\ldots+v_n+\ldots$ сходятся и их суммы соответственно равны S_1 и S_2, то ряд $(u_1+v_1)+(u_2+v_2)+\ldots+(u_n+v_n)+\ldots$ (***представляющий сумму данных рядов***) также сходится, и его сумма равна S_1+S_2.

4. Если сумму n-ого остатка ряда обозначить через r_n, т.е. $r_n=u_{n+1}+u_{n+2}+\ldots+u_{n+m}+\ldots=\lim_{m\to\infty}\sigma_m$, то сумму ряда можно представить в виде: $S=S_n+r_n$,

то есть, суммам ряда **S** равна сумме n-ой частичной суммы ряда S_n и суммы n-ого остатка ряда r_n.

Теорема о необходимом и достаточном условии сходимости ряда с положительными членами:

Для того чтобы ряд $\sum_{n=1}^{\infty}$ u_n сходился, **необходимо и достаточно**, чтобы при $n\to\infty$ остаток ряда стремился к нулю, т.е. чтобы $\lim_{n\to\infty} r_n = 0$.

Теорема (необходимый признак сходимости):

Если ряд $\sum_{n=1}^{\infty} u_n$ сходится, то **предел его общего члена u_n при $n\to\infty$ равен нулю**, т.е. $\lim_{n\to\infty} u_n = 0$.

Признаки сходимости и расходимости рядов

с положительными членами:

I. Первый признак сравнения:

Пусть даны два ряда: $u_1+u_2+u_3+\ldots+u_n+\ldots$ (1)

$v_1+v_2+v_3+\ldots+v_n+\ldots$ (2),

причём ***каждый член первого ряда не превосходит соответствующего члена второго ряда*** т.е., $u_n \le v_n$ $(n=1,2,3,\ldots)$.

Тогда если сходится ряд (2), то сходится и ряд (1); если расходится ряд (1), то расходится и ряд (2).

Замечание: этот признак остаётся в силе, если неравенство $u_n \le v_n$ выполняется не при всех n, а лишь начиная с некоторого номера $n=N$ (т.к. **сходимость ряда не изменяется при отбрасывании конечного числа членов ряда**).

II. Второй признак сравнения (предельный признак сравнения):

Если $\sum_{n=1}^{\infty} u_n$ и $\sum_{n=1}^{\infty} v_n$ ряды с положительными членами и существует **конечный** предел отношения их общих членов: $\lim_{n\to\infty} \frac{u_n}{v_n} = k \neq 0$, то ряды **одновременно сходятся, либо расходятся.**

III. Теорема (признак Даламбера):

Если для ряда $\sum_{n=1}^{\infty} u_n$ *с положительными членами* существует конечный предел отношения (n+1)-го члена к n-му члену $\lim_{n\to\infty} \frac{u_{n+1}}{u_n} = l$, то этот ряд:

сходится, если $l < 1$ и ряд расходится если $l > 1$,

если $l = 1$, то вопрос о сходимости ряда остается нерешённым (нужно применить другой признак сходимости).

Рассмотрим применение этих теорем при решении задач.

Пример 1. Исследовать сходимость ряда: $\frac{1}{2}+\frac{2}{5}+\frac{3}{8}+\ldots+\frac{n}{3n-1}+\ldots$

$$\lim_{n\to\infty} u_n = \lim_{n\to\infty} \frac{n}{3n-1} = \left[\frac{\infty}{\infty}\right] = \lim_{n\to\infty} \frac{1}{3-\frac{1}{n}} = \frac{1}{3} \neq 0$$

▲ Так как , то ряд **расходится** (т.к. не выполняется необходимый признак сходимости $\lim_{n\to\infty} u_n = 0$).

Ответ: *ряд расходящийся*

Пример 2. Исследовать сходимость ряда: $\frac{1}{11}+\frac{1}{12}+\frac{1}{13}+\ldots\frac{1}{n+10}+\ldots$

▲ Данный ряд получен из **гармонического** ряда $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{n}+\ldots$ отбрасыванием первых десяти членов. Значит, он **расходится** (т.к. гармонический ряд является расходящимся и сходимость ряда не изменяется при отбрасывании конечного числа членов ряда).

Ответ: *ряд расходящийся*

Пример 3. Исследовать сходимость ряда и найти его сумму $\frac{2}{3}+\frac{1}{3}+\frac{1}{6}+\frac{1}{12}+\ldots+\frac{2}{5}\left[\frac{1}{2}\right]^{n-1}+\ldots$

▲ Данный ряд составлен из членов **бесконечно убывающей геометрической** прогрессии и поэтому сходится: $a=\frac{2}{3}$, $q=\frac{1}{2}$, (знаменатель геометрической прогрессии). Найдем сумму ряда:

$$S=\frac{a}{1-q}=\frac{\frac{2}{3}}{1-\frac{1}{2}}=\frac{4}{3}$$

Ответ: *ряд сходящийся,* $S=\frac{4}{3}$

ТЕОРЕМА (интегральный признак сходимости)

Если f (x) при $x \geq 1$ – непрерывная, положительная и монотонно убывающая функция, то ряд $\sum_{n=1}^{\infty} u_n$, где $u_n = f(n)$, сходится или расходится в зависимости от того, сходится или расходится несобственный интеграл $\int_1^{\infty} f(x)dx$.

Пример 1. Исследовать сходимость ряда: $1+\frac{1}{2^2}+\frac{1}{3^2}+\ldots+\frac{1}{n^2}+\ldots$

▲ Имеем: $u_n=\frac{1}{n^2}$; $u_{n+1}=\frac{1}{(n+1)^2}$. Найдём отношение:

$\frac{u_{n+1}}{u_n}=\frac{n^2}{(n+1)^2}=\frac{1}{\left[1+\frac{1}{n}\right]^2}$; тогда предел этого отношения равен:

$\lim_{n\to\infty}\frac{u_{n+1}}{u_n}=\lim_{n\to\infty}\frac{1}{\left[1+\frac{1}{n}\right]^2}=1$, т.к. $l=1$, то с помощью признака ***Даламбера не удаётся решить вопрос о сходимости ряда.***

Применим **интегральный признак:**

$u_n = \frac{1}{n^2}$, следовательно $f(x) = \frac{1}{x^2}$; вычислим несобственный интеграл: $\int_1^{\infty} \frac{dx}{x^2} = -\frac{1}{x}\Big|_1^{\infty} = 1$.

Интеграл сходится (является конечной величиной), поэтому ***сходится*** и заданный ряд согласно интегральному признаку сходимости рядов.

Ответ: *ряд сходящийся*

Пример 2. Исследовать сходимость ряда: $\frac{1}{2\ln 2} + \frac{1}{3\ln 3} + \frac{1}{4\ln 4} + \ldots + \frac{1}{(n+1)\ln(n+1)} + \ldots$

▲ Применим **интегральный признак**: $u_n = \frac{1}{(n+1)\ln(n+1)}$; $f(x) = \frac{1}{(x+1)\ln(x+1)}$. Найдём несобственный интеграл:

$$\int_1^{\infty} \frac{dx}{(x+1)\ln(x+1)} = \int_1^{\infty} \frac{d(x+1)}{x+1} = \int_1^{\infty} \frac{d(\ln(x+1))}{\ln(x+1)} = \ln(\ln(x+1))\Big|_1^{\infty} = \infty$$

- интеграл расходится,

поэтому ***расходится*** и заданный ряд.

Ответ: *ряд расходящийся*

Пример 3. Исследовать сходимость ряда $u_n = \frac{e^{-\sqrt{n}}}{\sqrt{n}}$

▲ Применим **интегральный признак:** $u_n = \frac{e^{-\sqrt{n}}}{\sqrt{n}}$; $f(x) = \frac{e^{-\sqrt{x}}}{\sqrt{x}}$. Найдём несобственный интеграл $\int_1^{\infty} \frac{e^{-\sqrt{x}}}{\sqrt{x}} dx = -2\int_1^{\infty} e^{-\sqrt{x}} d\left(-\sqrt{x}\right) = -2e^{-\sqrt{x}}\Big|_1^{\infty} = 2e^{-\sqrt{x}}\Big|_{\infty}^{1} = 2\left(e^{-1} - e^{-\infty}\right) = \frac{2}{e}$,

интеграл сходится, значит, ***сходится*** и заданный ряд.

Ответ: *ряд сходящийся*

Определение: Ряды, членами которых являются степенные функции, называются **степенными рядами**: $c_0 + c_1x + c_2x^2 + \ldots + c_nx^n + \ldots$,

а числа $c_0, c_1, \ldots, c_n$ – ***коэффициенты степенного ряда***.

Определение: Областью сходимости степенного ряда называется совокупность тех значений x, при которых степенной ряд сходится.

Пример: Найти область сходимости степенного ряда: $\mathbf{1 + x + x^2 + x^3 + \cdots + x^n + \cdots}$

▲**Решение:** Данный ряд можно рассматривать как геометрический ряд со знаменателем q=x, который сходится при |q| = |x| < 1. Отсюда -1<x<1, т.е. областью сходимости является интервал (-1;1).

Ответ: *(-1;1)*

Тригонометрическим рядом Фурье для функции $f(x)$ на отрезке $[-\pi;\pi]$ называется ряд вида $f(x): \frac{a_0}{2} + (a_1\cos x + b_1\sin x) + (a_2\cos 2x + b_2\sin 2x) + \ldots +$

$$+(a_n\cos nx + b_n\sin nx) + \ldots = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n\cos nx + b_n\sin nx),$$

коэффициенты которого, называемые коэффициентами Фурье для функции $f(x)$, вычисляются по следующим формулам:

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\,dx;$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos nx\,dx \qquad (n \in N);$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin ax\,dx \qquad (n \in N).$$

Условия, которые достаточно наложить на функцию $f(x)$ для того, чтобы её ряд Фурье сходился во всём промежутке разложения, определяются **теоремой Дирихле:**

Если функция $f(x)$, заданная на отрезке $[-\pi;\pi]$, удовлетворяет в этом промежутке условиям Дирихле, т.е.

1. непрерывна за исключением только конечного числа точек разрыва 1 рода;
2. имеет конечное число экстремумов,

то ряд Фурье этой функции сходится на всём отрезке $[-\pi;\pi]$ и сумма этого ряда:

равна $f(x)$ во всех точках непрерывности данной функции, лежащих внутри промежутка $(-\pi;\pi)$, $S(x) = f(x)$;

равна $\frac{f(x_0 - 0) + f(x_0 + 0)}{2}$ во всех точках разрыва: $S(x_0) = \frac{1}{2}(f(x_0 - 0) + f(x_0 + 0))$;

равна $\frac{f(-\pi + 0) + f(\pi - 0)}{2}$ на концах промежутка: $S(x) = \frac{1}{2}(f(-\pi + 0) + f(\pi - 0))$, $x = \pm\pi$.

Так как члены ряда – функции периодические с периодом 2π, то из сходимости ряда на отрезке $[-\pi;\pi]$ вытекает его сходимость на всей числовой оси, причем сумма этого ряда является периодической функцией с тем же периодом 2π.

Тогда для того, чтобы ряд Фурье функции $f(x)$ сходился именно к этой функции на всей числовой оси, надо и её считать периодической с периодом 2π.

ПРАКТИЧЕСКАЯ РАБОТА №24

«Исследование рядов».

Вариант 1

1. Найти сумму членов ряда

$$\sum_{n=1}^{\infty} \frac{1}{n(n+1)} = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \cdots + \frac{1}{n(n+1)} + \cdots$$

2. Исследовать сходимость ряда

$$\sum_{n=1}^{\infty} \frac{n}{n+1} = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \cdots + \frac{n}{n+1} + \cdots$$

Вариант 2

1. Найти сумму членов ряда

$$\sum_{n=1}^{\infty} \frac{1}{(3n-2)(3n+2)} = \frac{1}{1 \cdot 4} + \frac{1}{4 \cdot 7} + \frac{1}{7 \cdot 10} + \cdots + \frac{1}{(3n-2)(3n+2)} + \cdots$$

2. Исследовать сходимость ряда

$$\sum_{n=1}^{\infty} \frac{1}{(2n-1) \cdot 2^n} = \frac{1}{1 \cdot 2} + \frac{1}{3 \cdot 2^2} + \frac{1}{5 \cdot 2^3} + \cdots + \frac{1}{(2n-1) \cdot 2^n} + \cdots$$

Раздел 4. Основы теории комплексных чисел

ТЕМА: ***«ОСНОВЫ ТЕОРИИ КОМПЛЕКСНЫХ ЧИСЕЛ».***

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Что необходимо знать по теме:

Комплексным числом z называется выражение вида **a+bi**, a и b—действительные числа, а символ i удовлетворяет условию **i = - 1**

Число a называется действительной частью комплексного числа, **bi** — мнимой частью, i— мнимой единицей.

Множество комплексных чисел обозначается буквой ***C***.

Заметим, что множество **R** действительных чисел содержится в множестве комплексных чисел **C: R** $\subset$ **C** . В самом деле, всякое действительное число a можно рассматривать как комплексное число вида **a+0i**.

Комплексные числа вида **bi***называются чисто мнимыми. Они получаются из комплексных* чисел $z_1 = a + bi$при a = 0.

Определение 2. Два комплексных числа $z_1 = a + bi$ и $z_2 = c + di$ называются равными, если, соответственно, равны их действительные части и коэффициенты при мнимой единице, т. е. если a = c, b= d.

Комплексное число z = 0 + 0i называется нулем и обозначается через 0. Оно совпадает с числом нуль множества действительных чисел. Таким образом, z = a + bi = 0 тогда и только тогда, когда a = 0 и b= 0, или, что то же самое, когда $a^2 + b^2 = 0$.

Определение 3. Комплексные числа a + bi и a — bi называются комплексно-сопряженными.

Число, комплексно-сопряженное числу z, обозначается через $\bar{z}$. Так, если z = a + bi, то $\bar{z}$ = a — bi, если же z = a − bi, то $\bar{z}$ = a + bi. Понятие сопряженности взаимное. Например, для комплексного числа z = = -2 + 4i комплексно-сопряженным является комплексное число $\bar{z}$ = - 2 - 4i; точно также для комплексного числа -2 - 4i комплексно-сопряженным является число -2 + 4i .

2. Геометрическая интерпретация комплексных чисел

Комплексное число z = a + bi геометрически можно представить точкой координатной плоскости Oxy с координатами a, b.

Определение 4. Плоскость, служащая для изображения множества комплексных чисел, называется комплексной плоскостью. Так как любое комплексное число единственным образом определяется его действительной и мнимой частями, то каждому комплексному числу в комплексной плоскости соответствует единственная точка на плоскости. Очевидно, что справедливо и обратное утверждение: каждой точке (x; y) плоскости Oxy соответствует единственное комплексное число z = x + yi.

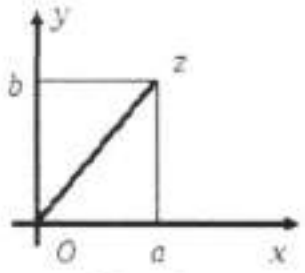


Рис. 1.

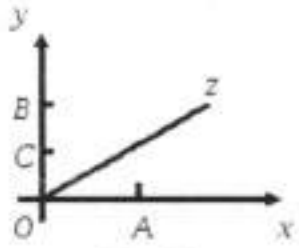


Рис. 2.

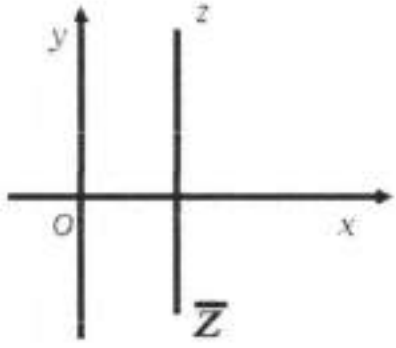


Рис. 3.

Таким образом, между множеством комплексных чисел и множеством точек плоскости существует взаимно-однозначное соответствие. При этом соответствии всякому действительному числу z = a + 0i соответствует точка A(a; 0) оси абсцисс, а всякому чисто мнимому числу z = 0 + bi — точка B(0; b) оси ординат. Числу z = i соответствует точка C(0; 1) (рис. 2). Если каждой точке M комплексной плоскости поставить в соответствие радиус-вектор OM этой точки, между множеством комплексных чисел множеством радиус-векторов можно также установить взаимно-однозначное соответствие.
Ось Ox будем называть действительной осью, а ось Oy— мнимой.

Действия над комплексными числами определяются таким образом, чтобы для частного случая действительных чисел эти операции совпадали с известными операциями над ними.

Сумма z комплексных чисел $z_1 = a_1 + b_1 i$ и $z_2 = a_2 + b_2 i$ определяется как комплексное число $z = (a_1 + a_2) + (b_1 + b_2)i$. Его обозначают $z = z_1 + z_2$.

Таким образом,

$$z_1 + z_2 = (a_1 + a_2) + (b_1 + b_2)i$$

В частности, если $z = a + bi$, то, $\bar{z} = a - bi$ поэтому $z + \bar{z} = 2a$, следовательно, сумма комплексно-сопряженных чисел есть число действительное. Операцию сложения легко распространить и на сумму любого конечного числа комплексных чисел. Так, если

$$z_1 = a_1 + b_1 i,\ z_2 = a_2 + b_2 i,\ \ldots,\ z_n = a_n + b_n i$$

то

$$z_1 + z_2 + \ldots + z_n = (a_1 + a_2 + \ldots + a_n) + (b_1 + b_2 + \ldots + b_n)i$$

Выберем на комплексной плоскости вместе с прямоугольной и полярную систему координат таким образом, как это изображено на рис. 6 Так как произвольное комплексное число $z = x + yi$ изображается точкой, то числа x и y являются ее прямоугольными координатами. Пусть z и φ — полярные координаты точки M(x;y). Полярный радиус $r = \sqrt{x^2 + y^2}$ называется модулем, или абсолютной величиной комплексного числа z и обозначается $|z|$, а полярный угол ϕ называется аргументом комплексного числа и обозначается arg $z (0 \le \arg z < 2\pi)$. Так как $x = r\cos\varphi$ и $y = r\sin\varphi$, то $z = x + yi = r\cos\varphi + ir\sin\varphi$ или

$$z = r(\cos\varphi + i\sin\varphi)$$

Правая часть (13) называется тригонометрической формой комплексного числа.

Запись комплексного числа z в виде $z = x + yi$ называется алгебраической формой комплексного числа. Переход от алгебраической формы записи к тригонометрической и обратно осуществляется по формулам

$r = |z| = \sqrt{x^2 + y^2}$, $tg\varphi = tg(\arg z) = \frac{y}{x}$;

$x = r\cos\varphi$, $y = r\sin\varphi$.

Пример 1. Составить тригонометрическую форму записи комплексного числа $z = -2 + 3i$.

Решение. Имеем: $r = \sqrt{(-2)^2 + 3^2} = \sqrt{13}$ $\varphi = arctg\left(-\frac{2}{3}\right)$, $(x < 0, y > 0, \frac{\pi}{2} \le \varphi < \pi)$.

По таблице тригонометрических функций или на калькуляторе находим $\varphi \sim 34^0$.

Следовательно, $z = \sqrt{13}(\cos 34^0 + i\sin 34^0)$.

Пусть даны два комплексных числа в тригонометрической форме:
$z_1 = r_1(\cos\varphi_1 + i\sin\varphi_1)$; $z_2 = r_2(\cos\varphi_2 + i\sin\varphi_2)$.

Перемножая их, получим
$z_1 z_2 = r_1 r_2(\cos\varphi_1 + i\sin\varphi_1)\cdot(\cos\varphi_2 + i\sin\varphi_2) = r_1 r_2(\cos\varphi_1\cos\varphi_2 + i\sin\varphi_1\cos\varphi_2 + i\cos\varphi_1\sin\varphi_2 -$
$-\sin\varphi_1\sin\varphi_2) = r_1 r_2((\cos\varphi_1\cos\varphi_2 - \sin\varphi_1\sin\varphi_2) + i(\sin\varphi_1\cos\varphi_2 + \cos\varphi_1\sin\varphi_2)) = r_1 r_2(\cos(\varphi_1 +$
$+\varphi_2) + i\sin(\varphi_1 + \varphi_2))$.

т. е.
$z_1 z_2 = r_1 r_2(\cos(\varphi_1 + \varphi_2) + i\sin(\varphi_1 + \varphi_2))$.

Таким образом, **при умножении** комплексных чисел в тригонометрической форме их модули перемножаются, а аргументы складываются

ПРАКТИЧЕСКАЯ РАБОТА №25

«Действия над комплексными числами, заданными в алгебраической и тригонометрической форме».

Вариант 1

1. Представить в тригонометрической форме комплексные числа: 1)2; 2) $-1+i$
2. Найти произведение комплексных чисел:

$3\left(\cos\left(\frac{\pi}{8}\right)+i\sin\left(\frac{\pi}{8}\right)\right)\cdot\left(\cos\left(\frac{5\pi}{24}\right)+i\sin\left(\frac{5\pi}{24}\right)\right)$

3. Возвести в степень: $\left[2\left(\cos\left(\frac{\pi}{8}\right)+i\sin\left(\frac{\pi}{8}\right)\right)\right]^{8}$

Вариант 2

1. Представить в тригонометрической форме комплексные числа: 1)3; 2)$\sqrt{3}-i$
2. Найти произведение комплексных чисел:

$2\left(\cos\left(\frac{\pi}{3}\right)+i\sin\left(\frac{\pi}{3}\right)\right)\cdot 5\left(\cos\left(\frac{-\pi}{4}\right)+i\sin\left(\frac{-\pi}{4}\right)\right)$

3. Возвести в степень: $[\cos 35^\circ + i\sin 35^\circ]^{-12}$

КОНТРОЛЬНАЯ РАБОТА №7

«Основы теории комплексных чисел».

Вариант 1.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{\sqrt{3}-i^{17}}{i^{12}}$
   b. $\frac{(1+i)^8}{(1-i)^6}$
2. Выполнить действия и записать результат в показательной форме:
   a. $3(\cos\frac{\pi}{3}+i\,sin\frac{\pi}{3})^2$
   b. $\frac{-1+i}{\sqrt{2}e^{i\frac{\pi}{3}}}$

Вариант 2.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{2i^5}{1+i^{11}}$
   b. $\frac{(1-i)^2}{(1+i)^4}$
2. Выполнить действия и записать результат в показательной форме:
   a. $7(\cos\frac{\pi}{6}+i\,sin\frac{\pi}{6})^3$
   b. $\frac{1+i}{\sqrt{2}e^{i\frac{\pi}{2}}}$

Вариант 3.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{1+i\sqrt{3}}{1-i\sqrt{3}}$

b. $\frac{(i-1)^3}{i^{12}+i^{31}}$

2. Выполнить действия и записать результат в показательной форме:
   a. $(24\cos 75° + i\,sin 75°) : (3(cos 30° + i sin 30°)$
   b. $\frac{e^{-\frac{\pi}{3}}}{(-\sqrt{3}+i)^5}$

Вариант 4.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{5+i}{2+3i}$
   b. $\frac{3i^{15}+(i\sqrt{3})^2}{(1-i)^6}$
2. Выполнить действия и записать результат в показательной форме:
   a. $3(\cos\frac{11\pi}{12} + i\,sin\frac{11\pi}{12})^2$
   b. $\frac{(-\sqrt{2}-i\sqrt{2})^6}{12e^{-\frac{\pi}{2}}}$

Вариант 5.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{1-2i}{1+3i}$
   b. $\frac{(i^9-1)(i^9+1)}{1-i}$
2. Выполнить действия и записать результат в показательной форме:
   a. $3(\cos\frac{3\pi}{4} + i\,sin\frac{3\pi}{4})^4$
   b. $\frac{e^{i\frac{\pi}{3}}\cdot i}{(\sqrt{3}-i)^4}$

Вариант 6.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{3i+3}{2i^{19}}$
   b. $\frac{(\frac{1}{2}-i\frac{\sqrt{3}}{2})^2}{i^{44}}$
2. Выполнить действия и записать результат в показательной форме:
   a. $4(\cos\frac{\pi}{6} + i\,sin\frac{\pi}{6})^{10}$
   b. $\frac{(1+i)^{15}}{2^7\cdot e^{i\frac{\pi}{2}}}$

Вариант 7.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{(\sqrt{3}-i)^3}{i^{22}}$
   b. $(-1+i\sqrt{3})^6$
2. Выполнить действия и записать результат в показательной форме:

a. $(2(\cos 40° + i\ sin 40°)(cos 50° + isin 50°))^2$

b. $\frac{\sqrt{2}\cdot e^{i\frac{\pi}{4}i}}{(-1+i)^3}$

Вариант 8.

1. Выполнить действия и записать результат в тригонометрической форме:
   a. $\frac{(i^{17}-i\sqrt{3})^2}{i^{12}}$

   b. $\frac{(1+i)^8}{1-i^{26}}$
2. Выполнить действия и записать результат в показательной форме:
   a. $3(\cos\frac{4\pi}{75} + i\ sin\frac{4\pi}{75})^{60}$

   b. $\left(\frac{\sqrt{2}e^{i\frac{2\pi}{5}}}{-1+i}\right)^{10}$

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

КОНТРОЛЬНО-РАСЧЕТНАЯ РАБОТА

по темам «Кольцо целых чисел», «Кольца вычетов»

по дисциплине «Математические основы защиты информации»

для специальности
09.02.03 Программирование в компьютерных системах

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий естественнонаучных дисциплин
Протокол от «15» 01 2020 г. № 6
Председатель цикловой комиссии Е.А.Рейм

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 $au + bv = d$.

Пример

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

$$m = \frac{ab}{d};$$

$$m = \frac{12103 * 1425}{19} = 907725$$

3 Для нахождения целых чисел *u,v*, удовлетворяющие условию $au + bv = d$, воспользуемся теоремой

$$r_k = au_k + bv_k,\ k = 1, \ldots, n,$$

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 $u_0 = 0$, $v_0 = 1$, $u_1 = 1$ $v_1 = -q_1$

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

Вариант	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
a	363	448	524	436	574	472	364	786	614	978	978	834	905	515	525
b	285	216	182	264	272	186	168	256	328	434	343	354	723	223	185
Вариант	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
a	120	183	222	56	574	329	776	365	286	366	228	332	625	99	572
b	56	110	123	34	273	64	124	12	126	166	136	64	128	19	324

Задание 2. Проверить являются ли числа a и b взаимно простыми.

Пример

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.

Вариант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
a	173	171	367	165	845	926	654	347	451	842	723	561	459	723	896
b	72	111	234	134	752	521	567	254	356	735	524	348	430	428	548
Вариант	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
a	358	166	864	547	713	486	922	451	199	646	416	649	853	565	914
b	241	127	345	165	647	169	657	133	154	139	365	351	564	186	249

Задание 3. Определить в множестве элементы сравнимые с заданным числом по данному модулю.

Пример

Определить в множестве A элементы сравнимые с числом 6 по модулю 4, если A={3,7,9,10,12,31,29,24,20,17,1,26}

Решение:

1. Найдем остатки от деления элементов множества A на модуль 4

   $3 = 3 \pmod 4$; $7 = 3 \pmod 4$; $9 = 1 \pmod 4$; $10 = 2 \pmod 4$; $12 = 0 \pmod 4$;

   $31 \equiv 3 \pmod 4$; $29 \equiv 1 \pmod 4$; $24 \equiv 0 \pmod 4$; $20 \equiv 0 \pmod 4$; $17 \equiv 2 \pmod 4$;

   $1 \equiv 1 \pmod 4$; $26 \equiv 2 \pmod 4$

2. Проанализируем полученные результаты и выберем те элементы множества A, 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

$$6 \equiv 10 \pmod 4;\ 6 \equiv 17 \pmod 4;\ 6 \equiv 26 \pmod 4$$

Вариант	**Множество**	**Число**	**Модуль**
1.	A={3,7,9,11,12,31,29,24,20,17,6}	20	3
2.	C={3,17,9,18,12,31,29,25,20,39,6}	45	4
3.	B={13,7,4,11,20,31,26,60,78,56,22}	12	8
4.	A={23,17,9,21,32,11,29,24,30,47,16}	9	6
5.	M={3,5,6,14,27,31,29,41,20,55,44}	13	7

Вариант	Множество	Число	Модуль
6.	D={13,25,36,24,17,1,28,31,23,54,50}	17	11
7.	C={43,7,19,29,12,34,29,25,20,39,6}	8	3
8.	K={24,17,9,30,12,31,28,25,34,39,6}	13	7
9.	C={3,17,9,18,12,31,29,25,20,38,73}	23	5
10.	P={4,27,18,10,62,51,29,74,29,87,26}	34	4
11.	M={17,7,36,14,27,37,29,49,20,53,41}	23	6
12.	A={23,7,9,11,14,31,29,84,20,17,16}	28	7
13.	C={3,11,89,18,12,31,29,25,20,39,6}	32	6
14.	B={33,59,4,11,20,31,26,60,75,56,22}	43	8
15.	A={23,47,50,21,32,15,29,24,30,47,16}	36	7
16.	M={64,5,16,44,27,31,29,41,20,56,44}	26	3
17.	D={13,26,36,27,38,2,28,31,23,54,61}	18	4
18.	C={43,7,26,129,14,31,29,46,20,39,66}	16	5
19.	K={24,14,194,37,12,131,33,25,95,39,4}	41	9
20.	C={3,17,9,18,12,31,30,25,90,38,72}	12	6
21.	P={4,27,19,10,62,53,29,74,29,87,26}	25	3
22.	A={3,67,99,11,62,31,29,55,20,17,46}	47	4
23.	C={43,17,79,46,12,31,29,125,20,39,16}	32	7
24.	B={33,7,44,11,26,31,126,80,68,54,22}	14	6
25.	A={23,17,309,121,32,114,29,24,147,42,16}	21	9
26.	M={17,5,46,14,27,31,29,42,21,58,47}	72	5
27.	D={13,26,39,24,127,1,27,131,23,54,50}	15	4
28.	C={43,7,119,25,12,34,129,25,95,39,26}	18	7
29.	K={24,17,58,30,13,31,28,25,135,29,97}	34	3
30.	C={13,114,19,78,12,44,29,125,20,38,73}	24	9

Задание 4. Определить функцию Эйлера для числа m.
Пример

m = 24

Решение.

Если число m простое то $\varphi(m) = m - 1$.

Если число m имеет каноническое разложение

$$m = p_1^{k1} \cdots p_r^{kr},$$

то

$$\varphi(m) = m(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})...(1 - \frac{1}{p_s})$$

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 $\varphi(24) = 8$.

Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m	145	154	238	162	128	172	321	345	142	168	225	155	198	174	237
Вариант	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
m	135	164	234	189	124	161	159	160	176	186	147	153	171	150	130

Задание 5. Найти остаток от деления числа a на число b.

Пример

$a = 13^{162}$; m = 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.

Вариант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
a	7^{130}	13^{146}	5^{147}	11^{132}	13^{134}	7^{218}	11^{252}	5^{281}	13^{173}	13^{196}	7^{188}	5^{237}	17^{145}	17^{145}	15^{132}
m	19	19	11	19	19	11	17	11	11	17	17	13	11	19	11
Вариант	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
a	17^{114}	11^{245}	7^{194}	5^{145}	19^{127}	17^{244}	13^{149}	17^{215}	11^{198}	19^{152}	7^{251}	7^{235}	15^{159}	15^{145}	16^{145}
m	23	13	19	23	17	11	23	13	23	23	23	11	13	19	19

Задание 6. Определить принадлежат ли следующие числа заданного множества полной системе вычетов по модулю m.

Пример

A = {0,13,25,37,49,61,73}; m = 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

$3 \equiv 3 \pmod 7$; $13 \equiv 6 \pmod 7$; $25 \equiv 4 \pmod 7$; $37 \equiv 2 \pmod 7$; $49 \equiv 0 \pmod 7$;

$61 \equiv 5 \pmod 7$; $73 \equiv 1 \pmod 4$

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.

Вариант	Множество	Модуль	Вариант	Множество	Модуль
1	A={3,11,24,20,17}	5	**16**	A={3,11,24,20,17}	5
2	C={14,9,18,31,29,34}	6	**17**	C={17,9,18,31,56,34}	6
3	B={13,7,11,20,26,72,57,22}	8	**18**	B={13,7,65,44,28,31,26,60,80,56,30}	11
4	A={26,34,21,32,29,24,30}	7	**19**	A={23,17,9,22,46,61,29,24,30}	9
5	M={3,14,27,31,29,42,19,53,34}	9	**20**	M={46,6,14,27,31,29}	6
6	D={12,23,37,27,16,2}	6	**21**	D={56,34,59,27,24,17,1,28,31,21,54,50}	12
7	C={43,7,19,30,34,45,18}	7	**22**	C={43,7,32,46,12,34,29,25}	8
8	K={24,19,9,30}	4	**23**	K={24,16,9,30,22,84,57,25,34,39,4}	11
9	C={31,29,26,21,36,82}	6	**24**	C={16,9,56,12,31,29}	6
10	P={88,18,10,62,51,29,75,32,22}	9	**25**	P={34,10,67,51,23,72,39,84,26}	9
11	M={29,50,22,51,43}	5	**26**	M={125,91,56,74}	4
12	A={23,37,32,11,14,31,28,82,24,18,19}	11	**27**	A={23,7,9,27,14,34,42,84,20,17,16,3}	12
13	C={8,3,15,89,18,12,32,45,6}	8	**28**	C={52,27,91,18,12,30,29}	7
14	B={20,31,23,60,75,56,22}	7	**29**	B={33,59,4,12,28,31,97,60,75,56}	10
15	A={23,47,50,21,33,15,29,24,30,86,16}	11	**30**	A={67,47,54,21,32,16,28,24,98}	9

Задание 7. Решить сравнение.

Пример.

$729x \equiv 33 \pmod{321}$

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 из полного алгоритма Евклида

$$v_k = v_{k-2} - v_{k-1}q_k,$$

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 $x = 48*11 = 528 \equiv 100(mod\ 107)$.

Или решения исходного сравнения будут равны:

$x_1 \equiv 100(mod\ 321)$, $x_2 \equiv 207(mod\ 321)$, $x_3 \equiv 314(mod\ 321)$.

Вариант	Сравнение	Вариант	Сравнение	Вариант	Сравнение
1	$48x \equiv 18 \pmod{26}$	**11**	$28x \equiv 21 \pmod{35}$	**21**	$82x \equiv 14 \pmod{35}$
2	$36x \equiv 16 \pmod{22}$	**12**	$38x \equiv 14 \pmod{22}$	**22**	$67x \equiv 17 \pmod{18}$
3	$72x \equiv 15 \pmod{21}$	**13**	$42x \equiv 16 \pmod{28}$	**23**	$57x \equiv 12 \pmod{45}$
4	$79x \equiv 24 \pmod{32}$	**14**	$25x \equiv 5 \pmod{15}$	**24**	$84x \equiv 24 \pmod{46}$
5	$42x \equiv 21 \pmod{28}$	**15**	$41x \equiv 13 \pmod{25}$	**25**	$53x \equiv 16 \pmod{24}$
6	$12x \equiv 7 \pmod{15}$	**16**	$56x \equiv 12 \pmod{18}$	**26**	$122x \equiv 5 \pmod{19}$
7	$27x \equiv 9 \pmod{15}$	**17**	$125x \equiv 15 \pmod{45}$	**27**	$94x \equiv 16 \pmod{33}$
8	$25x \equiv 15 \pmod{45}$	**18**	$48x \equiv 24 \pmod{27}$	**28**	$76x \equiv 14 \pmod{16}$
9	$57x \equiv 19 \pmod{38}$	**19**	$47x \equiv 12 \pmod{17}$	**29**	$52x \equiv 23 \pmod{29}$
10	$14x \equiv 12 \pmod{19}$	**20**	$25x \equiv 15 \pmod{45}$	**30**	$75x \equiv 26 \pmod{55}$

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

Методические рекомендации по выполнению реферата

по теме «Кодирование источников сообщений»

по дисциплине «Математические основы защиты информации»

специальности
09.02.03 Программирование в компьютерных системах

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий естественнонаучных дисциплин
Протокол от «15» 04 2020 г. №6
Председатель цикловой комиссии Е.А.Рейм

СОДЕРЖАНИЕ

ВВЕДЕНИЕ……..... 4

1 Структура реферата…… 5

2 Оформление реферата…… 6

3 Примерная тематика реферата…… 8

БИБЛИОГРАФИЧЕСКИЙ СПИСОК……... 9

ВВЕДЕНИЕ

Реферат - краткое изложение в письменном виде или в форме публичного выступления содержания книги, научной работы, результатов изучения научной проблемы.

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, по сути, нужно пересказать его (желательно своими словами). В реферате не требуется наличия большого фактического материала, глубокого анализа, фундаментальных выводов.

1 Структура реферата

Реферат выполняется в строгом соответствии с ГОСТ 7.32-2017 «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]

Реферат должен включать оглавление, введение, несколько глав (от 2 до 5), заключение и список использованных источников.

Структура обычного реферата:

- содержание;
- введение;
- несколько глав (от 2 до 5);
- заключение;
- список литературы (или библиографический список).

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 — 20 мм, правое — 15 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.

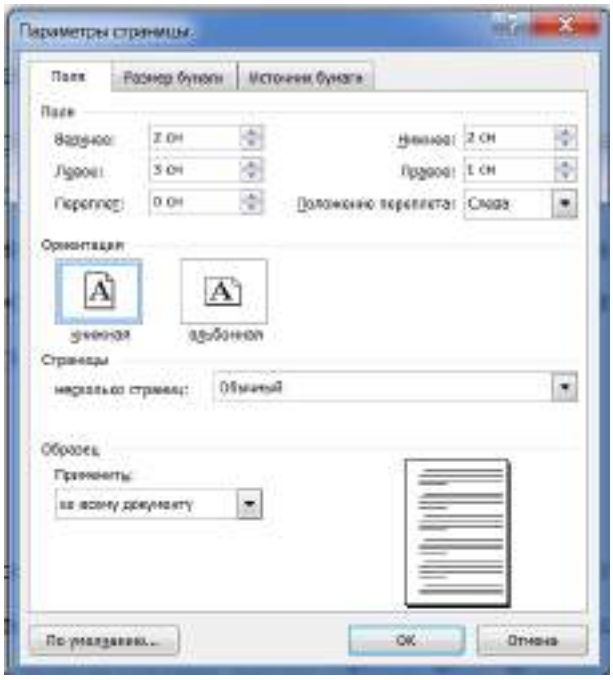


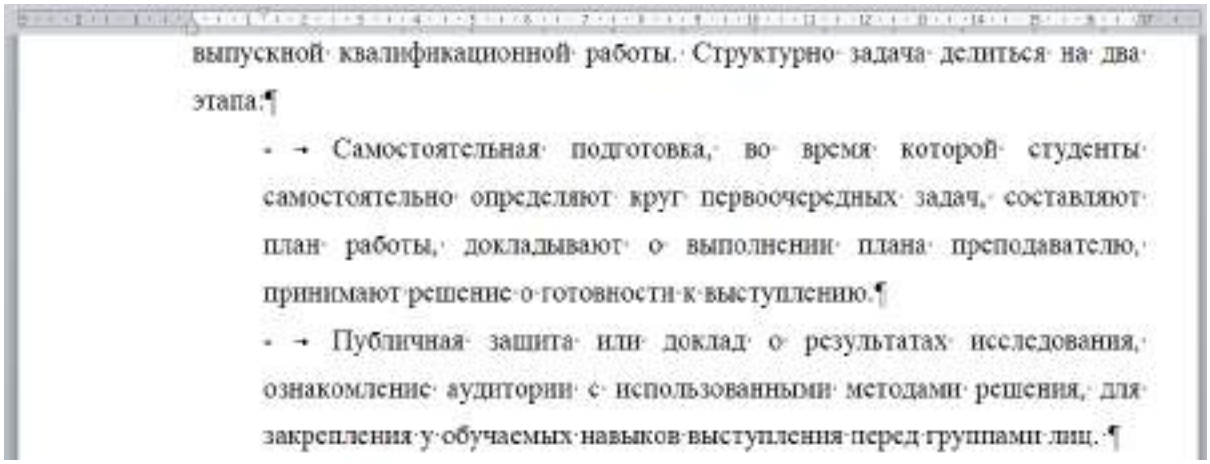
Рисунок 1 - Образец настройки параметров страницы

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 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.

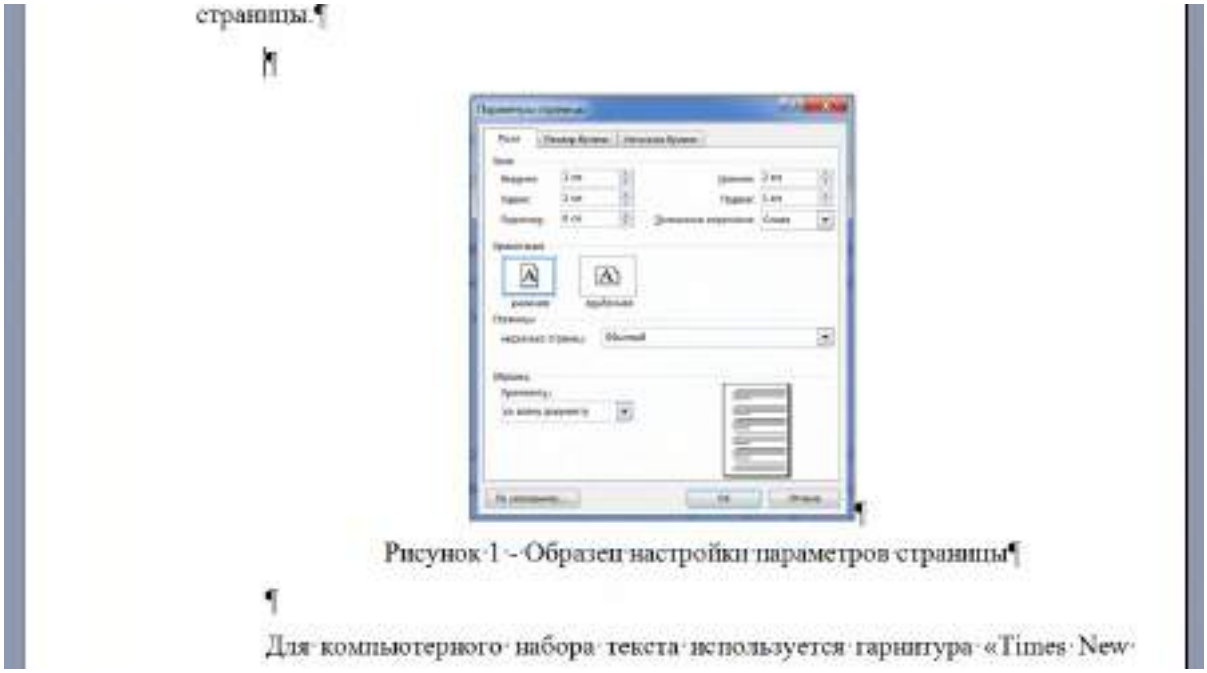


Рисунок 3 – Образец оформления рисунка

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.

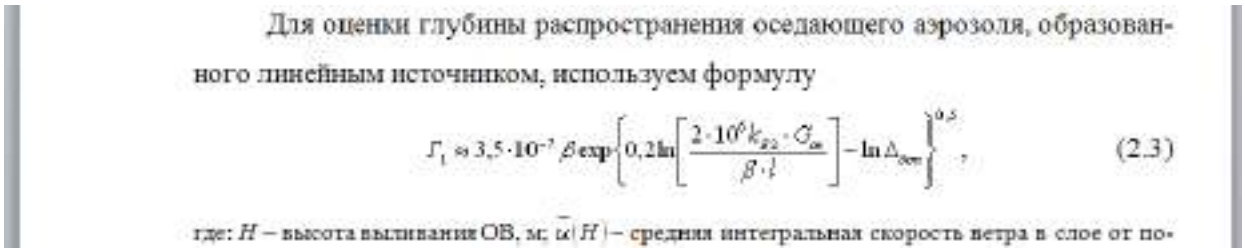
Таблица 5 – Распределение участников «Конкурса» по рабочим группам

№ 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	Глубин 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

$$\Gamma_1 \approx 3{,}5\cdot 10^{-7}\,\beta \exp\left\{0{,}2\ln\left[\frac{2\cdot 10^6 k_{B2}\cdot G_{\text{ов}}}{\beta\cdot l}\right]-\ln\Delta_{\text{доп}}\right\}^{0{,}5}, \qquad (2.3)$$

где: H – высота выливания ОВ, м; $\bar{u}(H)$ – средняя интегральная скорость ветра в слое от по-

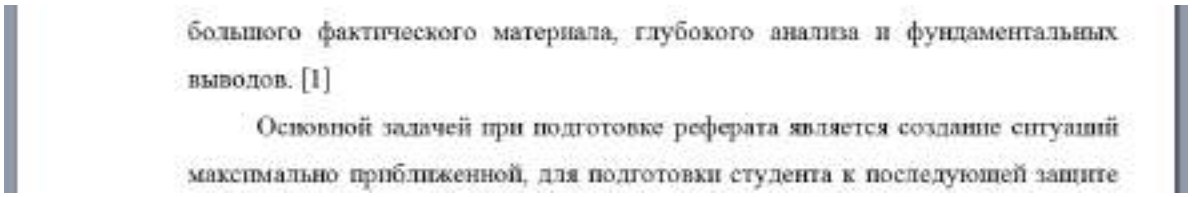
Рисунок 5 – Образец оформления формул

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 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» [Текст]. — Введ. 2001-05-22. — М.: Госстандарт России: Изд-во стандартов, 2001.

2. 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

3. 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

Рисунок 7 - Образец заполнения списка использованных источников

Обратите внимание, что при указании Интернет-ресурса, обязательно указывается и его название и электронный адрес.

3 Примерная тематика реферата

1. Арифметические коды
2. Метод кодирование Шеннона-Фано
3. Методы оптимального кодирования
4. Сжатие данных
5. Метод кодирование Лемпела-Зива
6. Сжатие с потерями
7. Линейные коды
8. Циклические коды
9. Корректирующее кодирование
10. Алфавитное кодирование
11. Сверточные коды
12. Каскадное кодирование
13. Сетевое кодирование
14. Оценка эффективности кодов
15. Способы кодирования
16. Метод четности
17. Блоковые коды
18. Коды CRC
19. Коды БЧХ
20. Коды Рида-Соломона
21. Неравномерное кодирование
22. Коды с памятью
23. Коды без памяти
24. История кодирования
25. Штриховое кодирование

БИБЛИОГРАИЧЕСКИЙ СПИСОК

1. **ГОСТ 7.32-2017** «Отчет о научно-исследовательской работе. Структура и правила оформления» [Текст]. . – Введ. 2001-05-22. – М. : Госстандарт России : Изд-во стандартов, 2001.

2. 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

3. 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

4. Пилиди, В. С. Математические основы защиты информации: учебное пособие / В. С. Пилиди. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2019. — 308 с. — ISBN 978-5-9275-3363-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/95786.html. — Режим доступа: для авторизир. пользователей

5. Осокин, А. Н. Теория информации : учебное пособие для среднего профессионального образования / А. Н. Осокин, А. Н. Мальчуков. — Москва : Издательство Юрайт, 2020. — 205 с. — (Профессиональное образование). — ISBN 978-5-534-11417-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/457083.

6. Фомичёв, В. М. Криптографические методы защиты информации в 2 ч. Часть 1. Математические аспекты : учебник для вузов / В. М. Фомичёв, Д. А. Мельников ; под редакцией В. М. Фомичёва. — Москва : Издательство Юрайт, 2020. — 209 с. — (Высшее образование). — ISBN 978-5-9916-7088-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/450820

Дополнительные источники:

1. Фомичёв, В. М. Криптографические методы защиты информации в 2 ч. Часть 2. Системные и прикладные аспекты : учебник для вузов / В. М. Фомичёв, Д. А. Мельников ; под редакцией В. М. Фомичёва. — Москва : Издательство Юрайт, 2020. — 245 с. — (Высшее образование). — ISBN 978-5-9916-7090-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451486

2. 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 Book.ru [сайт]. — URL: https://book.ru/book/933943

3. Гашков, С. Б. Дискретная математика : учебник и практикум для среднего профессионального образования / С. Б. Гашков, А. Б. Фролов. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 483 с. — (Профессиональное образование). — ISBN 978-5-534-13535-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/448573

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. НЭБ eLibrary. - Интернет-ссылка https://www.elibrary.ru/

ПРИЛОЖЕНИЕ

Пример оформления титульного листа

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

РЕФЕРАТ

по дисциплине «Математические основы защиты информации»

на тему: «Кодирование информации»

Автор работы,
студентка гр.4-090203 **А.А.Петрова**

Руководитель,
преподаватель **П.П.Иванова**

Тула 202_

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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению практических и самостоятельных работ

по дисциплине «Математические основы защиты информации»

для специальности 09.02.03 Программирование в компьютерных системах

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий естественнонаучных дисциплин
Протокол от «15» 01 2020 г. № 6
Председатель цикловой комиссии Е.А.Рейм

Тема 1.1 Группы

Практическое занятие № 1 Исследование группоидов

1 Теоретическая часть

Определение 1. Бинарной алгебраической операцией на множестве M называется произвольное правило f, которое любой упорядоченной паре элементов из M сопоставляет вполне определенный элемент из M. Другими словами, бинарная алгебраическая операция на M есть отображение множества M^2 во множество M.

Определение 2. Непустое множество G с одной бинарной алгебраической операцией * называется группоидом и обозначается в виде $(G, *)$, или просто G, если операция * известна из контекста.

Определение 3. Группоид $(G, *)$, а также и его операция *, называются ассоциативными и коммутативными, если соответственно выполняются равенства

$$(a* b) * c = a* (b*c), \qquad (1)$$

$$a* b = b* a\,, \qquad (2)$$

для любых элементов $a, b, c \in G$. Ассоциативный группоид называется также полугруппой.

Определение 4. Группоид $(G, *)$ называется группой, если выполняются три условия:

1) операция * ассоциативна, т. е. $(G, *)$ — полугруппа,

2) в G есть нейтральный элемент λ, т. е. такой элемент, что

$$a* \lambda = \lambda * a = a \text{ для любого } a \in G,$$

3) для любого элемента $a \in G$ в G существует симметричный к нему элемент a', т. е. такой элемент, что $a * a' = a' * a = \lambda$.

2 Практическая часть

Задача Выяснить являются ли полугруппами (группами) группоиды

а) числовое множество N с операцией сложения;

Решение

1 Операция сложения на множестве натуральных чисел является арифметической операцией, которая обладает свойством ассоциативности, следовательно, группоид N с операцией сложения является полугруппой.

2 Для того, чтобы выяснить является ли группоид N с операцией сложения группой, согласно определения, необходимо проверить выполнение трех условий.

1) Группоид N с операцией сложения является полугруппой (доказано выше).
2) В группоиде N (множестве натуральных чисел) отсутствует нейтральный элемент (нулевой элемент) для операции сложения:

$0 \notin N \Rightarrow$, т.е. условие 2 не выполняется, т.е. группоид N с операцией сложения не является группой.

б) числовое множество Z с операцией сложения.

1 Операция сложения на множестве целых чисел является арифметической операцией, которая обладает свойством ассоциативности, следовательно, группоид Z с операцией сложения является полугруппой.

2 Для того, чтобы выяснить является ли группоид Z с операцией сложения группой, согласно определения, необходимо проверить выполнение трех условий.

1) Группоид Z с операцией сложения является полугруппой (доказано выше).

2) В группоиде Z (множестве натуральных чисел) есть нейтральный элемент (нулевой элемент) относительно операции сложения:

$0 \notin Z \Rightarrow$ условие 2 выполняется;

3) Для любого элемента a $\in$ Z множестве целых чисел существует противоположный ему элемент -a, т. е. такой элемент, что a +(- a) = (-a) + a =0 следовательно, условие 2 выполняется.

Таким образом, группоид Z с операцией сложения является группой.

3 Задания для самостоятельной работы

1. Выяснить являются ли группоидом:
   - *a.* числовое множество R^+ с операциями сложения (умножения, деления, вычитания);
   - *b.* множество Q всех рациональных чисел относительно операции взятия среднего арифметического (среднего геометрического);
   - *c.* множество всех натуральных чисел меньших (больших) заданного числа n относительно операции сложения (умножения);
   - *d.* множество всех невырожденных матриц размером *nxn* относительно операции сложения (умножения) матриц.
2. Выяснить являются ли полугруппами (группами) группоиды:
   - *a.* $(G,+)$, $(G, \bullet)$, где $G = \{a/2k: a \in Z, k \in N_0\}$
   - *b.* $(G, \bullet)$, где $G = \{1, -1\}$
   - *c.* (G,+), (G, •), где *G* множество всех верхнетреугольных матриц размером *nxn*
3. Доказать, что группами являются:
   - *a.* Множество подстановок K_4 = {e, (1,2),(3,4), (1,3)(2,4), (1,4)(2,3)} с операцией умножения
   - *b.* Множество положительных рациональных чисел относительно операции умножения.

Тема 1.3 Кольцо целых чисел
Занятие 2 Делимость целых чисел. Нахождение НОД чисел

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, например, $b \neq 0$. 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

$$\begin{array}{ll}
a = bq_1 + r_1, & 0 < r_1 < |b|, \\
b = r_1q_2 + r_2, & 0 < r_2 < r_1, \\
r_1 = r_2q_3 + r_3, & 0 < r_3 < r_2, \\
\dots\dots\dots\dots\dots\dots & \\
r_{n-3} = r_{n-2}q_{n-1} + r_{n-1}, & 0 < r_{n-1} < r_{n-2}, \\
r_{n-2} = r_{n-1}q_n + r_n, & 0 < r_n < r_{n-1}, \\
r_{n-1} = r_nq_{n+1}. &
\end{array} \qquad (3)$$

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 $d|r_1, \ldots, r_n$. Значит r_n — наибольший общий делитель чисел a, b.

Определение 4. Целые числа a, b называются взаимно простыми, если их неотрицательный НОД равен 1, т. е. $(a, b) = 1$.

2 Практическая часть

Задача 1. Найти наибольший делитель $d=(a,b)$ следующих чисел

$a = 12354$; $b = 184$

Решение:

1. Применим алгоритм Евклида к числам $a = 12354$; $b = 184$:

$$12354 = 184*67 + 26$$

$$184 = 26*7 + 2$$

$$26 = 2*13 + 0$$

Таким образом, получили, что наибольший делитель

$$d = (12354, 184) = 2$$

Задача 2. Найти наибольший делитель $d=(a,b)$ следующих чисел

$a = 234$; $b = 180$

Решение:

1. Применим алгоритм Евклида к числам $a = 234$; $b = 180$:

$$234 = 180*1 + 54$$

$$180 = 54*3 + 18$$

$$54 = 18*3 + 0$$

Таким образом, получили, что найти наибольший делитель

$$d = (234, 180) = 18$$

Задача 3. Проверить являются ли числа a и b взаимно простыми.

$a = 141$; $b = 119$

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

Задание 1

Найти наибольший делитель $d=(a,b)$ чисел:

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

Задание 2

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

Занятие 3 Применение полного алгоритма Евклида. Нахождение НОК чисел

1 Теоретическая часть

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 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 $m \mid m_1$.

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

Задача 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 $au + bv = d$.

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

$$m = \frac{ab}{d};$$

$$m = \frac{12103 * 1425}{19} = 907725$$

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 таблицы:

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

3 Задания для самостоятельной работы

Задание 1

С помощью алгоритма Евклида найти целые числа u,v, удовлетворяющие условию $au + bv = d$.

а) 1814 и 194 б) 326 и 78 в) 1472 и 144 г) 528 и 132
д) 2346 и 642 е) 12237 и 1212 ж) 15105 и 1125

Задание 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

Тема 1.4 Кольца вычетов

Занятие 4 Исследование систем вычетов. Нахождение остатка от деления больших 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) $a \equiv a(\mathrm{mod}\ m)$,
2) $a \equiv b(\mathrm{mod}\ m), \Rightarrow b \equiv a(\mathrm{mod}\ m)$,
3) $a \equiv b(\mathrm{mod}\ m), b \equiv c(\mathrm{mod}\ m) \Rightarrow a \equiv c(\mathrm{mod}\ m)$,
4) $a \equiv b(\mathrm{mod}\ m) \Leftrightarrow m \mid a-b \Leftrightarrow a = b+mt$ *при некотором* $t \in Z$,
5) $a \equiv b,\ c \equiv d(\mathrm{mod}\ m) \Rightarrow a \pm c \equiv b \pm d(\mathrm{mod}\ m),\ ac \equiv bd(\mathrm{mod}\ m)$,

(*т. е. сравнения можно почленно складывать, вычитать и перемножать*),

6) если d есть общий делитель числа a, b, m, то $a \equiv b(\mathrm{mod}\ m) \Leftrightarrow \frac{a}{d} \equiv \frac{b}{d}(\mathrm{mod}\frac{m}{d})$,

7) если числа a, b делятся на числа d *и* d взаимно просто с m, то $a \equiv b(\mathrm{mod}\ m) \Leftrightarrow \frac{a}{d} \equiv \frac{b}{d}(\mathrm{mod}\ m)$.

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

$$m = p_1^{k1} \dots p_s^{ks},$$

то

$$\varphi(m) = m(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})\dots(1 - \frac{1}{p_s})$$

Теорема Эйлера. Если числа a и m взаимно просты, то

$$a^{\varphi(m)} \equiv 1(\mathrm{mod}\, m) \qquad (*)$$

Если m = p – простое число, то, $\varphi(p) = p - 1$, и сравнение (*) в этом случае принимается вид:

$$a^{p-1} \equiv 1(\mathrm{mod}\, p).$$

Это свойство чисел называют малой теоремой Ферма.

2 Практическая часть

Задача 1. Определить в множестве A элементы сравнимые с числом 6 по модулю 4, если A={3,7,9,10,12,31,29,24,20,17,1,26}

Решение:

1. Найдем остатки от деления числа 6 на модуль 4

   $$6 \equiv 2 \ (\mathrm{mod}\ 4).$$

2. Найдем остатки от деления элементов множества A на модуль 4

   $3 \equiv 3 \ (\mathrm{mod}\ 4)$; $7 \equiv 3 \ (\mathrm{mod}\ 4)$; $9 \equiv 1 \ (\mathrm{mod}\ 4)$; $10 \equiv 2 \ (\mathrm{mod}\ 4)$; $12 \equiv 0 \ (\mathrm{mod}\ 4)$;

   $31 \equiv 3 \ (\mathrm{mod}\ 4)$; $29 \equiv 1 \ (\mathrm{mod}\ 4)$; $24 \equiv 0 \ (\mathrm{mod}\ 4)$; $20 \equiv 0 \ (\mathrm{mod}\ 4)$; $17 \equiv 1 \ (\mathrm{mod}\ 4)$;

   $1 \equiv 1 \ (\mathrm{mod}\ 4)$; $26 \equiv 2 \ (\mathrm{mod}\ 4)$

3. Проанализируем полученные результаты и выберем те элементы множества A, которые имеют тот же остаток от деления на модуль 4, который получился в п.1.

   $10 \equiv 2 \ (\mathrm{mod}\ 4)$; $26 \equiv 2 \ (\mathrm{mod}\ 4)$

   Таким образом, получаем

   $$6 \equiv 10 \ (\mathrm{mod}\ 4); \ 6 \equiv 26 \ (\mathrm{mod}\ 4)$$

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, функция 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

Занятие 5 Решение сравнений первой степени с одним неизвестным

1 Теоретическая часть

Пусть

$$a_n x^n + a_{n-} x^{n-1} + ... + a_1 x^1 + a_0 = 0(\mathrm{mod}\, m) \qquad (5)$$

есть сравнения, в котором $a_0, a_1, ..., a_n \in Z$.

Определение 6. 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 $x_1 = x_2 (\mathrm{mod}\, m)$, 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

$$a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x^1 + a_0 \equiv 0(\mathrm{mod}\, m_j), i = 1,2.$$

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

$ax \equiv b(mod\, m)$, (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 $x = 48*11 = 528 \equiv 100 (mod\ 107)$.

Или решения исходного сравнения будут равны:

$x_1 \equiv 100 (mod\ 321)$, $x_2 \equiv 207 (mod\ 321)$, $x_3 \equiv 314 (mod\ 321)$.

3 Задания для самостоятельной работы

Задания

Выяснить какие из сравнений имеют решения и решить их:

а) $3x \equiv 5 (\mathrm{mod}\, 19)$ б) $5x \equiv 12 (\mathrm{mod}\, 26)$ в) $8x \equiv 7 (\mathrm{mod}\, 14)$ г) $18x \equiv 15 (\mathrm{mod}\, 69)$

д) $285x \equiv 51 (\mathrm{mod}\, 363)$ е) $2237x \equiv 100 (\mathrm{mod}\, 1021)$ ж) $12103x \equiv 385 (\mathrm{mod}\, 1425)$

Тема 1.5 Кольцо многочленов
Занятие 6 Нахождение НОД и НОК многочленов

1 Теоретическая часть

Определение 1. Пусть $a(x)$, $b(x)$ – произвольные многочлены из $P[x]$. Говорят, что многочлен $a(x)$ *кратен* многочлену $b(x)$ или делится на многочлен $b(x)$, если в $P[x]$ существует многочлен $q(x)$, удовлетворяющий условию

$$a(x) = b(x)\, q(x).$$

Обозначения:

$b(x) \mid a(x)$ – $b(x)$ делит $a(x)$,

$b(x) \nmid a(x)$ – $b(x)$ не делит $a(x)$.

Теорема 1. *Для любых многочленов $a(x)$, $b(x)$, $c(x) \in P[x]$ имеют место следующие свойства делимости:*

1) $a(x) \mid b(x) \Rightarrow r_1 \cdot a(x) \mid r_2 \cdot b(x)$ *при любых* $r_1, r_2 \in P\setminus\{0\}$,
2) $a(x) \mid b(x),\ b(x) \mid c(x) \Rightarrow a(x) \mid c(x)$,
3) $a(x) \mid b(x),\ a(x) \mid c(x) \Rightarrow a(x) \mid b(x) \pm c(x)$,
4) $a(x) \mid b(x) \Rightarrow a(x) \mid b(x)\, c(x)$,
5) $a(x) \mid b(x),\ a(x) \nmid c(x) \Rightarrow a(x) \nmid b(x) \pm c(x)$,
6) $r \mid a(x),\ r \cdot a(x) \mid a(x)$ *при любом* $r \in P\setminus\{0\}$,
7) $a(x) \mid b(x),\ b(x) \mid a(x) \Leftrightarrow \exists r \in P\setminus\{0\}: b(x) = r \cdot a(x)$.

Определение 2. Пусть $a(x)$, $b(x)$ – любые многочлены из кольца $P[x]$. Говорят, что $a(x)$ *делится с остатком* на $b(x)$, если существуют такие многочлены $q(x)$, $r(x) \in P[x]$, что выполняются условия:

1) $a(x) = b(x)q(x) + r(x)$,

2) $\deg r(x) < \deg b(x)$.

При этом многочлены $q(x)$ и $r(x)$ называются соответственно неполным частным и остатком от деления находятся однозначно.

Теорема 2. *Если $a(x)$, $b(x)$ – многочлены над полем P и $b(x) \neq 0$, от $a(x)$ можно разделить с остатком на $b(x)$, при этом неполное частное и остаток от деления находятся однозначно.*

Определение 3. Пусть $a_1(x)$, … $a_n(x)$ –многочлены над полем P, где n>1. *Наибольшим общим делителем* (НОД) многочленов $a_1(x)$, … $a_n(x)$ называется любой многочлен $d(x)$, который удовлетворяет двум условиям:

1) $d(x) \mid a_1(x), \ldots, d(x) \mid a_n(x)$,

2) если $d_1(x) \mid a_1(x), \ldots, d_1(x) \mid a_n(x)$, то $d_1(x) \mid d(x)$.

1. Если все многочлены $a_1(x)$, … $a_n(x)$ равны нулю, то для них существует единственный НОД, который равен нулю.

2. Если исходные многочлены не равны нулю, то для них существуют НОД, и все они исчерпываются многочленами вида rd(x), где $r \in P\setminus\{0\}$. Среди них

существует единственный унитарный многочлен. В дальнейшем через $(a_1(x), \ldots a_n(x))$ будем обозначать нулевой НОД, если $a_1(x) = \ldots = a_n(x)=0$, и унитарный НОД в противном случае.

3. Нахождение НОД любого числа многочленов сводится к нахождению НОД двух многочленов с помощью формулы

$$(a_1(x), a_2(x), \ldots a_n(x)) = ((a_1(x), a_2(x)), \ldots a_n(x)).$$

4. НОД двух многочленов находится с помощью алгоритма Евклида.

Алгоритм Евклида для многочленов $a(x)$ и $b(x)$при $b(x)\neq 0$ состоит в следующем.

Делят с остатком: многочлен $a(x)$ на $b(x)$. Если полученный остаток $r_1(x)$ равен нулю, то процесс окончен, и НОД, очевидно, является многочлен $b(x)$. В противном случае $b(x)$ делят на полученный остаток $r_1(x)$, затем $r_1(x)$ – на остаток $r_2(x)$ от деления $b(x)$ на $r_1(x)$, и т.д., пока не получат остаток, равный нулю. Весь этот процесс можно записать следующим образом:

$$\begin{gathered}
(x)= b(x)\, q_1(x)+ r_1(x),\\
b(x)= r_1(x)\, q_2(x)+ r_2(x),\\
r_1(x)= r_2(x)\, q_2(x)+ r_2(x),\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
r_{n-3}(x)= r_{n-2}(x)\, q_{n-1}(x)+ r_{n-1}(x),\\
r_{n-2}(x)= r_{n-1}(x)\, q_n(x)+ r_n(x).
\end{gathered}$$

Теорема 3. *1) Последний отличный от нуля остаток $r_n(x)$ в алгоритме Евклида, примененном к многочленам $a(x)$ и $b(x)\neq 0$ из кольца $P[x]$, является наибольшим общим делителем многочленов $a(x)$, $b(x)$.*

3) Многочлен $d(x)$ из кольца $P[x]$ является наибольшим общим делителем многочленов $a(x)$, $b(x)$ в том и только в том случае, когда $d(x)=cr_n(x)$, где $c\in P$ и $c\neq 0$.

Из алгоритма Евклида, как и в кольце Z, легко получается

Следствие. *Если $a(x)$, $b(x)\in P[x]$ и $d(x)$ есть НОД многочленов $a(x)$, $b(x)$, то существуют многочлены $u(x)$, $v(x)\in P[x]$ такие что*

$$d(x)= a(x)\, u(x)+b(x)v(x).$$

Определение 4. Многочлены $a_1(x)$, $a_2(x)$, $\ldots$ $a_n(x)$ называют *взаимно простыми*, если $(a_1(x), a_2(x), \ldots a_n(x))=1$.

Приведем простейшие свойства взаимно простых многочленов.

Теорема 4. *Для любых многочленов $a(x)$, $b(x)$, $c(x)$ имеют место утверждения:*

1) $(a(x), b(x))=1 \Leftrightarrow$ существуют $u(x)$, $v(x)\in P[x]$:

$$a(x)u(x)+b(x)v(x)=1,$$

2) $(a(x), b(x))=1, (a(x), c(x))=1 \Rightarrow (a(x), b(x)\, c(x))=1,$

3) $a(x) \mid b(x)\, c(x), (a(x), b(x))=1 \Rightarrow a(x) \mid c(x),$

4) $(a(x), b(x))=d(x)\neq 0 \Rightarrow \left(\frac{a(x)}{d(x)}, \frac{b(x)}{d(x)}\right)=1,$

5) если $a(x) \mid c(x)$, $b(x) \mid c(x)$ *и* $(a(x), b(x))=1$, *то* $a(x)b(x) \mid c(x)$,

6) если $(a(x), b(x))=1$ *и* $a_1(x) \mid a(x)$, $b_1(x) \mid b(x)$, *то* $(a_1(x), b_1(x))=1$

7) если $(a(x), b(x))=1$ *и* m, n – *любые натуральные числа, то* $(a(x)^m, b(x)^m)=1$.

Определение 5. *Наибольшим общим кратным* (НОК) многочленов $a_1(x), a_2(x), \ldots a_n(x)$, n>1 над полем P называется любой многочлен $m(x)$ из $P[x]$, удовлетворяющий условиям:

1) $a_1(x) \mid m(x), a_2(x) \mid m(x), \ldots a_n(x) \mid m(x)$;

2) если $m'(x) \in P[x]$ и $a_1(x) \mid m'(x), a_2(x) \mid m'(x), \ldots, a_n(x) \mid m'(x)$, то $m(x) \mid m'(x)$.

Для нахождения НОК двух многочленов можно воспользоваться его связью с НОД этих многочленов. Если $a(x)$, $b(x)$ – ненулевые многочлены над полем P и $d(x)=(a(x), b(x))$, то $a(x)\, b(x)/d(x)$ есть НОК многочленов $a(x)$, $b(x)$.

2 Практическая часть

Задача Найти НОД и НОК многочленов $a(x)$, $b(x)$ над полем P

$a(x) = x^5 + x^4 - 2x^3 - x^2 + 2x - 1$, $b(x) = x^4 - 2x^2 + x$

Решение.

1 Разделим «уголком» многочлены $a(x)$ на $b(x)$.

$$\begin{array}{l|l} x^5 + x^4 - 2x^3 - x^2 + 2x - 1 & x^4 - 2x^2 + x \\ \underline{x^5 - 2x^3 + x^2} & x + 1 \\ x^4 - 2x^2 + 2x - 1 & \\ \underline{x^4 - 2x^2 + x} & \\ x - 1 & \end{array}$$

2 Остаток $r_1(x) = x - 1$ отличен от нуля, поэтому продолжим деление: разделим «уголком» многочлен $b(x)$ на остаток $r_1(x)$.

$$
\begin{array}{l|l}
\_\,x^4 - 2x^2 + x & x - 1 \\
\underline{x^4 - x^3} & \\
\quad \_\,x^3 - 2x^2 + x & x^3 + x^2 - x \\
\quad \underline{x^3 - x^2} & \\
\quad\quad \_\,-x^2 + x & \\
\quad\quad \underline{-x^2 + x} & \\
\quad\quad\quad 0 &
\end{array}
$$

3 Остаток $r_2(x) = 0$, следовательно, предыдущий остаток $r_1(x) = x - 1$ является НОД исходных многочленов $a(x)$ и $b(x)$, т.е.

$$d = (a(x), b(x)) = x - 1.$$

4 Для нахождения НОК двух многочленов воспользуемся формулой

$$[a(x), b(x)] = a(x)\,b(x)/d(x), \text{ где } d(x) = (a(x), b(x))$$

$$[a(x), b(x)] = \frac{(x^5 + x^4 - 2x^3 - x^2 + 2x - 1)(x^3 + x^2 - x)}{x - 1} = x^8 + 2x^7 - 2x^6 - 4x^5 + 3x^4 + 2x^3 - 3x^2 + x$$

Таким образом,

$$[a(x), b(x)] = x^8 + 2x^7 - 2x^6 - 4x^5 + 3x^4 + 2x^3 - 3x^2 + x$$

3 Задания для самостоятельной работы

Задание. Найти НОД и НОК многочленов $a(x)$, $b(x)$ над полем P

а) $a(x) = x^5 + x^4 - 2x^3 - x^2 + 2x - 1$, $b(x) = x^4 - 2x^2 + x$

б) $a(x) = x^2 - 3x + 5$, $b(x) = 3x - 1$

в) $a(x) = x^4 + x^3 - 3x^2 - 6x - 3$, $b(x) = x^3 + 2x^2 + 2x + 1$

г) $a(x) = 2x^3 - 3x + 5$, $b(x) = x - 1$

д) $a(x) = x^4 + x^3 - 4x^2 + 5$, $b(x) = x^3 - x^2 - 2x + 2$

е) $a(x) = x^6 + x^5 + x^3 + x - 2$, $b(x) = x^5 + x^4 + x^3 + x^2 + 1$

Занятие 7 Нахождение корней многочленов по схеме Горнера

1 Теоретическая часть

Пусть

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

– многочлен над кольцом R.

Определение 6. Элемент r кольца R называется *корнем* многочлена $f(x) \in R[x]$, если значение многочлена $f(x)$ при $x = r$ равно нулю.
Короче: r – корень $f(x)$, если $f(r) = 0$.

Теорема 5 (Безу). Значение многочлена $f(x)$ над полем P при $x = a$ равно остатку от деления многочлена $f(x)$ на многочлен $x-a$.

Следствие 1. Элемент a поля P является корнем многочлена $f(x) \in P[x]$ тогда и только тогда, когда $f(x)$ делится на многочлен $x-a$.

Следствие 2. Многочлен $f(x)$ степени n над полем P может иметь в поле P не более n различных корней.

Ввиду следствия 1 из теоремы Безу особый смысл приобретает задача деления многочлена на двучлен вида $x-c$. Это деление можно осуществить несколько проще, чем шаблонным делением «уголком». Пусть требуется разделить с остатком многочлен $f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ степени $n > 1$ на двучлен $b(x) = b(x)g(x) + r$, где $g(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \ldots + b_1 x + b_0$ – частное, r – остаток от деления. Отсюда, приравнивая коэффициенты при одинаковых степенях x в многочленах $f(x)$ и $b(x)g(x) + r$, получим систему равенств:

$$b_{n-1} = a_n,\ b_{n-2} = a_{n-1} + cb_{n-1},$$
$$b_{n-3} = a_{n-2} + cb_{n-2},\ \ldots,\ b_0 = a_1 + cb_1,\ r = a_0 + cb_0.$$

Исходные данные и результаты вычислений обычно записывают в таблицу, которую называют *схемой Горнера* по имени английского математика В.Д. Горнера (1786 – 1837) (несмотря на то, что в действительности она была известна задолго до Горнера).

Теорема 6. Если несократимая дробь p/g является корнем многочлена $f(x) \neq 0$ с целыми коэффициентами, то p является делителем свободного члена многочлена $f(x)$, а g – делителем его старшего коэффициента.

2 Практическая часть

Задача 1. Разделить с остатком многочлен $f(x) = 2x^5 - 13x^4 + 15x^2 - 7x + 5$ на $x-6$.

Решение.
Схема Горнера имеет следующий вид:

	2	-13	0	15	-7	5
6	2	-1	-6	-21	-133	-793

Ответ: частное $g(x) = 2x^4-x^3-6x^2-21x-133$, остаток $r = -793$. Отсюда на основании следствия 1 из теоремы Безу видно, в частности, что 6 не является корнем многочлена $f(x)$.

Задача 2. Найти все рациональные корни многочлена

$$f(x) = 2x^5 + x^4 + x^3 - 5x^2 - 3x + 4.$$

Решение.

Из теоремы 2 следует, что рациональными корнями этого многочлена могут быть лишь числа

$$\pm 1,\ \pm 2, \pm 4, \pm\frac{1}{2}.$$

Далее проведем непосредственную проверку по схеме Горнера:

	2	1	1	-5	-3	4
1	2	3	4	-1	-4	0

Остаток $r = 0$. Отсюда на основании следствия 1 из теоремы Безу видно, в частности, что *1* является корнем многочлена $f(x)$.

	2	1	1	-5	-3	4
-1	2	-1	2	-7	4	0

Остаток $r = 0$, следовательно, *-1* является корнем многочлена $f(x)$.

	2	1	1	-5	-3	4
2	2	5	11	17	31	66

Остаток r = 66. Отсюда на основании следствия 1 из теоремы Безу видно, что 2 не является корнем многочлена $f(x)$.

	2	1	1	-5	-3	4
-2	2	0	1	-7	11	-18

Остаток r = -18, следовательно, -2 не является корнем многочлена $f(x)$.

	2	1	1	-5	-3	4
4	2	9	37	143	569	2280

Остаток r = 2280, следовательно, 4 не является корнем многочлена $f(x)$.

	2	1	1	-5	-3	4
-4	2	-7	29	-121	484	-1920

Остаток r = -1920, следовательно, -4 не является корнем многочлена *f(x)*.

	2	1	1	-5	-3	4
$\frac{1}{2}$	2	2	2	-4	-5	$\frac{3}{2}$

Остаток r = $\frac{3}{2}$, следовательно, $\frac{1}{2}$ не является корнем многочлена *f(x)*.

	2	1	1	-5	-3	4
$-\frac{1}{2}$	2	0	1	$-\frac{11}{2}$	$-\frac{1}{4}$	$4\frac{1}{8}$

Остаток r = $4\frac{1}{8}$, следовательно, $-\frac{1}{2}$ не является корнем многочлена *f(x)*.

Таким образом, убеждаемся, что в действительности рациональными корнями являются лишь числа

$$x_1 = 1, x_2 = -1.$$

Задача 3. Разложить на множители многочлен $x^3 - 5x^2 - 2x + 16$ над полем *R*.

Решение.

Данный многочлен имеет целые коэффициенты. Если целое число является корнем этого многочлена, то оно является делителем числа 16 (теорема 2). Таким образом, если у данного многочлена есть целые корни, то это могут быть только числа ±1; ±2; ±4; ±8; ±16. Непосредственной проверкой убеждаемся, что число 2 является корнем этого многочлена, то есть x3 – 5x2 – 2x + 16 = (x – 2)Q(x), где Q(x) – многочлен второй степени. Следовательно, многочлен разлагается на множители, один из которых (x – 2). Для поиска вида многочлена Q(x) воспользуемся так называемой схемой Горнера.

	1	-5	-2	16
2	1	-3	-8	0

Полученные числа 1, −3, −8 являются коэффициентами многочлена, который получается при делении исходного многочлена на x – 2. Значит

$$x3 - 5x2 - 2x + 16 = (x - 2)(x2 - 3x - 8).$$

Корни многочлена второй степени ищутся по формуле

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

В данном случае:

$x_1 = \frac{3+\sqrt{41}}{2}$ и $x_2 = \frac{3-\sqrt{41}}{2}$

Окончательно:

$$x^3 - 5x^2 - 2x + 16 = (x-2)\left(x - \frac{3-\sqrt{41}}{2}\right)\left(x - \frac{3+\sqrt{41}}{2}\right)$$

Ответ:

$$(x-2)\left(x - \frac{3-\sqrt{41}}{2}\right)\left(x - \frac{3+\sqrt{41}}{2}\right).$$

3 Задания для самостоятельной работы

Задание 1. Разделить многочлен $f(x)$ на двучлен с остатком, используя схему Горнера:

а) $f(x) = 3x^4 - 5x^2 + 3x - 1$ на $x-2$
б) $f(x) = 2x^3 - 3x^2 + 5x - 14$ на $x-3$

Задание 2. Найти все рациональные корни многочленов:
а) $f(x) = 6x^5 + 7x^4 + 5x^3 + 5x^2 - x - 2;$
б) $f(x) = x^4 + 3x^2 - 4;$
в) $f(x) = 3x^5 - 2x^3 - 7x + 6$

Задание 3. Разложить на множители многочлен

а) $x^4 + 5x^3 - 7x^2 - 5x + 6.$
б) $5x^4 + 3x^3 + 2x^2 + 5x + 1$

Задание 4. Методом перебора с использованием схемы Горнера найти все корни многочлена $a(x)$ над полем P и разложить $a(x)$ на множители
а) $a(x) = x^6 + x^5 + 2x^4 + 3x^3 + 2x^2 + 2x + 1.$
б) $a(x) = x^5 + 5x^4 + x^3 + 6x^2 + 6x + 4$
в) $a(x) = x^4 - 15x^2 - 10x + 24$

Задание 5. Элемент a поля P называется k-кратным корнем многочлена $a(x) \in P[x]$ при $k>0$, если $a(x)$ делится на $(x-a)^k$ и не делится на $(x-a)^{k+1}$. При этом 1-кратные корни называются простыми корнями. Найти все корни и их кратность для многочленов:
а) $f(x) = x^7 + 4x^6 + 4x^5 + 2x^4 + 4x^3 + 3x^2 + 3x + 4$ над полем $P = Z/5;$
б) $a(x) = 4x^6 + 24x^5 + 29x^4 - 37x^3 - 34x^2 + 39x - 9$ над полем $P = Q.$

Тема 2.2 Кодирование источников сообщений
Занятие 8 Кодирование информации по методу Хаффмана

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 1. 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

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

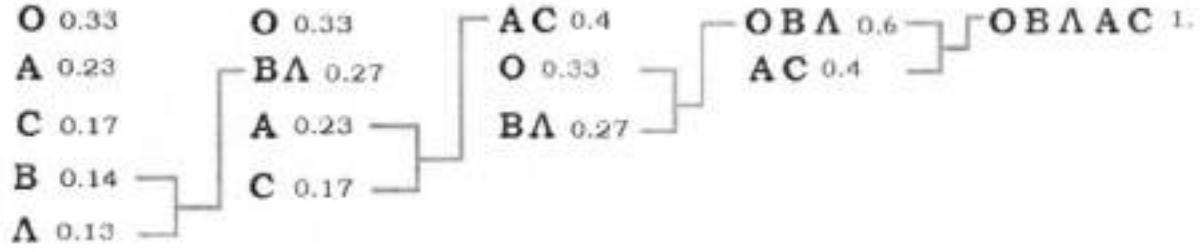
В полученном списке объединяем **о** и **вл**:

овл ас

0.6 0.4

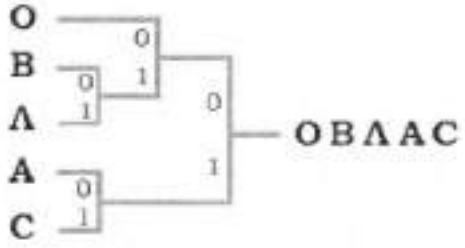
Последним шагом объединяем двухбуквенное и трехбуквенное слова в одно единое **овлас** с приписанной ему вероятностью 1.

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

3 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) ааабббвввааагггггббвввдддддаааа;

б) пппрлллуууурррраааааллллуууупппаааууурпп;

в) кккнннииииккннддддиииккккоооннн

Занятие 9 Кодирование информации по методу Хемминга

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 *К* 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

Рассмотрим принцип построения кода Хемминга для 16-разрядных данных (М= 16), исправляющего все одиночные ошибки.

Сначала определим необходимое число проверочных разрядов *К [K=N-M]* из соотношения:

$$2^{N-M}-1 \geq N.$$

Откуда для *М*= 16 находим N=21 и *К=5.*

Верхние границы *М* и *N* для различных *К* 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.

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

$a_j x2^j+a_{j-1}x2^{j-1}+...+a_2x2^2+a_1x2^1+ 1x2^0$, где $j=1og_2N$, a_i— произвольное значение. Таким образом, первый контрольный разряд контролирует все нечетные номера.

Второй контрольный разряд контролирует разряды кода Хемминга с номерами

$a_j x2^j+a_{j-1}x2^{j-1}+...+a_2x2^2+1x2^1+ a_0x2^0$.

Соответственно третий контрольный разряд контролирует разряды кода Хемминга с номерами $a_j x2^j+a_{j-1}x2^{j-1}+...+1x2^2+a_1x2^1+ a_0x2^0$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).

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).

3 Задания для самостоятельной работы

1Закодировать сообщение, используя алгоритм кода Хэмминга:

а) 1011100011110111
б) 1100011000111101
в) 1101110111011110011
г) 1110110111011100011011
д) 1111100011111000111111

2 Определите верное ли получено сообщение, закодированное кодом Хэмминга. В случае необходимости исправьте ошибку.

а) 0101011000011000111110011011
б) 1101000010101100111011010100 1
в) 0100111110111111011011
г) 1010110000101000011011
д) 101100111011011

Тема 3.1 Генераторы псевдослучайных последовательностей
Занятие 10 Основные типы генераторов псевдослучайных последовательностей

1 Теоретическая часть

МЕТОД СРЕДИННЫХ КВАДРАТОВ

Одним из первых программных ГСЧ является метод срединных квадратов, предложенный в 1946 г. Дж. фон Нейманом. Этот ГСЧ формирует следующий элемент последовательности на основе предыдущего путем возведения его в квадрат и выделения средних цифр полученного числа.

Метод используется для генерации k-разрядных псевдослучайных чисел. Должно быть задано k-разрядное начальное число x_0 (для удобства полагаем, что k четно). Это число возводится в квадрат и получается число y. Число y должно иметь $2k$ разрядов. Если число разрядов меньше, чем $2k$, то число y дополняется слева нулями. Затем в y выделяют средние k разрядов, которые дают новое случайное число. Десятичная точка помещается перед первым разрядом число x_i до поступления его на выход случайного генератора.

Пусть имеется k-разрядное число, меньше 1:

$$x_i = 0,a_1a_2...a_k.$$

Возведем его в квадрат:

$$y = x_i^2 = 0,b_1b_2...b_{2k},$$

затем отберем средние k разрядов

$$x_{i+1} = 0,b_{k/2+1}b_{k/2+2}...b_k,$$

которые будут являться очередным числом псевдослучайной последовательности.

Например, если начальное число

$x_0 = 0,2152$, то
$(x_0)^2 = 0,04631104$, т.е.
$x_1 = 0,6311$, затем
$(x_1)^2 = 0,39828721$, т.е.
$x_2 = 0,8287$, и т.д.

ЛИНЕЙНЫЙ КОНГРУЭНТНЫЙ МЕТОД

Метод используется для генерации последовательности

$$x_1, x_2, ..., x_m,$$

из m псевдослучайных чисел. Должны быть заданы следующие входные значения:

b – целочисленный множитель, $b \geq 1$

x_0 – начальное «необработанное» случайное целое число $x_0 \geq 1$

k – шаг, $k \geq 0$, целое

m – целочисленный модуль, $m \geq x_0, b, k$

$$x_{i+1} = (bx_i + k) \bmod m$$

Например, необходимо получить три случайных числа линейным конгруэнтным методом.

$x_0 = 27$; $b = 5$; $k = 10$; $m = 40$;
$x_1 = (5*27+10) \bmod 40 = 145 \bmod 40 = 25$;
$x_2 = (5*25+10) \bmod 40 = 15$;
$x_3 = (5*15+10) \bmod 40 = 5$

2 Практическая часть

Задача 1. Получить три случайных числа методом срединных квадратов: $k = 4$; $x_0 = 0{,}3167$.

Решение.
$k = 4$; $x_0 = 0{,}3167$ $y = (x_0)^2 = 10029889$
$x_1 = 0{,}0298$; $y = 00088804$
$x_2 = 0{,}0888$; $y = 00788544$
$x_3 = 0{,}7885$; $y = 00788544$

Задача 2.
Для $k = 4$; $x_0 = 2134$ определить первые десять псевдослучайных чисел, выработанные алгоритмом срединных квадратов:

Решение.

$(x_0)^2 = 04\ 5539\ 56$	$x_1 = 0{,}5539$
$(x_1)^2 = 30\ 6805\ 21$	$x_2 = 0{,}6805$
$(x_2)^2 = 46\ 3080\ 25$	$x_3 = 0{,}3080$
$(x_3)^2 = 09\ 4864\ 00$	$x_4 = 0{,}4864$
$(x_4)^2 = 23\ 6584\ 96$	$x_5 = 0{,}6584$
$(x_5)^2 = 43\ 3490\ 56$	$x_6 = 0{,}3490$
$(x_6)^2 = 12\ 1801\ 00$	$x_7 = 0{,}1801$
$(x_7)^2 = 03\ 2436\ 01$	$x_8 = 0{,}2436$
$(x_8)^2 = 05\ 9340\ 96$	$x_9 = 0{,}6340$
$(x_9)^2 = 87\ 2356\ 00$	$x_{10} = 0{,}2356$

Задача 3. Сгенерировать псевдослучайную последовательность (ПСП) из 8-ми чисел линейным конгруэнтным методом при следующих входных параметрах:

$$k = 0;\ m = 2^{10};\ b = 101;\ x_0 = 432$$

Решение.

$x_1 = 624$	$y_1 = 624/1024 = 0{,}610$
$x_2 = 560$	$y_2 = 560/1024 = 0{,}546$
$x_3 = 240$	$y_3 = 240/1024 = 0{,}234$

$x_4 = 688$	$y_4 = 688/1024=0,673$
$x_5 = 880$	$y_5 = 880/1024=0,859$
$x_6 = 816$	$y_6 = 816/1024=0,796$
$x_7 = 496$	$y_7 = 496/1024=0,486$
$x_8 = 944$	$y_8 = 944/1024=0,923$

2 Задания для самостоятельной работ

Задание 1.

Определить первые десять псевдослучайных чисел, выработанные алгоритмом срединных квадратов, при следующих входных параметрах:

а) $k = 10$; $x_0 = 5772156649$;

б) $k = 8$; $x_0 = 47252431$;

в) $k = 4$; $x_0 = 8526$;

г) $k = 10$; $x_0 = 3458544186$.

д) $k = 6$; $x_0 = 621438$.

Задание 2.

Сгенерировать псевдослучайную последовательность (ПСП) из 10-и чисел линейным конгруэнтным методом при следующих входных параметрах:

а) $k = 10$; $m = 2^{11}$; $b = 102$; $x_0 = 547$;

б) $k = 100$; $m = 2^{10}$; $b = 111$; $x_0 = 326$;

в) $k = 200$; $m = 2^{12}$; $b = 201$; $x_0 = 489$;

г) $k = 0$; $m = 2^{10}$; $b = 303$; $x_0 = 654$;

д) $k = 20$; $m = 2^{12}$; $b = 115$; $x_0 = 743$

СОДЕРЖАНИЕ

Тема 1.1 Группы..... 3
Занятие 1 Исследование группоидов..... 3
Тема 1.3 Кольцо целых чисел..... 5
Занятие 2 Делимость целых чисел. Нахождение НОД чисел..... 5
Занятие 3 Применение полного алгоритма Евклида. Нахождение НОК чисел..... 8
Тема 1.4 Кольца вычетов 10
Занятие 4 Исследование систем вычетов. Нахождение остатка от деления больших чисел..... 10
Занятие 5 Решение сравнений первой степени с одним неизвестным..... 13
Тема 1.5 Кольцо многочленов..... 16
Занятие 6 Нахождение НОД и НОК многочленов..... 16
Занятие 7 Нахождение корней многочленов по схеме Горнера..... 20
Тема 2.2 Кодирование источников сообщений..... 24
Занятие 8 Кодирование информации по методу Хаффмана..... 24
Занятие 9 Кодирование информации по методу Хемминга..... 27
Тема 3.1 Генераторы псевдослучайных последовательностей..... 30
Занятие 10 Основные типы генераторов псевдослучайных последовательностей..... 30

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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных и самостоятельных работ

по междисциплинарному курсу
МДК 01.01«Системное программирование»

для специальности 09.02.03 Программирование в компьютерных системах (углубленная подготовка)

Тула 2020

УТВЕРЖДЕНА
Цикловой комиссией информационных технологий

Протокол от «15» января 2020 № 6

Председатель цикловой комиссии И.В. Миляева

Автор: Басалова Г.В., доцент кафедры информационной безопасности ТулГУ, канд. техн. наук

Лабораторная работа 1.
Сбор информации о системе

Цель работы: изучение возможностей использования классов Application, Environment, DriveInfo, Path в среде программирования MS Visual C++.

Краткие теоретические положения

Класс Application предоставляет методы и свойства для управления приложением, например методы для запуска и остановки приложения, для обработки сообщений Windows и свойства для получения сведений о приложении. Основные свойства этого класса:

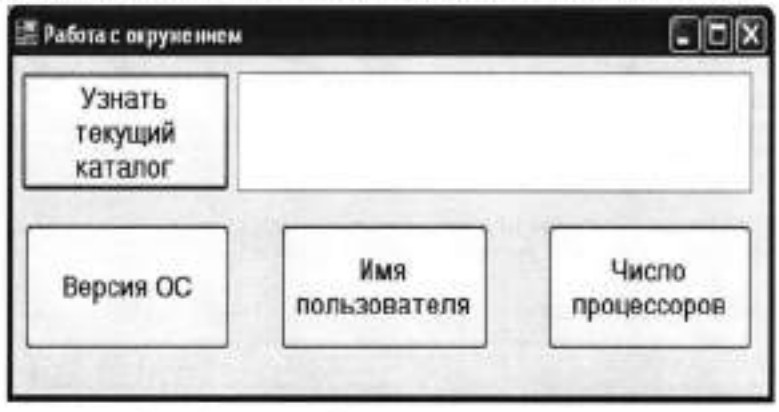
Имя	Описание
CommonAppDataPath	Получает путь для данных приложения, являющихся общими для всех пользователей.
CommonAppDataRegistry	Получает раздел реестра для данных приложения, являющихся общими для всех пользователей.
CompanyName	Возвращает название компании, связанное с приложением.
ExecutablePath	Возвращает путь исполняемого файла, запустившего приложение, включая имя исполняемого файла.
OpenForms	Возвращает коллекцию открытых форм, принадлежащих приложению.
ProductName	Получает название продукта, связанное с этим приложением.
ProductVersion	Возвращает версию продукта, связанное с этим приложением.
StartupPath	Получает путь для исполняемого файла, запустившего приложение, не включая исполняемое имя.
UserAppDataPath	Получает путь для данных приложения пользователя.
UserAppDataRegistry	Получает раздел реестра для данных приложения пользователя.
UseWaitCursor	Возвращает или задает, используется ли курсор ожидания для всех открытых форм приложения.

Класс Environment предоставляет сведения о текущей среде и платформе, а также необходимые для управления ими средства. Класс Environment используется для получения сведений об аргументах командной строки, коде выхода, значениях переменных среды, содержании стека вызова, времени, прошедшего с момента последней загрузки, и версии среды CLR.

Основные свойства этого класса:

Имя	Описание
CommandLine	Возвращает командную строку для данного процесса.
CurrentDirectory	Возвращает или задает полный путь к текущей рабочей папке.
Is64BitOperatingSystem	Определяет, является ли текущая операционная система 64-разрядной.
Is64BitProcess	Определяет, является ли текущий процесс 64-разрядным.
MachineName	Возвращает имя NetBIOS данного локального компьютера.
NewLine	Возвращает строку, обозначающую в данной среде начало новой строки.
OSVersion	Получает объект OperatingSystem, который содержит идентификатор текущей платформы и номер версии.
ProcessorCount	Возвращает число процессоров на текущем компьютере.
SystemDirectory	Возвращает полный путь к системному каталогу.
TickCount	Возвращает время, истекшее с момента загрузки системы (в миллисекундах).
UserDomainName	Возвращает имя сетевого домена, связанное с текущим пользователем.
UserName	Возвращает имя пользователя, который на данный момент выполнил вход в операционную систему Windows.
WorkingSet	Возвращает объем физической памяти, сопоставленной контексту процесса.

Пример программы «Работа с окружением – WinForms App». Эта программа демонстрирует возможности использования класса `Environment` для получения информации об окружении процесса. Для создания приложения необходимо:



1. Создать проект Windows Forms Application.

2. В форме разместить одно текстовое поле (`tl`) и четыре кнопки.

3. Создать следующие функции обработки событий `Click` для кнопок:

```
private: System::Void button1_Click(System::Object^  sender,
                                    System::EventArgs^  e)
         {String ^d;
```

```
                d = Environment::CurrentDirectory;
                t1->Text = d;
                }
private: System::Void button2_Click(System::Object^  sender,
                            System::EventArgs^  e)
                {String ^d;
                 d = Environment::OSVersion->ToString();
                 t1->Text = d;
                }
private: System::Void button3_Click(System::Object^  sender,
                            System::EventArgs^  e)
                {int proc;
                 proc = Environment::ProcessorCount;
                 t1->Text = "Число процессоров: ";
                 t1->Text += Convert::ToString(proc);
                }
private: System::Void button4_Click(System::Object^  sender,
                            System::EventArgs^  e)
           {String ^d;
            d = Environment::SystemDirectory;
            t1->Text = d;
           }
```

Пример программы «Работа с окружением – Console Application». Консольный аналог предыдущего приложения (тип приложения: CLR Console Application) может иметь такой код:

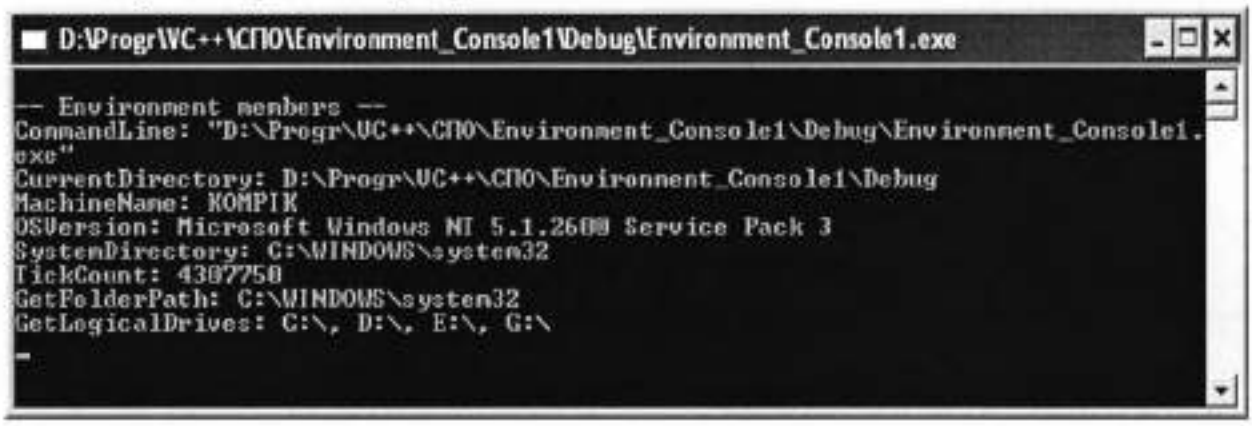
```
#include "stdafx.h"
#include <iostream>

using namespace System;
using namespace System::Collections;
int main()
{
   setlocale(LC_ALL, "Russian");
   Console::WriteLine();
   Console::WriteLine( "-- Environment members --" );

   Console::WriteLine("CommandLine:
{0}",Environment::CommandLine );
   Console::WriteLine("CurrentDirectory: {0}",
                           Environment::CurrentDirectory );
   Console::WriteLine("MachineName: {0}",
Environment::MachineName );
   Console::WriteLine( "OSVersion: {0}",
Environment::OSVersion );
   Console::WriteLine("SystemDirectory: {0}",
                           Environment::SystemDirectory );
   Console::WriteLine( "TickCount: {0}",
Environment::TickCount );
```

```
    Console::WriteLine("GetFolderPath: {0}",
                       Environment::GetFolderPath
                       (Environment::SpecialFolder::Syste
                       m ));
    array<String^>^drives = Environment::GetLogicalDrives();
    Console::WriteLine("GetLogicalDrives: {0}", String::Join(
                       ", ", drives ) );
    Console::ReadLine();
    return 0;
}
```

Результат работы программы:



Класс DriveInfo. Этот класс предоставляет методы и свойства для запроса сведений о диске. DriveInfo позволяет определить, какие диски доступны и какого они типа. Также можно запрашивать сведения для определения объема диска и объема свободного места на диске.

Можно использовать, например, следующие свойства объектов этого класса:

Название свойства	Тип	Описание
DriveFormat	String	Получает имя файловой системы, например NTFS или FAT32.
DriveType	Одно из значений перечисления DriveType, указывающее тип дисков.	Получает тип диска, такой как компакт-диск, съемный, сетевой или жесткий диск. Возможные значения: CDRom, Fixed, Network, NoRootDirectory, Ram, Removable или Unknown. Эти значения описаны в перечислении DriveType.
IsReady	bool	Получает значение, указывающее состояние готовности диска (Значение **true**, если диск готов; значение **false**, если диск не готов).
Name	String	Получает имя диска, например C: \.
RootDirectory	DirectoryInfo	Получение корневого каталога диска.

TotalFreeSpace	long int	Получает общее количество свободного места на диске, доступным в байтах.
TotalSize	long int	Возвращает общий размер дискового пространства на диске, в байтах.
VolumeLabel	String	Получает или задает метку тома диска.

Метод GetDrives() возвращает имена дисков всех логических дисков данного компьютера. Возвращаемое значение (тип: System.IO.DriveInfo[]) – массив типа DriveInfo, представляющий логические диски на компьютере. Этот метод возвращает имена всех логических дисков компьютера. Пользователь может использовать эти сведения для итерационного перебора массива и получить сведения о дисках, используя другие методы класса DriveInfo и свойства.

Следующий пример кода демонстрирует применение класса DriveInfo для отображения информации о логическом диске D.

Пример «**Информация о диске**». Создайте новый проект типа CLR Console Application. Код программы:

```
using namespace System;
using namespace System::IO;

int main()
{DriveInfo ^d = gcnew DriveInfo("D"); //создаем новый
объект
                                      //типа DriveInfo
 if (d->IsReady)  //проверяем состояние готовности диска
 {
    Console::WriteLine("Drive {0}", d->Name);
    Console::WriteLine("Drive type: {0}", d-
>DriveType);
  }
 else
    Console::WriteLine("Drive is not ready!!!");
 Console::ReadLine();
 return 0;
}
```

При создании нового объекта у конструктора в качестве аргумента необходимо указать действительный путь к диску или букву диска. Буквы могут быть как в верхнем, так и в нижнем регистре, от "a" до "z".

Порядок выполнения работы

1. Ознакомиться с теоретическими положениями, приведенными в данных методических указаниях (МУ), а также с конспектом лекций по данной теме.

2. Разработать примеры **«Работа с окружением – WinForms App»** и **«Работа с окружением – Console Application»** по инструкциям, приведенным в пункте «Краткие теоретические положения». Изучить возможности использования класса `Environment` для получения информации о свойствах приложения и системы.

3. Разработать пример **«Информация о диске»** по инструкциям, приведенным в пункте «Краткие теоретические положения». Изучить возможности использования класса `DriveInfo` для получения информации о дисках системы.

4. Разработать программу по своему варианту. Варианты заданий см. в таблицах 1 и 2. Таблица 1 задает тип приложения: консольное или Windows Forms Application. Таблица 2 задает набор необходимых для реализации функций.

Требования к отчету по работе

Отчет по работе должен содержать: название и цель работы; фамилию, инициалы и номер группы студента, выполнившего ЛР; номер варианта; для каждого из заданий – текст задачи по своему варианту, результаты работы программы для разных наборов исходных данных.

Таблица 1. Тип приложения

	Консольное приложение	Windows Forms приложение
Варианты	четные	нечетные

Таблица 2. Функции для реализации

№ варианта / Функции	1	2	3	4	5	6	7	8	9	10	11	12	13	14	14	16	17	18	19	20
Полный путь к программе		+					+													
Каталог, из к-го запущена программа			+					+												
Название и версия программного продукта		+		+				+												
Логич. диск, из к-го запущена пр-ма					+					+	+									

Путь к текущей папке				+			+				+									
Является ли текущ ОС 64-разр.		+								+										
Имя данного локального компьютера	+			+																
Версия ОС			+				+													
Число процессоров на компьютере				+				+												
Системный каталог	+																			
Время с момента загрузки компьютера (в секундах)			+			+					+									
Время с момента загрузки компьютера (в минутах)	+				+					+										
Имя текущего пользователя				+	+			+												
Тип файловой системы на текущем диске		+					+													
Тип текущего диска (CDRom, Fixed и т.д.)		+		+						+										
Корневой каталог текущего диска			+			+	+													
Кол-во свободного места на текущем диске (в Кб)		+			+			+												
Общий р-р текущего логич. диска (в Мб)			+				+													
Метка тома текущего диска	+		+					+												

Тип файловой системы на всех дисках компьютера						+				+										
Тип всех дисков ПК (CDRom, Fixed и т.д.)						+					+									
Кол-во свободного места на всех дисках ПК	+					+					+									
Общий размер всех логич. дисков	+				+					+										
Метки тома всех дисков					+					+	+									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Лабораторная работа 2.
Создание программного обеспечения для работы с файлами

Цель работы: изучение простейших принципов работы с текстовыми файлами в среде программирования MS Visual C++.

Краткие теоретические положения

Файловые операции ввода-вывода можно выполнять после включения в программу заголовка `<fstream>`, в котором определены все необходимые для этого классы и значения.

Открытие и закрытие файла

В C++ файл открывается путем связывания его с потоком. Существуют потоки трех видов: ввода, вывода и ввода-вывода. Чтобы открыть входной поток, необходимо объявить потоковый объект `ifstream`. Для открытия выходного потока нужно объявить поток типа `ofstream`. Поток, который предполагается использовать для операций ввода и вывода должен быть объявлен как объект класса `fstream`. Например:

```
ifstream in;
ofstream out;
fstream f;
```

Создав поток, нужно его связать с файлом. Это можно сделать с помощью функции open(). Ее используют следующим образом:

```
имя_потока.open(имя_файла, режим);
```

- имя_файла может включать путь к файлу.
- режим должен принимать одно или несколько значений перечисления openmode:

```
ios:app
ios:ate
ios:binary
ios:in
ios:out
ios:trunc
```

Назначение каждого из значений:

- **app** - присоединение к концу файла всех выводимых данных (только для файлов, открытых для вывода данных);
- **ate** – поиск будет начинаться с конца файла;
- **binary** - открытие файла в двоичном режиме; по умолчанию все файлы открываются в текстовом режиме. По умолчанию все файлы открываются в текстовом режиме. В текстовом режиме могут происходить некоторые преобразования символов (например, последовательность, состоящая из символов возврата каретки и перехода на новую строку, может

быть преобразована в символ новой строки). При открытии файла в двоичном режиме никакого преобразования символов не происходит. Любой файл можно открыть как в текстовом, так и в двоичном режиме. Единственное различие между этими режимами состоит в преобразовании (или нет) символов.

- **in** - открытие файла для ввода данных (обычно это значение по умолчанию);
- **out** – открытие файла для вывода данных (обычно это значение по умолчанию);
- **trunc** – при открытии уже существующего файла его содержимое удаляется.

Пример создания и открытия обычного выходного файла:

```
ofstream out;
out.open("test1.txt");
```

Так как параметр `mode` функции `open()` по умолчанию устанавливается равным значению, соответствующему типу открываемого потока, в предыдущем примере нет необходимости задавать его значение.

Можно проверить успешность выполнения предыдущей операции по созданию и открытию файла следующим образом:

```
if(!out)
   {cout<< "НЕ удается открыть файл.\n";
    //return 1;
    }
```

Прежде чем выполнять какие-либо дальнейшие операции с файлом, всегда следует проверять результат выполнения функции `open()`.

Можно также проверить факт успешного открытия файла с помощью функции `is_open()`:

```
if(!out.is_open())
   {cout<< "файл не открыт!\n";
    //return 1;
    }
```

Производить операцию открытия файла можно также так:

```
ofstream out1("test1.txt");
ofstream out2("test2.txt",ios::out);
```

Параметры у конструкторов классов `ifstream`, `ofstream`, `fstream` совпадают с соответствующими параметрами функции `open()`.

Чтобы закрыть файл после использования необходимо вызвать функцию `close()`, например:

```
out1.close();
```

Пример создания и открытия файла:

```
#include <fstream>
using namespace std;

int main()
{ofstream out1("test1.txt");
 //или так:
 //ofstream out1("test1.txt",ios::out);
 out1.close();
 return 0;
}
```

Чтение и запись текстовых файлов

Проще всего считывать данные из текстового файла или записывать их в него с помощью операторов << и >>. Например, в следующей программе выполняется запись в файл целого числа и строки (**Пример 1**):

```
#include <fstream>
using namespace std;

int main()
{ofstream out1("test1.txt");
 if(!out1)
   {cout<< "НЕ удается открыть файл.\n";
   return 1;
   }
 out1 << "Это пример ФАЙЛА. \n";
 out1 << 10;
 out1.close();
 cout<< "Файл test1.txt успешно создан.\n";
 return 0;
}
```

Следующая программа считывает строку и целое число из файла, созданного при выполнении предыдущей программы (**Пример 2**):

```
#include <fstream>
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{int x; char str[80];
 ifstream in("test1.txt");
```

```
    if(!in)
      {cout<< "НЕ удается открыть файл.\n";
      getch();
      return 1;
      }
    in >> str;  in >> x;
    cout << str << "  " << x << "\n";
    in.close();
    getch();    return 0;
   }
```

Следует иметь в виду, что при использовании оператора >> для считывания данных из текстовых файлов пробельные символы могут опускаться, кроме того, ввод данных прекращается при обнаружении первого пробельного символа.

Пример 3. Чтение файла с целыми числами в цикле. Поиск суммы чисел, записанных в файле.

```
#include <conio.h>
#include <iostream>
#include <fstream>
using namespace std;

int main(void)
{int n, max, k=0, s=0;

setlocale(LC_ALL, "Russian");

ifstream in("Числа.txt");
if(!in) {cout<< "НЕ удается открыть файл.\n";
        getch();
        return 1;}
while (!in.eof())
    {in>>n; k++; s+=n;
     }
in.close();

cout<< "Всего " << k << " чисел. \n";
cout<< "Их сумма равна " << s << ". \n";
getch();
```

```
return 0;
}
```

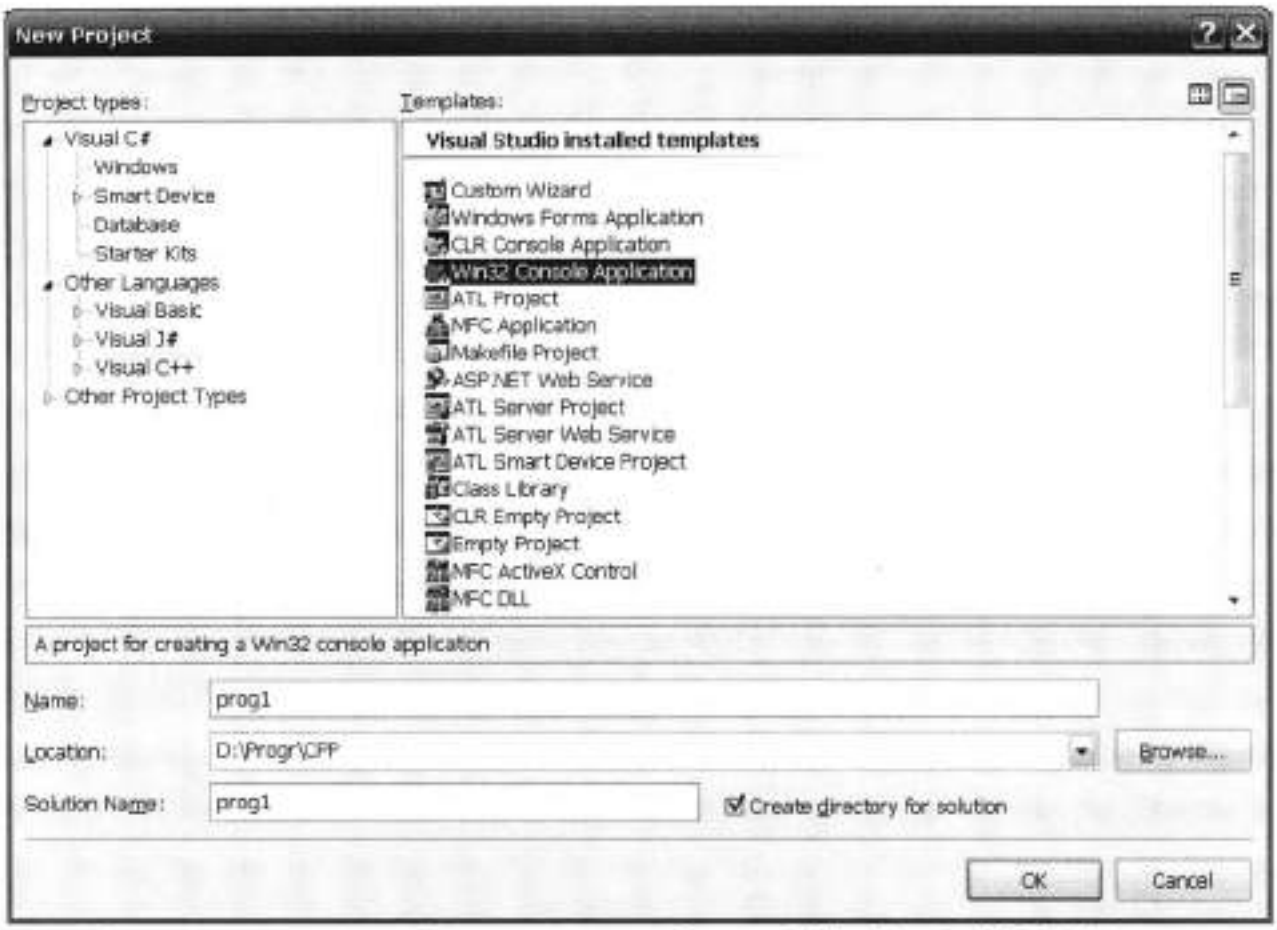
Порядок выполнения работы

1. Ознакомиться с теоретическими положениями, приведенными в данных методических указаниях (МУ), а также с конспектом лекций по данной теме.

2. **Создание программы по примеру 1.** Самыми простыми с точки зрения пользовательского интерфейса являются классические консольные приложения, то есть такие приложения, которые для ввода и вывода информации используют консоль - т.е. дисплей, работающий в текстовом режиме. Взаимодействие с консольным приложением в современных операционных системах происходит через консольное окно – окно для работы с командной строкой.

Для того, чтобы создать консольное приложение на языке C++, выберем в IDE-среде в меню File следующую последовательность опций: **New->Project-> ->Visual C++ -> Win32 Console Application**. После этого определяем имя проекта в поле **Name** и местонахождение каталога с файлами проекта в поле **Location**. Отметим, что имя проекта является одновременно и именем каталога и подкаталога, которые будут созданы средой разработки. Например, если мы ввели имя проекта `prog1` и местоположение `D:\Progr\CPP`, то файлы проекта и подкаталог Debug будут находиться в каталоге `D:\Progr\CPP\prog1` (см. рис.).

После нажатия **OK** в окне **New Project** появится окно **Win32 Application Wizard**, в котором необходимо выбрать пункт **Application Settings**. В правой половине окна появится группа селективных кнопок **Application Type** для указания типа приложения и две группы кнопок-переключателей. Для консольного проекта необходимо выбрать селективную кнопку **Console application** и переключатель **Empty Project**, а затем нажать кнопку **Finish**.



В интегрированной среде разработки Visual Studio.NET проекты логически организуются в решения (solutions). После компиляции и линкования проекта мы получаем исполнимый модуль, называемый сборкой (например, модуль библиотеки DLL). Каждое решение состоит из одного или нескольких проектов. В свою очередь, каждый проект может состоять из любого количества исходных файлов, ссылок на внешние сборки (которые используются этим проектом) и прочих ресурсов, которые и образуют приложение.

После создания пустого проекта Prog1 в него необходимо добавить исходные файлы программы. Для этого выполним в окне менеджера проектов (**Solution Explorer**) выделим имя проекта (**Prog1**) или имя подраздела для исходных файлов (**Source Files**) и в контекстном меню выберем пункты **Add | Add New Item.** Далее выбираем **Code | C++ File (*.cpp)** и укажем имя файла Hello. (Отметим, что нажатие клавиши Del при выделенной пиктограмме файла приводит к удалению файла из проекта, при этом *файл удаляется и из каталога*, в котором располагаются файлы проекта.)

Далее в появившемся окне наберем указанный код программы из **примера 1**.

Теперь попробуем запустить программу на выполнение. Чтобы приложение заработало, его надо откомпилировать, то есть перевести написанное на языке C++ в машинные коды. В интегрированную среду Visual Studio встроен компилятор языка C++. Для компиляции и запуска новой программы требуется либо выбрать команду **Debug|Start Debugging**, либо

нажать «горячую» клавишу **F5**. При выполнении этой команды интегрированная среда Visual Studio вначале запускает компилятор, а после его успешной работы запускает на выполнение нашу программу.

Если Вы правильно создали проект и верно ввели весь исходный код, то на экране откроется окно черного цвета со строкой `"Файл test1.txt успешно создан."` Для завершения просмотра результатов нужно нажать любую клавишу на клавиатуры. После этого с помощью любого файлового менеджера перейдите в папку с созданным проектом и убедитесь, что файл `test1.txt` действительно создан. Содержимое файла можно просмотреть с помощью, например, Блокнота или другого текстового редактора.

Если же интегрированная среда выдает сообщение об ошибках, то необходимо внимательно проверить текст программы, который Вы ввели в окне редактора, исправить ошибки и заново запустить программу на выполнение.

3. **Проверить работу примеров 2-3**, приведенных в тексте МУ. Каждую из программ сохранять на диске компьютера.

4. **Разработать программу по своему варианту.** Варианты заданий см. в таблице.

Требования к отчету по работе

Отчет по работе должен содержать: название и цель работы; фамилию, инициалы и номер группы студента, выполнившего ЛР; номер варианта; текст задачи по своему варианту, код программы с пояснениями, схема алгоритма работы программы, результаты работы программы для разных наборов исходных данных (не менее двух).

Защита лабораторной работы

При защите данной лабораторной работы необходимо продемонстрировать и объяснить результаты выполнения всех заданий, предъявить отчет по работе и ответить на контрольные вопросы.

Варианты заданий

Вар.	Задание
1	1. Программа просит ввести целое положительное число M. Затем программа открывает текстовый файл, который содержит некоторое количество целых чисел. Далее создается новый файл, в который записываются те числа из открытого файла, которые делятся на M.
2	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Найти сумму минимального и

	максимального из чисел, записанных в файле, и дописать эту сумму в конец имеющегося файла.
3	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Найти разность первого и максимального компонентов файла. Результат записать в новый файл.
4	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Далее создается новый файл, в который записываются все числа из открытого файла за исключением тех, которые кратны К.
5	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Создаются два новых пустых файла. Все числа, меньшие числа К, записываются в один файл, а числа, большие К – в другой файл.
6	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Создаются два новых пустых файла. Все элементы файла с нечетными номерами записываются в один файл, а элементы с четными номерами – в другой файл.
7	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Найти разность первого и последнего компонентов файла и записать результат в конец открытого файла.
8	1. Программа просит ввести целое положительное число N. Затем программа открывает текстовый файл, который содержит некоторое количество целых чисел. Далее создается новый файл, в который записываются те числа из открытого файла, которые не делятся на N.
9	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Найти разность последнего и минимального компонентов файла. Результат записать в конец имеющегося файла.
10	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Подсчитать количество положительных и отрицательных чисел. Результат вывести в новый файл.
11	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Создать новый файл из четных чисел исходного файла. Если таких чисел нет, то вывести сообщение об этом факте.
12	1. Программа открывает текстовый файл, который содержит некоторое количество целых чисел. Выяснить, какие числа встречаются чаще — положительное или отрицательное.

Лабораторная работа 3.
Создание программного обеспечения для работы с файлами и каталогами

Цель работы: изучение возможностей использования классов `Directory`, `DirectoryInfo`, `File`, `FileInfo` в среде программирования MS Visual C++.

Краткие теоретические положения

`Directory`, `DirectoryInfo` - Эти классы используются для манипуляций структурой каталогов машины. Тип Directory представляет функциональность, используя статические члены. Тип DirectoryInfo обеспечивает аналогичную функциональность через действительную объектную ссылку.

`DriveInfo` - Этот класс предоставляет детальную информацию относительно дисковых устройств, используемых данной машиной.

`File`, `FileInfo` - Эти классы служат для манипуляций множеством файлов данной машины. Тип File представляет функциональность через статические члены. Тип FileInfo обеспечивает аналогичную функциональность через действительную объектную ссылку.

Классы **`Directory`** и **`File`** содержат только статические методы, а их экземпляры никогда не создаются. Для их использования нужно просто предоставить путь к интересующему объекту файловой системы при каждом вызове метода-члена. Если требуется выполнить только одну операцию в отношении папки или файла, то применение этих классов является наиболее эффективным подходом, поскольку позволяет избегать накладных расходов, связанных с созданием экземпляров.

Классы **`DirectoryInfo`** и **`FileInfo`** реализуют практически те же общедоступные методы, что и **`Directory`** и **`File`**, а также некоторые общедоступные свойства и конструкторы, но поддерживают состояние, а их члены не являются статическими. Сначала понадобится создать их экземпляр, который затем ассоциировать с определенной папкой или файлом. Это означает, что применение этих классов будет более эффективным подходом, если необходимо выполнять множество операций в отношении одного и того же объекта. Дело в том, что во время создания они считывают информацию об аутентификации и прочие сведения о соответствующем объекте файловой системы и больше не нуждаются в ее повторном чтении, сколько бы методов не вызывалось для данного объекта (экземпляра класса). В отличие от этого, классы, не поддерживающие состояние, нуждаются в проверке деталей, связанных с файлом или папкой, при каждом вызове какого-либо метода.

Класс **`File`** предоставляет, например, следующие методы.

Имя	Описание
AppendAllLines	Добавляет строки в файл, затем закрывает файл. Если указанный файл не существует, этот метод создает файл, записывает указанные строки в файл и затем закрывает файл.
AppendText	Создает объект StreamWriter, который добавляет текст с кодировкой UTF-8 в существующий файл, или в новый файл, если указанный файл не существует.
Copy	Копирует существующий файл в новый файл. Перезапись файла с тем же именем не разрешена.
Create	Создает или перезаписывает файл по указанному пути.
Delete	Удаляет указанный файл.
Exists	Определяет, существует ли заданный файл.
GetAttributes	Получает значение FileAttributes для файла в пути.
GetCreationTime	Возвращает дату и время создания заданного файла или каталога.
GetLastAccessTime	Возвращает время и дату последнего обращения к указанному файлу или каталогу.
GetLastWriteTime	Возвращает время и дату последней операции записи в указанный файл или каталог.
Move	Перемещает заданный файл в новое местоположение и разрешает переименование файла.
Open	Открывает объект FileStream по указанному пути с доступом для чтения и записи.
ReadAllBytes	Открывает двоичный файл, считывает содержимое файла в массив байтов и затем закрывает файл.
ReadAllLines	Открывает текстовый файл, считывает все строки файла и затем закрывает файл.
ReadAllText	Открывает текстовый файл, считывает все строки файла и затем закрывает файл.
ReadLines	Считывает строки файла.
Replace	Заменяет содержимое заданного файла на содержимое другого файла, удаляя исходный файл и создавая резервную копию замененного файла.
WriteAllBytes	Создает новый файл, записывает в него указанный массив байтов и затем закрывает файл. Если целевой файл уже существует, он будет перезаписан.
WriteAllLines	Создает новый файл, записывает в него коллекцию строк, затем закрывает файл.
WriteAllText	Создает новый файл, записывает в него указанную строку

	и затем закрывает файл. Если целевой файл уже существует, он будет перезаписан.

Многие из методов `File` возвращают другие типы ввода-вывода при создании или открытии файла. Эти другие типы можно использовать для дальнейших операций с файлом. Дополнительные сведения смотрите в разделах справочной системы, посвященных определенным элементам `File`, таким как `OpenText`, `CreateText` или `Create`.

Пример 1. Простейшая программа, которая проверяет, существует ли заданный файл:

```
#include "stdafx.h"

using namespace System;
using namespace System::IO;

int main()
{String ^path = "c:\\temp\\Test.txt";
if (File::Exists(path))
   Console::WriteLine("Файл существует!");
else
   Console::WriteLine("Файл не существует!");
Console::ReadLine();
return 0;
}
```

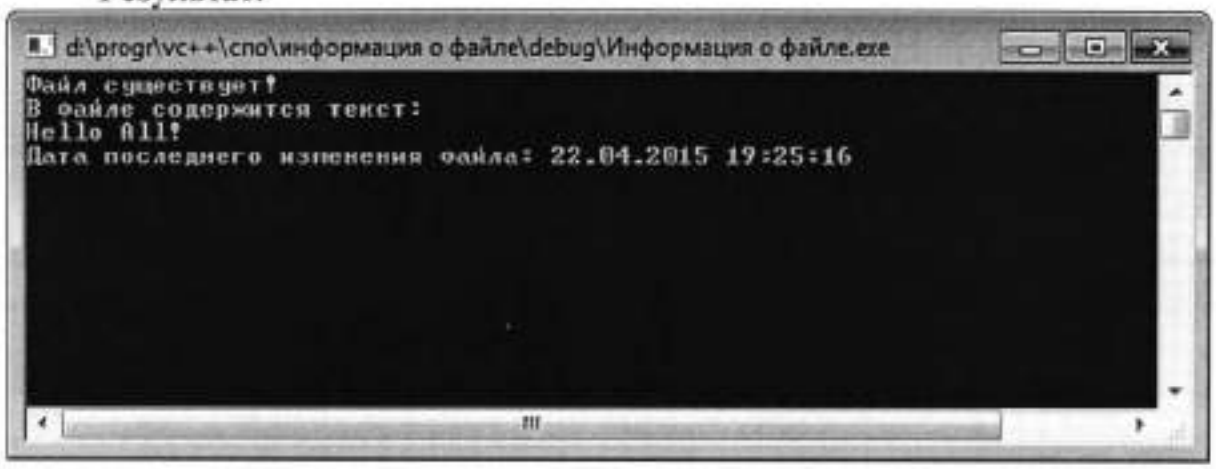
Усовершенствуем программу, что бы она выводила содержимое текстового файла и дату его изменения:

```
#include "stdafx.h"
using namespace System;
using namespace System::IO;

int main()
{String ^path = "c:\\temp\\Test.txt";

if (File::Exists(path))
  {Console::WriteLine("Файл существует!");
   Console::WriteLine("В файле содержится текст:");
   Console::WriteLine("{0}", File::ReadAllText(path));
   Console::WriteLine("Дата последнего изменения файла:
                       {0}",
                       File::GetLastWriteTime(path));
  }
else
  Console::WriteLine("Файл не существует!");
Console::ReadLine();
return 0;
}
```

Результат:



Класс **FileInfo** предоставляет следующие свойства и методы (в таблице указаны не все доступные свойства и методы. Полный список возможностей класса FileInfo можно посмотреть в справочной системе.)

Свойства

Directory	возвращает папку типа DirectoryInfo, в которой находится данный файл
DirectoryName	возвращает название папки, в которой лежит текущий файл
IsReadOnly	позволяет определить является ли наш файл доступным только для чтения
Exists	возвращает True, если файл существует
LastWriteTime	время последней модификации
Length	определяет размер данного файла
Attributes	возвращает атрибуты файла
CreationTime	возвращает время создания файла

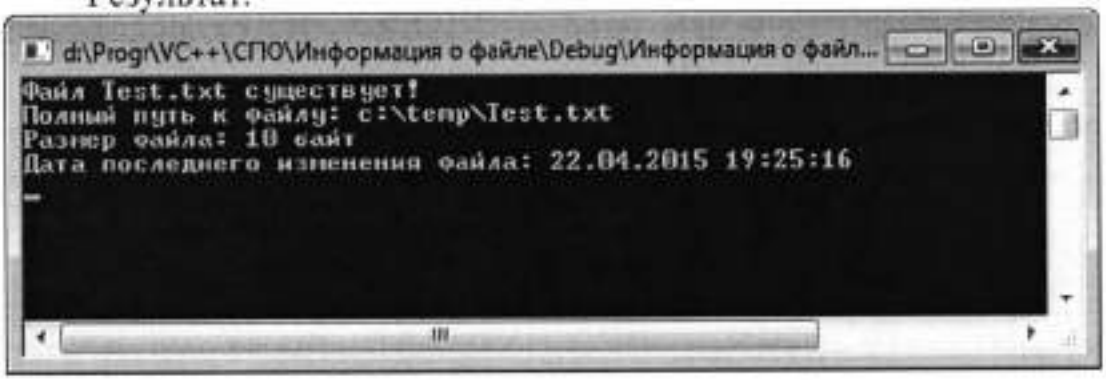
Методы

Name	Description
AppendText	Creates a StreamWriter that appends text to the file represented by this instance of the FileInfo.
CopyTo	Copies an existing file to a new file, disallowing the overwriting of an existing file.
Create	Creates a file.
Delete	Permanently deletes a file. (Overrides FileSystemInfo::Delete().)
MoveTo	Moves a specified file to a new location, providing the option to specify a new file name.

Пример 2. Фрагмент программы, которая проверяет, существует ли заданный файл:

```
//вариант с использование класса FileInfo
FileInfo ^f = gcnew FileInfo(path);
if (f->Exists)
{Console::WriteLine("Файл {0} существует!", f->Name);
 Console::WriteLine("Полный путь к файлу: {0}", f-
>FullName);
 Console::WriteLine("Размер файла: {0} байт", f->Length);
 Console::WriteLine("Дата последнего изменения файла: {0}",
                          f->LastWriteTime);
    }
else
    Console::WriteLine("Файл не существует!");
```

Результат:



Класс **Directory** предоставляет статические методы для создания, перемещения и перечисления в каталогах и вложенных каталогах. Перечислим некоторые методы класса Directory.

Имя	Описание
CreateDirector	Создает все каталоги и подкаталоги, по указанному пути.
Delete(String)	Удаляет пустой каталог по заданному пути.
Delete	Удаляет заданный каталог и, при наличии соответствующей инструкции, все подкаталоги и файлы в нем.
EnumerateDirectories	Возвращает перечисляемую коллекцию имен каталогов, расположенных по указанному пути.
EnumerateFiles	Возвращает перечисляемую коллекцию имен файлов, расположенных по указанному пути.
EnumerateFile-SystemEntries	Возвращает перечисляемую коллекцию элементов файловой системы, расположенных по указанному пути.
Exists	Определяет, указывает ли заданный путь на существующий каталог на диске.
GetCreationTime	Получает дату и время создания каталога.

GetCurrentDirectory	Получает текущий рабочий каталог приложения.
GetDirectories	Получает имена подкаталогов (включая их пути) в указанной директории.
GetDirectoryRoot	Возвращает для заданного пути сведения о томе и корневом каталоге по отдельности или сразу.
GetFiles	Возвращает имена файлов (с указанием пути к ним) в указанном каталоге.
GetFileSystemEntries	Возвращает имена всех файлов и подкаталогов из указанного каталога.
GetLastAccessTime	Возвращает время и дату последнего обращения к указанному файлу или каталогу.
GetLastWriteTime	Возвращает время и дату последней операции записи в указанный файл или каталог.
GetLogicalDrives	Извлекает имена логических устройств данного компьютера в формате "<имя устройства>:\".
GetParent	Извлекает родительский каталог, на который указывает абсолютный или относительный путь.
Move	Перемещает файл или каталог со всем его содержимым в новое местоположение.
SetCurrentDirectory	Устанавливает заданный каталог в качестве текущего рабочего каталога приложения.

Класс Directory позволяет выполнять такие стандартные операции, как создание, копирование, перемещение, переименование и удаление папок. Класс Directory также позволяет получать и задавать сведения о типе DateTime, относящиеся к созданию, доступу и записи каталога.

Все методы Directory статические, поэтому если необходимо выполнить только одно действие, более эффективным может оказаться использование метода Directory, а не соответствующего экземпляра метода DirectoryInfo. Большинство методов класса Directory требуют указывать путь к каталогу, с которым работает пользователь.

Класс **DirectoryInfo** также позволяет выполнять такие стандартные операции, как создание, копирование, перемещение, переименование и удаление папок.

Основные свойства и методы этого класса

Свойства

Имя	Описание
Attributes	Получает или задает атрибуты для текущего файла или каталога.
CreationTime	Получает или задает время создания текущего файла или каталога.
Exists	Получает значение, определяющее наличие каталога.
Extension	Получает строку, содержащую расширение файла.

FullName	Получает полный путь к каталогу или файлу
LastAccessTime	Получает или задает время последнего доступа к текущему файлу или каталогу.
LastWriteTime	Получает или задает время последней операции записи в текущий файл или каталог.
Name	Получает имя данного экземпляра DirectoryInfo.
Parent	Получает родительский каталог заданного подкаталога.
Root	Получает корневой элемент пути.

Методы

Имя	Описание
Create()	Создает каталог.
CreateSubdirectory	Создает один или несколько подкаталогов по заданному пути. Путь может быть задан относительно текущего экземпляра класса DirectoryInfo.
Delete()	Удаляет этот DirectoryInfo, если он пуст.
EnumerateDirectories()	Возвращает перечисляемую коллекцию сведений о каталогах в текущем каталоге.
EnumerateFiles()	Возвращает перечисляемую коллекцию сведений о файлах в текущем каталоге.
EnumerateFileSystemInfos()	Возвращает перечисляемую коллекцию сведений о файловой системе текущего каталога.
Equal	Определяет, равен ли заданный объект текущему объекту.
GetDirectories()	Возвращает подкаталоги текущего каталога.
GetFiles()	Возвращает список файлов текущего каталога.
GetFileSystemInfos()	Возвращает массив строго типизированных объектов FileSystemInfo, представляющих все файлы и подкаталоги в том или ином каталоге.
MoveTo	Перемещает экземпляр DirectoryInfo и его содержимое в местоположение, на которое указывает новый путь.
Refresh	Обновляет состояние объекта.
ToString	Возвращает исходный путь, переданный пользователем.

Порядок выполнения работы

1. Ознакомиться с теоретическими положениями, приведенными в данных методических указаниях (МУ), а также с конспектом лекций по данной теме.

2. Разработать **пример 1** (с классом `File`) по инструкциям, приведенным в пункте «Краткие теоретические положения». Изучить возможности использования класса `File`.

3. Разработать **пример 2** (с классом `FileInfo`) по инструкциям, приведенным в пункте «Краткие теоретические положения». Изучить возможности использования класса `FileInfo`.

4. Разработать программу по своему варианту. Варианты заданий см. в таблицах 1 и 2. Таблица 1 задает тип приложения: консольное или Windows Forms Application. Таблица 2 задает набор необходимых для реализации функций.

Требования к отчету по работе

Отчет по работе должен содержать: название и цель работы; фамилию, инициалы и номер группы студента, выполнившего ЛР; номер варианта; для каждого из заданий – текст задачи по своему варианту, результаты работы программы для разных наборов исходных данных; краткое описание использованных методов и свойств классов `Directory`, `DirectoryInfo`, `File`, `FileInfo`.

Таблица 1. Тип приложения

	Консольное приложение	Windows Forms приложение
Варианты	нечетные	четные

Таблица 2. Функции для реализации

№ варианта / **Функции**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Создание текстового файла в текущей папке					+					+				+				+		
Создание текстового файла в произвольной папке				+			+						+						+	
Добавление указанной строки в текстовый файл	+								+						+		+			+

Чтение текстового файла в текущей папке		+				+					+					+			
Чтение текстового файла в произвольной папке			+					+						+					+	
Копирование файла в текущей папке			+		+			+		+						+				+
Копирование файла по указанному пути	+					+					+	+			+			+		
Удаление указанного пользователем файла		+		+			+		+				+			+				
Вывод списка всех файлов в текущей папке			+							+		+				+				+
Вывод списка всех файлов в указанной папке							+		+						+				+	
Вывод даты создания всех файлов в текущей папке						+							+					+		
Вывод имени и размера всех файлов в указанной папке					+			+						+			+			
Вывод имени и даты изменения всех файлов в папке, родительской по отношению к текущей	+										+				+					
Вывод полной информации о запрошенном пользователем файле		+		+								+								

Вывод списка всех каталогов в текущем каталоге		+	+					+					+			+				
Вывод списка всех каталогов в корневом каталоге текущего диска	+										+						+			+
Вывод имен и дат создания всех каталогов в указанном каталоге				+			+		+					+				+		
Вывод списка всех каталогов в родительском каталоге с указанием даты создания					+	+				+		+							+	
Создание каталога	+	+			+		+	+	+			+			+					+
Удаление каталога			+	+		+				+	+			+			+			
Определение суммарного размера всех файлов в указанном каталоге						+			+				+					+		
Поиск файла в текущем каталоге	+						+			+							+			
Поиск файла в указанном каталоге		+	+												+				+	
Поиск подкаталога по имени в указанном каталоге					+			+			+							+		
Перейти из текущего каталога на один уровень вверх				+									+			+			+	
Сменить текущий каталог												+		+			+			+
№ варианта	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**

Лабораторная работа 4.
Работа со строками

Цель работы: изучение принципов работы со строками, а также повторение методов работы с текстовыми файлами в среде программирования MS Visual C++.

Краткие теоретические положения

Использование одномерных массивов для хранения строк

Одномерные массивы могут использоваться для хранения символьных строк. В C++ строка – это символьный массив, который завершается нулевым символом ('\0'). При определении длины символьного массива необходимо учитывать признак завершения строки и задавать длину массива на 1 больше длины самой строки.

Строку-массив можно инициализировать при объявлении строковым литералом (списком символов, заключенным в кавычки). Например:

```
char str[12]="Hello All!\n";
cout << str;
```

В конец строки не нужно добавлять нулевые символы. Компилятор C++ сделает это самостоятельно. Следовательно, строка `"Hello All!\n"` будет размещаться в памяти, как указано на рисунке:

Номер символа	0	1	2	3	4	5	6	7	8	9	10	11
Символ	H	e	l	l	o		A	l	l	!	\n	'\0'

Обработка строк обычно связана с перебором всех символов от начала до конца строки. Признаком конца такого перебора является обнаружение нулевого символа. Например, можно использовать оператор цикла `while` для просмотра всех элементов строки:

```
while(str[i]) {// что-нибудь делать      }
```

В этом примере цикл повторяет свое выполнение, пока `str[i]` не получит нулевого значения.

Пример. Программа вводит символьную строку и обращает ее (меняет порядок символов на обратный).

```
char c, str[80];
int i;
cout << "Enter a string: ";
gets(str);
for (i=0; i<=(strlen(str)-1)/2; i++)
    {c=str[i];
     str[i]=str[strlen(str)-i-1];
     str[strlen(str)-i-1]=c;
     }
puts(str);
```

Считывание строк с клавиатуры

Проще всего в консольном приложении считать строку с клавиатуры, создав массив, который примет эту строку, и использовав инструкцию cin >>. Например:

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{char str[80]; cout << "Enter a string: ";
 cin >> str; cout << "This string was entered: ";
 cout << str; getch();  return 0;
 }
```

Однако, несмотря на то, что эта программа формально корректна, она не лишена недостатков. Рассмотрим следующий вариант ее применения:

```
Enter a string: Hello All!
This string was entered: Hello
```

При выводе строки, введенной с клавиатуры, программа отображает только слово «Hello», а не всю строку. Дело в том, что оператор «>>» прекращает считывать строку, как только встречает символ пробела, табуляции или новой строки (эти символы иногда называются *пробельными*). Для решения этой проблемы можно использовать библиотечную функцию `gets()`, а в качестве аргумента передавать ей имя массива. Функция `gets()` считывает строку до тех пор, пока пользователь при вводе строки не нажмет клавишу "Enter".

```
#include <iostream>
#include <conio.h>
#include <cstdio>
using namespace std;

int main()
{char str[80]; cout << "Enter a string: ";
 //cin >> str;
 gets(str); cout << "This string was entered: ";
 cout << str;  getch();  return 0;
 }
```

Эта программа будет работать корректно.

При использовании оператора «>>» и функции `gets()` следует иметь в виду, что они не выполняют проверки на отсутствие нарушения границ массива. Поэтому, если пользователь введет строку, длина которой превышает размер массива, возможна некорректная работа программы и даже ее «зависание».

Некоторые библиотечные функции для обработки строк

Определение длины строки `str` выполняет функция `strlen(str)`.

Например: `int k = strlen(str);`

Копирование строк выполняет функция `strcpy (to, from)`.

Содержимое строки `from` копируется в строку `to`. Например:

```
#include <iostream>
#include <conio.h>
#include <cstring>
using namespace std;

int main()
{char str[80];   strcpy(str, "Hello!");
 cout << str;    getch();
 return 0;
 }
```

Объединение строк выполняет функция `strcat(s1, s2)`, которая присоединяет строку `s2` к `s1`. При этом строка s2 не изменяется. Обе строки должны завершаться нулевым символом. Результат вызова этой функции, то есть результирующая строка s1 также будет завершаться нулевым символом. Пример:

```
#include <iostream>
#include <conio.h>
#include <cstring>
using namespace std;

int main()
{char s1[20],s2[10]; strcpy(s1, "Hello ");
 strcpy(s2, "All!\n");    strcat(s1,s2);
 cout << s1;     getch();  return 0;
 }
```

Сравнение строк выполняет функция `strcmp(s1, s2)`, которая сравнивает строку `s2` и `s1` и возвращает целый результат, равный 0, если они равны, и другое значение, если строки не равны.

```
//Пример 1 чтения файла с символами.
//Подсчет количества букв 'й' в файле.
#include <conio.h>
#include <iostream>
#include <fstream>
using namespace std;

int main(void)
{int k=0,s=0; char c;

setlocale(LC_ALL, "Russian");
```

```
ifstream in("Буквы.txt");
 if(!in) {cout<< "НЕ удается открыть файл.\n"; return 1; }
while (!in.eof()) {in>>c; if (c == 'й') k++;
                    s++;
                      }
in.close();
cout<< "Всего " << s << " символов в файле. \n";
cout<< "Из них - " << k << " букв 'й'. \n";
getch(); return 0;
}
```

// Пример 2. Программа посимвольного копирования файлов.

```
#include <conio.h>
#include <iostream>
#include <fstream>
using namespace std;

int main(void)
{int s=0; char c;

setlocale(LC_ALL, "Russian");

ifstream f1("Пример1.txt", ios::binary);
ofstream f2("Пример2.txt", ios::binary);
if(!f1 || !f2)  {cout<< "НЕ удается открыть файл.\n";
                  return 1;
                  }
while (!f1.eof())
   {c=f1.get();
    if (f1.eof()) break; //Иначе посл.символ исходного файла
    f2 << c;             //повторяется лишний раз в вых.
файле
    // а можно так:  f2.put(c);
      s++;
     }
f1.close();
f2.close();
cout<< "Всего " << s << " символов в файле. \n";
getch(); return 0;
}
```

Работа с формами

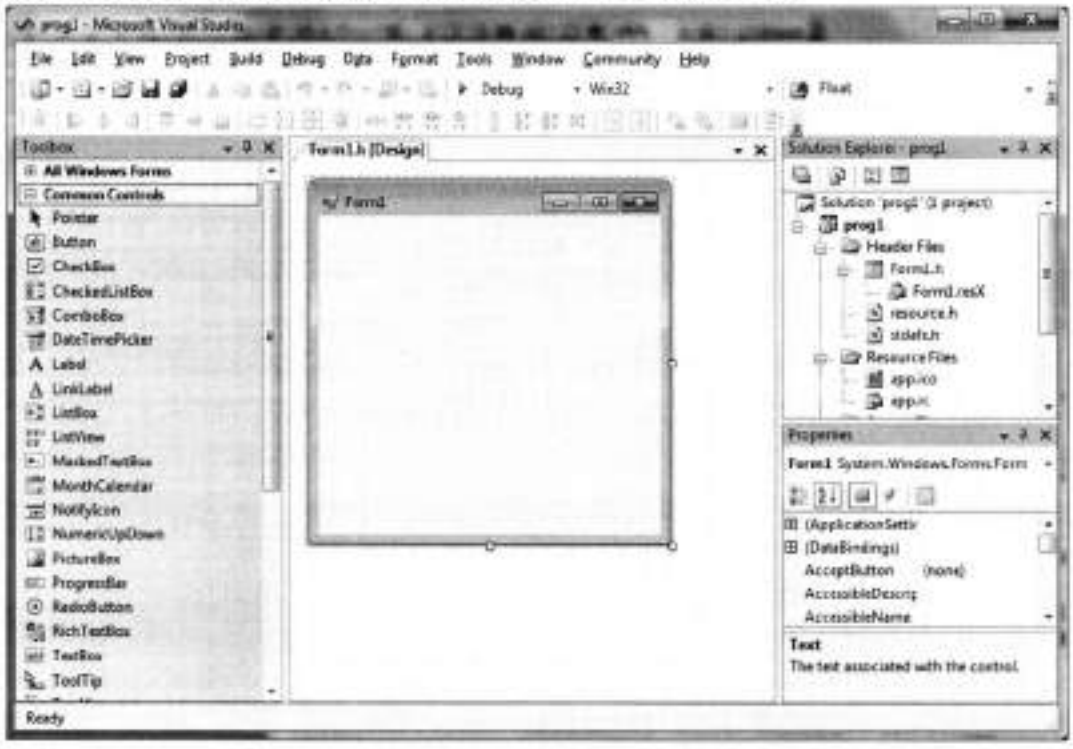
Интегрированная среда разработки приложений Visual Studio.NET является удобной средой для разработки как консольных приложений, так и приложений MS Windows, имеющих стандартный графический интерфейс, известный также как GUI - интерфейс (GUI - Graphic User Interface). В интегрированной среде программирования Visual C++ (IDE VC++)

приложения, имеющие графический интерфейс называются *Windows Forms Application*.

Основным компонентом среды визуального программирования является форма (Form). Форма – это *контейнер*, в который помещаются остальные элементы приложения, определяющие впоследствии всю функциональность приложения. В форме можно разместить различные визуальные *компоненты* (элементы управления), такие как кнопка, текстовое поле для ввода данных, список, поле со списком и т.д. Для работы формами используется *дизайнер форм*.

При создании нового проекта типа Windows Forms Application автоматически создается форма Form1. Вместе с ней создается специальный программный модуль с именем формы и расширением .h (Form1.h). В этом модуле находится описание формы и в нем же будут располагаться обработчики событий компонентов.

VC++ оформляет создаваемое приложение в виде двух контейнеров, вложенных один в другой. Один (главный контейнер) называется ***Solution*** (решение), а другой – ***Project*** (проект). Проект является каркасом или контейнером, объединяющим группу файлов. В рамках проекта создается программа, подлежащая исполнению, то есть откомпилированная и построенная. Каждый проект содержит по крайней мере две подконфигурации: отладочную и обычную (исполняемую). Проекты входят в Solution (решение или групповой проект), которое может содержать множество объединенных проектов. В простейшем случае Solution содержит один проект. Для работы с групповым проектом используется инструментальное окно **Solution Explorer**. Следует отметить, что не все файлы проекта отображаются в окне **Solution Explorer**.



Основные окна Visual Studio при создании нового проекта

Для создания приложения с графическим интерфейсом в интегрированной среде Visual Studio 2005/2008 необходимо выбрать в меню File следующую последовательность опций: **New -> Project -> Visual C++ -> Windows Forms Application**. После этого определяем имя проекта в поле **Name** и местонахождение каталога с файлами проекта в поле **Location** и нажимаем **OK**. В результате этих действий создается заготовка приложения типа **Windows Forms Application**.

Основную часть экрана займет окно *дизайнера форм* (закладка с названием **Form1.h [Design]**), в нем будет открыто окно с заголовком `Form1` – заготовка *формы* приложения. В форме можно разместить различные *компоненты* (элементы управления), такие как кнопка, текстовое поле для ввода данных, список, поле со списком и т.д. В общем случае приложение может иметь много форм.

Компоненты, размещаемые в форме, позволяют пользователю программы выполнять различные действия, например, щелкать на кнопках или вводить данные. Назовем некоторые компоненты: `Label` – надпись, `Button` – кнопка, `TextBox` – текстовое поле (позволяет пользователю вводить новый или редактировать имеющийся текст), `CheckBox` – флажок для включения или отключения опций, `RadioButton` – переключатель, позволяет выбирать один элемент из группы. Для добавления в формы проекта различных компонентов используется панель **ToolBox**. Доступ к ней можно получить с помощью меню **View | Toolbox**.

Свойства компонентов. Все компоненты, размещаемые в формах, имеют уникальные имена. Имя компонента, как известно, задается с помощью свойства `Name`. По умолчанию в свойстве `Name` устанавливается значение, равное типу элемента управления с порядковым номером в конце. Так, например, для первой формы в проекте свойство `Name` будет равно `Form1` и т.д. Можно изменить имя компонента во время работы над приложением.

В окне **Properties** (свойства) можно просматривать и изменять *свойства компонентов*, в том числе и форм. Каждый объект имеет свой собственный набор описывающих его свойств. Некоторые свойства есть почти у всех компонентов: `Name` (Имя), `Left` (позиция элемента управления относительно левого края его контейнера), `Top` (позиция элемента относительно верхнего края его контейнера), `Height` (высота), `Width` (ширина), `Enabled` (определяет, можно ли пользователю работать с этим

компонентом), `Visible` (видимость во время выполнения программы), `Text` (подпись), `Font` (шрифт) и т.д.

В режиме проектирования манипулирование свойствами осуществляется с помощью Дизайнера Форм или в окне **Properties**. Например, для того чтобы изменить свойства `Height` (высоту) и `Width` (ширину) кнопки, достаточно "зацепить" мышкой за любой ее угол и раздвинуть до нужного представления. Того же результата можно добиться, просто подставив новые значения свойств `Height` и `Width` в окне **Properties**.

С другой стороны, в режиме выполнения программист имеет возможность не только манипулировать всеми свойствами, отображаемыми в окне **Properties**, но и управлять более обширным их списком. В режиме выполнения невозможно использовать окно **Properties**. Однако, доступ к свойствам компонентов довольно легко получить программным путем. Для изменения какого-либо свойства Вы должны написать строчку кода аналогично следующей:

```
MyComponent->Width = 35;
```

Например, если в форме имеется кнопка `button1`, то можно изменить ее свойства следующим образом:

```
button1->Text = "Нажми на меня";
button1->Left = 200;
```

При этом новое значение для свойства `Text` указывается в кавычках, так как оно имеет строковый тип, а для свойства `Left` – без кавычек, так как оно имеет числовой тип.

События и обработки событий. С компонентами связаны *события*, которые происходят в момент воздействия на компонент чего-либо. *Событие* – действие, вызывающее реакцию объекта, например, нажатие клавиши на клавиатуре или кнопки мыши. С помощью свойств, связанных с событиями, можно указать, что при возникновении события следует выполнить определенную функцию обработки событий. *Обработчик события* представляет собой фрагмент программного кода, который выполняется в ответ на некоторое событие. Создав специальную функцию обработки этого события, можно запрограммировать некоторую реакцию на это действие.

Заготовка обработчика события формируется следующим образом: надо открыть вкладку **Events** (события) в окне **Properties**, выбрать нужное событие (например, `Click`) и дважды щелкнуть мышкой в поле этого события. Среда программирования сразу создает в h-файле формы заготовку: заголовок функции, которая должна будет выполняться, как только наступить соответствующее событие, и пустое тело функции (открывающую и закрывающую фигурные скобки), в которое мы должны вписать свои инструкции на C++, отражающие алгоритм реакции на событие.

Компоненты, размещаемые в форме, позволяют пользователю программы выполнять различные действия, например, щелкать на

кнопках или вводить данные. Рассмотрим наиболее часто используемые элементы управления, размещенные на панели Toolbox.

Button – прямоугольная кнопка с надписью. Поместив Button на форму, Вы по двойному щелчку можете создать заготовку обработчика события нажатия кнопки. Далее нужно заполнить заготовку кодом. Вот основные свойства элемента управления Button: *Left* (позиция элемента управления относительно левого края его контейнера), *Top* (позиция элемента относительно верхнего края его контейнера), *Height* (высота), *Width* (ширина), *Enabled* (определяет, можно ли пользователю работать с этим элементом управления), *Visible* (видимость во время выполнения программы), *Text* (подпись). Все указанные свойства можно менять как во время разработки программы, так и во время ее работы. Чтобы изменить свойства кнопки во время работы программы, необходимо в процедуре использовать команду вида:

```
НазваниеЭлемента->НазваниеСвойства = НовоеЗначение;
```

Например, если в форме имеется кнопка cmd1, то можно изменить ее свойства следующим образом:

```
cmd1->Text = "Нажми на меня";
cmd1->Left = 200;
```

При этом новое значение для свойства Text указывается в кавычках, так как оно имеет тип String, а для свойства Left – без кавычек, так как оно имеет числовой тип.

Label служит для отображения текста на экране. Вы можете изменить шрифт и цвет надписи, если щелкнете на кнопке с троеточием в строке Font в окне Properties. Текст надписи является значением свойства Text. Свойство TextAlign определяет способ выравнивания текста. Вывести текст в надпись во время работы программы можно, например, так:

```
Label1->Text = "Введите число:";
```

TextBox - стандартный управляющий элемент Windows для ввода. Он может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы. Начальное содержимое области редактирования определяет строка, являющаяся значением свойства Text.

Свойство Font определяет параметры шрифта текстового поля. Установив свойство ReadOnly в значение True, мы запрещаем пользователю программы вводить данные в текстовое поле. Свойство MultiLine позволяет установить возможность вывода текста в несколько строк. Свойство TextAlign определяет способ выравнивания текста.

Элементы управления **CheckBox** и **RadioButton** часто используются при программировании. *CheckBox* – это флажок для включения или отключения опций, а *RadioButton* – переключатель, который позволяет выбирать один элемент из группы.

Основные свойства элементов управления CheckBox и RadioButton: *Name* – имя элемента; *Text* – подпись; *Checked* – признак того, что элемент выбран; *Left*, *Top* – определяет положение элемента; *Height*, *Width* – задают размеры элемента. Свойство *Checked* для компонентов CheckBox и RadioButton равно True, если переключатель выбран и False, если элемент не выбран.

Для добавления в формы проекта различных компонентов используется панель **ToolBox**. Доступ к ней можно получить с помощью меню **View | Toolbox**.

Работа со строками в приложениях Windows Forms

Класс String позволяет работать со Unicode-строками в Windows Forms-приложениях. Создание новой пустой строки str производится следующим образом:

```
String ^str;
```

Затем переменную str можно использовать для хранения строк, например:

```
str = textBox1->Text;
```

Создание и использование новой непустой строки может быть выполнено так:

```
String ^str = "Hello All!";
textBox2->Text=str;
```

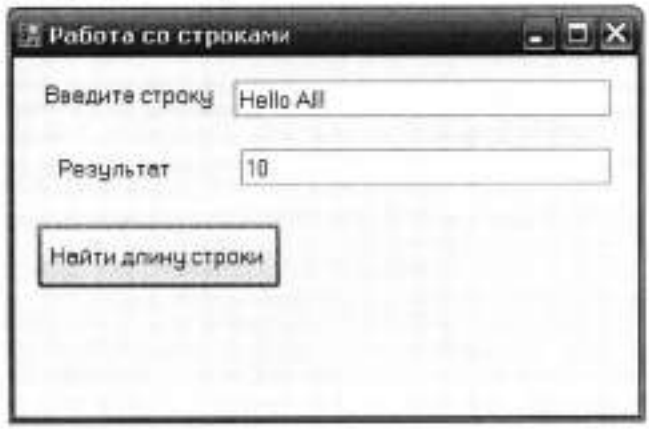
Класс String содержит свойства и методы, позволяющие работать со строками: определять длину строки, обращаться к отдельным символам, копировать, сравнивать строки и т.д. Все эти свойства и методы описаны в справочной системе в разделе **String Members.**

Рассмотрим некоторые возможности класса String.

Определение длины строки (Пример 3). Свойство Length объекта класса String содержит количество символов в строке. Пример. Разместим в форме 2 текстовых поля и кнопку «Определить длину строки». Текст функции обработки нажатия кнопки:

```
private: System::Void button1_Click(System::Object^  sender,
                System::EventArgs^  e)
{String ^str ;
 int k;
 str = textBox1->Text;
 k = str ->Length;
 textBox2->Text =
       Convert::ToString(k);
 }
```

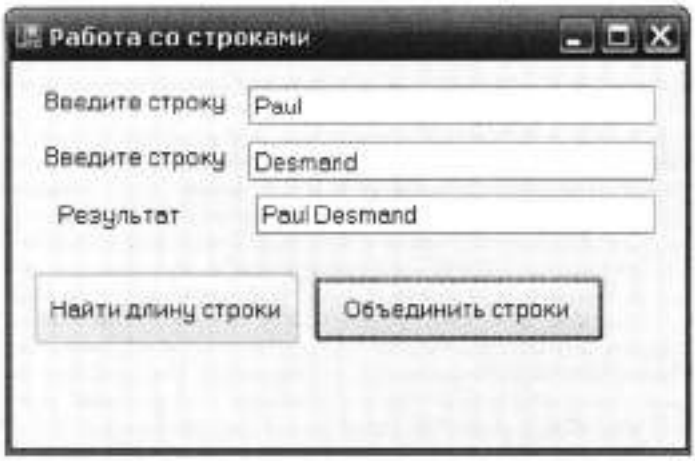
Результат работы программы представлен на рисунке.



Объединение строк может быть выполнено с помощью операции «+».

(Пример 4):

```
private: System::Void button2_Click(System::Object^  sender,
         System::EventArgs^  e)
```

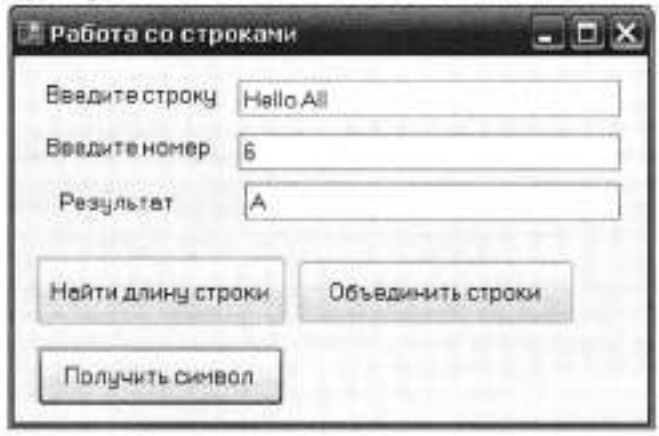



```
{String ^str1, ^str2 ;
 str1 = textBox1->Text;
 str2 = textBox3->Text;
 textBox2->Text = str1 +
" " +str2;
 }
```

Получение подстроки. Метод

`String^ Substring (int startIndex, int length)`

позволяет получить из строки подстроку. При этом *startIndex* – начальная позиция, с которой брать подстроку, а *length* -- длина подстроки. Например, этот метод можно использовать для получения одного символа из строки **(Пример 5)**:




```
private: System::Void
      button3_Click(System::
      Object^  sender,
      System::EventArgs^  e)
{String ^str1, ^str2 ;  int
      k;
 str1 = textBox1->Text;
 k = Convert::ToInt32 (textBox3->Text);
 str2=str1->Substring(k,1);
 textBox2->Text = str2;
 }
```

Порядок выполнения работы

1. Ознакомиться с теоретическими положениями, приведенными в данных методических указаниях (МУ), а также с конспектом лекций по данной теме.

2. Проверить работу примеров (1-3), приведенных в тексте МУ. Показать результаты преподавателю.

3. Разработать 2 программы по своему варианту. Варианты заданий см. в таблице. *Одну из программ реализовать как Windows Forms Application.*

Требования к отчету по работе

Отчет по работе должен содержать: название и цель работы; фамилию, инициалы и номер группы студента, выполнившего ЛР; номер варианта; для каждого из заданий – текст задачи по своему варианту, результаты работы программы для разных наборов исходных данных.

Вар.	Задание
1	1. Дан файл, содержащий текст на русском языке и некоторые два слова. Определить, сколько раз они встречаются в тексте и сколько из них — непосредственно друг за другом. 2. В текстовом поле записана строка. Подсчитать количество букв «я» в последнем ее слове.
2	1. Дан файл, содержащий текст, набранный заглавными русскими буквами. Провести частотный анализ текста, т. е. указать (в процентах), сколько раз встречается та или иная буква. 2. В текстовом поле дана строка, заканчивающаяся точкой. Подсчитать, сколько слов в строке.
3	1. Подготовить текстовый файл Stroka1.txt со строкой на русском языке. Разработать программу, удаляющую все буквы «а» из строки в файле и записывающую строку-результат в новый файл Stroka2.txt. 2. В текстовом поле записана строка, содержащая текст, заканчивающийся точкой. Вывести на экран слова, содержащие три буквы.
4	1. Дан файл, содержащий произвольный текст и некоторая буква. Определить, сколько слов в тексте начинается на указанную букву. Результат записать в новый файл. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и пробелом). 2. В текстовом поле записана исходная строка. Преобразовать ее, удалив каждый символ * и повторив каждый символ, отличный от *.
5	1. Даны файл, содержащий текст на русском языке, и некоторые буквы. Найти слово, содержащее наибольшее количество указанных букв. 2. В текстовом поле записана строка, содержащая буквы латинского алфавита и цифры. Вывести на экран длину наибольшей последовательности цифр, идущих подряд.
6	1. Дан файл, содержащий текст на русском языке. Составить в алфавитном порядке список всех слов, встречающихся в этом тексте. 2. В текстовом поле записана строка. Найти в нем те слова, которые начинаются и оканчиваются одной и той же буквой.
7	1. Дан файл, содержащий произвольный текст. Ввести с клавиатуры слово. Определить, содержится ли заданное слово в тексте. Результат записать в новый файл. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и пробелом). 2. В текстовом поле записана строка, содержащая текст, заканчивающийся точкой. Вывести в другое текстовое поле слова, содержащие ровно четыре буквы.
8	1. Дан файл, содержащий произвольный текст. Определить, сколько слов в тексте. Результат записать в новый файл. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и

	пробелом). 2. В текстовом поле записана строка. Найти длину самого короткого слова и самого длинного слова.
9	1. Дан файл, содержащий произвольный текст. Определить, сколько раз встречается в нем самое длинное слово. Результат записать в новый файл. 2. В текстовом поле записана строка. Определить, сколько слов в строке. Результат записать в новый файл. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и пробелом).
10	1. Дан файл, содержащий текст на русском языке. В предложениях некоторые из слов записаны подряд несколько раз (предложение заканчивается точкой или восклицательным знаком). Получить в новом файле отредактированный текст, в котором удалены повторные вхождения слов в предложениях. 2. В текстовом поле записана строка, содержащая произвольный текст. Создать новый файл и переписать в него текст из исходного файла, заменив каждую пару подряд идущих символов "а" (т.е. "аа") на "А". Остальные символы и их порядок не должны измениться.
11	1. В файле записан текст. Подсчитать, сколько слов в строке. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и пробелом). 2. В текстовом поле записана строка. Найти в нем те слова, которые начинаются и оканчиваются одной и той же буквой.
12	1. В файле записана строка символов, среди которых есть знак вопроса (?). Определить, сколько символов ему предшествует. 2. В текстовом поле слова зашифрованы — каждое из них записано наоборот. Расшифровать сообщение.
13	1. Даны файл, содержащий текст на русском языке, и некоторые буквы. Найти слово, содержащее наибольшее количество указанных букв. 2. Даны два текстовых поля *A* и *B*. Составьте программу, проверяющую, можно ли из букв, входящих в *A*, составить *B* (буквы можно использовать не более одного раза и можно переставлять). Например, *A:* ИНТЕГРАЛ; *B:* АТЕЛЬЕ — составить нельзя; *B:* ГРАНТ — составить можно.
14	1. В файле записана строка, содержащая текст. Найти длину самого короткого слова и самого длинного слова. 2. В текстовом поле записана строка. Подсчитать количество букв «я» в последнем ее слове.
15	1. Дан файл, содержащий текст на русском языке и некоторые два слова. Определить, сколько раз они встречаются в тексте и сколько из них — непосредственно друг за другом. 2. В текстовом поле записана строка. Определить, чего в ней содержится больше: русских букв или цифр.
16	1. Дан файл, содержащий текст, набранный заглавными русскими буквами. Провести частотный анализ текста, т. е. указать (в процентах), сколько раз встречается та или иная буква. 2. В текстовом поле записана строка, содержащая текст, заканчивающийся точкой. Вывести на экран слова, содержащие три

	буквы.
17	1. Подготовить текстовый файл Stroka1.txt со строкой на русском языке. Разработать программу, удаляющую все буквы «а» из строки в файле и записывающую строку-результат в новый файл Stroka2.txt. 2. В текстовом поле дана строка, заканчивающаяся точкой. Подсчитать, сколько слов в строке.
18	1. Дан файл, содержащий произвольный текст и некоторая буква. Определить, сколько слов в тексте начинается на указанную букву. Результат записать в новый файл. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и пробелом). 2. В текстовом поле записана строка. Найти в нем те слова, которые начинаются и оканчиваются одной и той же буквой.
19	1. Даны файл, содержащий текст на русском языке, и некоторые буквы. Найти слово, содержащее наибольшее количество указанных букв. 2. В текстовом поле записана исходная строка. Преобразовать ее, удалив каждый символ * и повторив каждый символ, отличный от *.
20	1. Дан файл, содержащий текст на русском языке. Составить в алфавитном порядке список всех слов, встречающихся в этом тексте. 2. В текстовом поле записана строка, содержащая буквы латинского алфавита и цифры. Вывести на экран длину наибольшей последовательности цифр, идущих подряд.
21	1. Дан файл, содержащий произвольный текст. Ввести с клавиатуры слово. Определить, содержится ли заданное слово в тексте. Результат записать в новый файл. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и пробелом). 2. В текстовом поле записана строка. Найти длину самого короткого слова и самого длинного слова.
22	1. Дан файл, содержащий произвольный текст. Определить, сколько слов в тексте. Результат записать в новый файл. (Слова могут разделяться только одним пробелом, точкой и пробелом, запятой и пробелом). 2. В текстовом поле записана строка, содержащая текст, заканчивающийся точкой. Вывести в другое текстовое поле слова, содержащие ровно четыре буквы.

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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к практическим и самостоятельным работам

по дисциплине «Математическое моделирование»

специальности 09.02.03 Программирование в компьютерных системах

(углубленная подготовка)

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий информационных технологий
Протокол от «15» января 2020 г. № 6
Председатель цикловой комиссии И.В.Миляева

Авторы: Воронцова Н.В., канд. техн. наук

Тема 2.2 Линейное программирование

Занятие 1. Решение задачи оптимального планирования

1.Теоретическая часть

Симплексный метод решения задачи линейного программирования основан на переходе от одного опорного плана к другому, при котором значение левой функции возрастает (при условии, что данная задача имеет оптимальный план и каждый ее опорный план является невырожденным). Указанный переход возможен, если известен какой-нибудь исходный опорный план. Рассмотрим задачу, для которой этот план можно непосредственно записать.

Пусть требуется найти максимальное значение функции

$$F=c_1x_1+c_2x_2+\dots c_nx_n$$

При условиях

$$\begin{cases} x_1 + a_{1m+1}x_{m+1} + \dots + a_{1n}x_n = b_1, \\ x_2 + a_{2m+1}x_{m+1} + \dots + a_{2n}x_n = b_2, \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ x_m + a_{mm+1}x_{m+1} + \dots + a_{mn}x_n = b_m, \\ x_j \geq 0 \left(j = \overline{1,n}\right) \end{cases}$$

Здесь a_{ij}, bi и, C_j (i= 1,m;;.j .=. 1,n) заданное постоянные числа $(m < n;\ b_i>0)$

Векторная форма данной задачи имеет следующий вид:

найти максимум функции $F=\sum_{j=1}^{n} c_j x_j \rightarrow \max$

при условиях

$$x_1A_1 + x_2A_2 + \dots + x_mA_m + \dots + x_nA_n = B,$$
$$x_j \geq 0 \ \left(j = \overline{1,n}\right)$$

где

$$A_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}; \quad A_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}; \dots; \quad A_m = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}; \quad A_{m+1} = \begin{pmatrix} a_{1m+1} \\ a_{2m+1} \\ \vdots \\ a_{mm+1} \end{pmatrix}; \dots;$$

$$A_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix}; \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

(13)- (15)

План $X = (x_1,x_2,...x_n)$ называется **опорным планом задачи** линейного программирования, если система векторов $A_1.A_2.A_n$. входящих в разложение (14) с положительными величинами Xj >0 (i = 1,n), линейно независима. Опорный план называется **невырожденным,** если он содержит ровно ***m*** положительных компонент, иначе -вырожденным.

Так как $b_1A_1+b_2A_2+...b_nA_n=B$, то вектор $X=(bj,b_2, ...b_n,,,0,... 0)$ является **опорным планом** данной задачи (последние *n-m* компонент вектора равны 0).

Этот план определяется системой единичных векторов $A_1,A_2,...,A_m$, которые образуют **базис m-мерного пространства. Поэтому** каждый из векторов $Ai,A_2,...,A_m$ а также вектор B могут быть представлены в виде линейной комбинации векторов данного базиса

Пусть $A_j = \sum_{i=1}^{m} x_{ij} A_i \ (j = \overline{1,n})$

Положим $z_j = \sum_{i=1}^{m} c_i x_{ij} \ (j = \overline{1,n}); \quad \Delta_j = z_j - c_j \ (j = \overline{1,n})$

Так как векторы $A_1, A_2, ..., A_m$ - единичные, то $x_{ij} = a_{ij}$ и $z_j = \sum_{i=1}^{m} c_i a_{ij}$, а

$$\Delta_j = \sum_{i=1}^{m} c_i a_{ij} - c_j.$$

Опорный план X =(x_1,x 2 , - - - x $_n$;0,0,...0) задачи (13)-(15)

Если $\Delta_к$ <0 для некоторого.i=k и среди чисел a_{ik}(i= l,m) нет положительных $(a,_к$ <0), то целевая функция (13) задачи (13)- (15) не ограничена на множестве ее планов.

Если опорный план X задачи (13) - (15) не вырожден и $\Delta_к$ <0, но среди чисел a_{lk} есть положительные (не все $a_{lk}<0$), то существует опорный- план X такой, что F(X)>F(X).

Сформулированные положения позволяют проверить, является ли найденный опорный план оптимальным, и выявить целесообразность перехода к новому опорному плану.

Исследование опорного плана на оптимальность, а также дальнейший вычислительный процесс удобнее вести,, если условия задачи и первоначальные данные, полученные после определения исходного опорного плана, записать

i	Базис	$c_б$	B	C_1	C_2	...	c_r	...	c_m	C_{m+1}	...	c_k	c_n
				X_1	X_2	...	X_r	...	X_m	X_{m+1}	...	X_k	X_n
1	X_1	C_1	B_1	1	0		0		0	A_{1m+1}		A_{1k}	A_{1n}
2	X_2	C_2	B_2	0	1		0		0	A_{2m+2}		A_{2k}	A_{2n}
...	...			0	0								
r	x_r	C_r	b_r	0	0		1		0	A_{rm+1}		a_{rk}	a_{rn}
...	...			0	0		0						
m	x_m	c_m	b_m	0	0		0		0	A_{mm+1}		A_{mk}	A_{mn}
M+1			F_0	0	0		0		1	Δ_{m+1}		$\Delta<0$	Δ_n

в виде табл.8.;:.

В столбце $C_б$ этой таблицы записывают коэффициенты при неизвестных (целевой функции, имеющие те же индексы, что и векторы данного базиса. В .столбце *B* записывают положительные компоненты исходного опорного плана, в нем же в результате вычислений получают положительные компоненты оптимального плана. Столбцы векторов *Aj* представляют собой коэффициенты разложения этих векторов по векторам данного базиса. В табл.8 первые *m* строк определяются исходными данными задачи, а показатели (т+1)-й строки вычисляют. В этой строке в столбце вектора В записывают значение целевой функции, которое она принимает при данном опорном плане, а в столбце вектора А,- значение $A_j = Zj - Cj$.

Значение величины z находится как скалярное произведение вектора

$C_0 = (c_1; c_2; \ldots; c_m)$ на вектор $A_j (j = \overline{1,m})$: $\quad z_j = \sum_{i=1}^{m} c_i a_{ij} \quad (j = \overline{1,n})$

Значение F_o равно произведению вектора $C_б$ на вектор В:

$$F_0 = \sum_{i=1}^{m} c_i a_{ij} \quad (j = 1, n)$$

После заполнения табл.8 исходный опорный план проверяют на оптимальность. Для этого просматривают элементы (m+1)-й строки таблицы. В результате может иметь место один из следующих трех случаев:

1. $\Delta j > 0$ для j =m+l, m+2, ..., *n* (при j =*l.m, Zj* = c_j. Поэтому в данном случае числа $\Delta j > 0$ для всех j от 1 до n;
2. $\Delta j < 0$ для некоторого *j,* и все соответствующие этому индексу величины $a_{ij} < 0$ (i=l,m)
3. $\Delta j < 0$ для некоторых индексов j, и для каждого такого j по крайней мере одно из чисел a_{ij} положительно.

В первом случае исходный опорный план является оптимальным.

Во втором случае целевая функция не ограничена сверху на множестве планов.

В третьем случае можно перейти от исходного плана к новому опорному плану, при котором значение целевой функции увеличится.

Этот переход от одного опорного плана к другому осуществляется исключением из исходного базиса какого-нибудь из векторов и введением в него нового вектора. В качестве вектора, вводимого в базис, можно взять любой из векторов *A,* имеющий индекс j, для которого $\Delta j < 0$. Пусть, например, $\Delta_к < 0$ и решено ввести в базис вектор $A_к$.

Для определения вектора, подлежащего исключению из базиса, находят $\min(bj/a_k)$ для всех $a_{ik} > Q$. Пусть этот минимум достигается при i = r. Тогда из базиса исключают вектор A_r а число a_r называют разрешающим элементом. Столбец и строку, на пересечении которых находится разрешающий элемент, называют направляющими.

После выделения направляющей строки и направляющего столбца находят новый опорный план и коэффициенты разложения векторов A_i,

через векторы нового базиса, соответствующего новому опорному плану. Это легко реализовать, если воспользоваться методом Жордана- Гаусса.

При этом можно показать, что положительные компоненты нового опорного плана вычисляются по формулам,

$$b_i = \begin{cases} b_i - (b_r / a_{rk}) a_{ik} \text{ при } i \neq r \\ b_r / a_{rk} \text{ при } i = r \end{cases}$$

а коэффициенты разложения векторов а, пересчитываются через векторы нового базиса, соответствующего новому опорному плану

$$a'_i = \begin{cases} a_i - (a_{ik} / a_{rk}) a_{rk} \text{ при } i \neq r \\ a_{ri} / a_{rk} \text{ при } i = r \end{cases}$$

После вычисления Ь, и a_{1j} согласно формулам (16) и (17) их значения заносят в табл.9. Элементы (т+1)-й строки этой таблицы могут .быть вычислены либо по формулам

$$F'_{ri} = F_{ri} - (b_r / a_{rk}) \Delta_i$$

$$\Delta'_i = \Delta_i - (a_{ri} / a_{rk}) \Delta_i$$

либо на основании их определения.

Наличие двух способов нахождения элементов (n+1)-й строки позволяет осуществлять контроль правильности проводимых вычислений.

Из формулы (18) следует, что при переходе от одного опорного плана к другому наиболее целесообразно ввести в базис вектор *Aj* (переменную $x_{,j}$ имеющий индекс j, при котором максимальным по абсолютной величине является число (b_r, a_{rj}) $\Delta j<0$, $a_n >0$). Однако с целью упрощения вычислительного процесса в дальнейшем будем вектор, вводимый в базис, определять исходя из максимальной абсолютной величины отрицательных чисел Δ_{ij}. Если же таких чисел несколько, то в базис будем вводить вектор, имеющий такой; индекс, как и максимальное из чисел C_{ij}, определяемых данными числами Δ_j.

Итак, переход от одного опорного плана к другому сводится к переходу от одной симплекс-таблицы к другой. Элементы новой симплекс-таблицы можно вычислить как с помощью рекуррентных формул (16) - (19), так и по правилам, непосредственно вытекающим из них. Эти правила состоят в следующем.

В столбцах векторов, входящих в базис, на пересечении строк и столбцов одноименных векторов проставляются единицы, а все остальные элементы данных столбцов полагают равными нулю. Элементы векторов A_i и A_j в строке новой симплекс-таблицы, в которой записан вектор, вводимый в базис, получают из элементов этой же строки исходной таблицы делением их на величину разрешающего элемента. В столбце C_6 в строке вводимого вектора проставляют величину c_r, где *к* - индекс вводимого вектора.

Остальные элементы столбцов вектора A_i и A_j новой симплекс-таблицы вычисляют по правилу треугольника. Для вычисления какого-нибудь из элементов находят три числа:

1)число, стоящее в исходной симплекс-таблице на месте искомого элемента новой симплекс-таблицы;

2)число, стоящее в исходной симплекс-таблице на пересечении строки, в которой находится искомый элемент новой симплекс-таблицы, и столбца, соответствующего вектору, вводимому в базис;

3)число, стоящее в новой симплекс-таблице на пересечении столбца, в котором стоит Искомый элемент, и строки вновь вводимого в базис вектора (как отмечено выше, эта строка получается из строки исходной симплекс-таблицы делением ее элементов на разрешающий элемент).

Эти три .числа, образуют своеобразный треугольник, две вершины которого соответствуют числам, находящимся в исходной симплекс-таблице, *а* третья - числу, находящемуся в новой симплекс-таблице. Для определения искомого элемента новой симплекс-таблицы из первого числа вычитают произведение второго и третьего. После заполнения новой симплекс-таблицы просматривают элементы (m+1)-й строки. Если все $z'j — Cj > 0$, то новый опорный план является оптимальным. Если же среди указанных чисел имеются отрицательные, то, используя описанную выше последовательность действий, находят новый опорный план. Этот процесс продолжают до тех пор, пока либо не получают оптимальный план задачи, либо устанавливают ее неразрешимость.

Примечание. Если задача имеет вырожденные опорные планы, то на одной из итераций одна или несколько переменных опорного плана могут оказаться равными нулю. Таким образом, при переходе от одного опорного плана к другому значение функции может остаться прежним. Более того, возможен случай, когда функция сохраняет свое значение в течение нескольких итераций, а также возможен возврат к первоначальному базису. В последнем случае обычно говорят, что произошло зацикливание. Однако при решении практических задач этот случай встречается очень редко и здесь могут быть применены специальные меры

Итак, нахождение оптимального плана симплексным методом включает следующие **этапы:**

1.Находят опорный план.

2.Составляют симплекс-таблицу.

3.Выясняют, имеется ли хотя бы одно отрицательное число А, . Если нет, то найденный опорный план оптимален. Если же среди чисел имеются отрицательные, то либо устанавливают неразрешимость задачи, либо переходят к новому опорному плану.

4. Находят направляющие столбец и строку. Направляющий столбец определяется наибольшим по абсолютной величине отрицательным числом Δ_j а направляющая строка - минимальным из отношений компонент столбца вектора. В к положительным компонентам направляющего столбца.

5.По формулам (16) - (19) определяют положительные компоненты нового опорного плана, коэффициенты разложения векторов $A>$ по векторам нового базиса и числа Fo', Δj. Все эти числа записываются в новой симплекс-таблице.

6. Проверяют найденный опорный план на оптимальность. Если план не оптимален и необходимо перейти к новому опорному плану, то возвращаются к этапу 4, а в случае получения оптимального плана или установления неразрешимости процесс решения задачи заканчивают.

2.Практическая часть

Пример. Для изготовления различных изделий А, В и С предприятие использует три различных вида сырья. Нормы расхода сырья на производство одного изделия каждого вида, цена одного изделия А, В, С а также общее количество сырья каждого вида, которое может быть использовано предприятием, приведены в табл.10.

Исходные данные задачи

Вид сырья	Нормы затрат сырья (кг) на одно изделие			Общее количество сырья (кг)
	A	B	C	
I	18	15	12	360
II	6	4	8	192
III	5	3	3	180
Цена одного изделия (руб)	9	10	16	

Изделия А, В и С- могут производиться в любых соотношениях (сбыт обеспечен), но производство ограничено выделенным предприятию сырьем каждого вида. Составить план производства изделий, при котором общая стоимость всей произведенной предприятием продукции является максимальной.

Решение. Составим математическую модель задачи. Искомый выпуск изделий А обозначим через x_h изделий В - через x_2, изделий С -через x_3. Поскольку имеются ограничения на выделенный предприятию фонд сырья каждого вида, переменные $x_h x_:, x_3$ должны удовлетворять следующей системе неравенств:

$$\begin{cases} 18x_1 + 15x_2 + 12x_3 \le 360, \\ 6x_1 + 4x_2 + 8x_3 \le 192, \\ 5x_1 + 3x_2 + 3x_3 \le 180. \end{cases}$$

Общая стоимость произведенной предприятием продукции составляет критерий

$$F=9x_i + 10x_2+16x_3 \quad -> \quad max. \qquad (21)$$

По своему экономическому содержанию переменные могут принимать только лишь неотрицательные значения:

$$X_ьx_2,x_3>0. \qquad (22)$$

Приходим к следующей математической задаче: среди всех неотрицательных решений системы неравенств (20) требуется найти такое, при котором функция (21) принимает максимальное значение.

Запишем эту задачу в форме основной задачи линейного программирования. Для этого перейдем от ограничений-неравенств к ограничениям-равенствам. Введем три дополнительные переменные, в результате чего ограничения запишутся в виде системы уравнении

$$\begin{cases} 18x_1 + 15x_2 + 12x_3 + x_4 = 360, \\ 6x_1 + 4x_2 + 8x_3 + x_5 = 192, \\ 5x_1 + 3x_2 + 3x_3 + x_6 = 180. \end{cases}$$

Эти дополнительные переменные по экономическому смыслу означают не используемое при данном плане производства количество сырья того или иного вида. Например, x_4 - это не используемое количество сырья I вида.

Преобразованную систему уравнений запишем в векторной форме:

$$x_1A_1+x_2A_2+x_3A_3+x_4A_4+x_5A_5+x_6A_6=B$$

$$A_1=\begin{pmatrix}18\\6\\5\end{pmatrix};\ A_2=\begin{pmatrix}15\\4\\3\end{pmatrix};\ A_3=\begin{pmatrix}12\\8\\3\end{pmatrix};\ A_4=\begin{pmatrix}1\\0\\0\end{pmatrix};\ A_5=\begin{pmatrix}0\\1\\0\end{pmatrix};\ A_6=\begin{pmatrix}0\\0\\1\end{pmatrix};\ B=\begin{pmatrix}360\\192\\180\end{pmatrix}$$

Для данной опорной задачи можно непосредственно записать опорный план X=(0; 0; 0; 360; 192; 180), определяемый системой трехмерных единичных векторов *A_4, A_5, A_6*, которые образуют базис трехмерного векторного пространства.

Составляем симплексную таблицу для I итерации (табл.11), подсчитываем значения F_o, Aj =*Zj-Cj* и проверяем исходный опорный план на оптимальность-

$F_0 = (C, B) = 0; \quad z_1 = (C, A_1) = 0; \quad z_2 = (C, A_2) = 0; \quad z_3 = (C, A_3) = 0;$

$\Delta_1 = z_1 - c_1 = 0 - 9 = -9; \quad \Delta_2 = z_2 - c_2 = 0 - 10 = -10; \quad \Delta_3 = z_3 - c_3 = -16.$

Для векторов базиса $\Delta_j = z_j - c_j = 0$.

Первая итерация решения задачи симплекс-методом

i	Базис	$C_б$	B	9	10	16	0	0	0
				x_1	x_2	x_3	x_4	x_5	x_6
1	x_4	0	360	18	15	12	1	0	0
2	x_5	0	192	6	4	8	0	1	0
3	x_6	0	180	5	3	3	0	0	1
4			$F_0=0$	-9	-10	-16	0	0	0

Из табл.11 видно, что значения всех основных переменных X_1, x_2, x_3 равны нулю, а дополнительные переменные принимают свои значения в соответствии с ограничениями задачи. Эти значения переменных отвечают такому плану, при котором ничего не производится, сырье не используется и значение целевой функции равно нулю (т.е. стоимость произведенной продукции отсутствует).

Этот план не является оптимальным. Это видно и из 4-й строки табл..] 1 так как в ней имеется три отрицательных числа: Zj — c_j =-9, $z_2 - c_2$,= -10 и Z_3-c_3=-16

Отрицательные числа не только свидетельствуют о возможности увеличения общей стоимости производимой продукции, но и показывают, насколько увеличится эта сумма при введении в план единицы того или другого вида продукции.

Так, число -9 означает, что при включении в план производства одного 1 изделия А обеспечивается увеличение выпуска продукции на 9 руб. Если включить в план производства по одному изделию В и С, то общая стоимость изготовляемой продукции возрастет соответственно на 10 и 16 руб. Поэтому с экономической точки зрения наиболее целесообразным является включение в план производства изделий С.

Это *же* необходимо сделать и на основании формального признака симплексного метода, поскольку максимальное по абсолютной величине отрицательное. число Δ_j, стоит в 4-й строке столбца вектора A_3. Следовательно, в базис введем вектор A_3.

Определяем вектор, подлежащий исключению из базиса. Для этого находим $\theta_o=min(b/_{13})$ для $a_{13}>0$, т.е. θ_0=min(360/12; 192/8; 180/3) = 192/8,; Найдя число 192/8=24, мы тем самым с экономической точки зрения определили, какое количество изделий С предприятие может изготовлять с учетом норм расхода и имеющихся объемов сырья каждого вида.

Так как сырья каждого вида соответственно имеется 360, 192 и 180 кг, а на одно изделие С требуется затратить сырья каждого вида соответственно 12, 8 и 3 кг, то максимальное число изделий С, которое может быть изготовлено предприятием, равно min(360/12; 192/8; 180/3) = 192/8 = 24, т.е. ограничивающим фактором для производства изделий С является имеющийся объем сырья 2 вида. С учетом его наличия предприятие может изготовить 24 изделия С. При. этом сырье 2 вида будет полностью использовано.

Следовательно, вектор A_5 подлежит исключению из базиса. Столбец вектора A_3 и 2-я строка являются направляющими. Составляем таблицу для II; итерации (табл.12).

Вторая итерация решения задачи симплекс-методом

i	Базис	C_6	B	9	10	16	0	0	0
				x_1	x_2	x_3	x_4	x_5	x_6
1	x_4	0	72	9	9	0	1	-3/2	0
2	x_3	16	24	3/4	1/2	1	0	1/8	0
3	x_6	0	108	11/4	3/2	0	0	-3/8	1
4			384	3	-2	0	0	2	0

Сначала заполняем строку вектора, вновь введенного в базис, т.е. строку, номер которой совпадает с номером направляющей строки. Здесь направляющей является 2-я строка. Элементы этой строки табл.. 12 получаются из соответствующих элементов табл.. 11 делением их на разрешающий элемент (т.е на 8). При этом в столбце C_6 записываем коэффициент C_3 =16, стоящий в столбце вводимого в базис вектора A_3.

Затем заполняем элементы столбцов для векторов, входящих в новый базис. В этих столбцах на пересечении строк и столбцов одноименных векторов проставляем единицы, а все остальные элементы полагаем равными нулю.

Для определения остальных элементов табл.12 применяем правило треугольника. Эти элементы могут быть вычислены и непосредственно по рекуррентным формулам.

Вычислим элементы табл.12, стоящие в столбце вектора *B*. Для вычисления первого элемента столбца находим три числа:

1)число, стоящее в табл.11 на пересечении столбца вектора *B* и 1-й строки (360);

2)число, стоящее в табл.11, на пересечении столбца вектора A_3 и 1-й строки (12);

3)число, стоящее в табл.12 на пересечении столбца вектора *B* и 2-йстроки (24).

Вычитая из первого числа произведение двух других, находим искомый элемент: 360-12*24 = 72; записываем его в 1-й строке столбца, вектора *B* табл.12. (Этот же результат получается и по данным табл.11: 360-12-192/8).

Второй элемент столбца вектора *B* табл.6.12 был уже вычислен ранее. Для вычисления третьего элемента столбца вектора *B* также находим три числа. Первое из них (180) находится на пересечении 3-й строки и столбца вектора *B* табл.11, второе (3) - на пересечении 3-й строки и столбца вектора A_3 , табл.11, третье (24) - на пересечении 2-й строки и столбца вектора *B* табл.12. Итак, указанный элемент есть^80-24*3 = 108. Число 108 записываем в 3-й строке столбца вектора *B* табл.12.

Значение F_o *в* 4-й строке столбца этого же вектора можно найти двумя способами:

1)по формуле F_o = (C,B),т.е. F_o =0-72 + 16-24+0*108-384;

2) по правилу треугольника; в данном случае треугольник образован числами 0, -16,24. Этот способ приводит к тому же результату; 0- (-16) • 24 = 384.

При определении по правилу треугольника элементов столбца вектора *B* третье число, стоящее в нижней вершине треугольника, все время оставалось неизменным и менялись лишь первые два числа. Учтем это при нахождении элементов столбца вектора A_1 табл.12. Для вычисления указанных

элементов первые два числа берем из столбцов векторов A_1 и A_3 табл.11, а третье число из табл.12.

... Это число стоит на пересечении 2-й строки и столбца вектора A_1 последней таблицы. В результате получаем значения искомых элементов:

18-12-(3/4) = 9; 5-3-(3/4) =11/4

Число z_i-c_i в 4-й строке столбца вектора A_1 табл.12 можно найти двумя способами:

1) по формуле Zi — Ci = *(C,A)-Cj* имеем 0*9-(-16)*(3/4)+0*(11/4)-9=3.

2) по правилу треугольника получим: - 9-(-16)•(3/4)=3.

Аналогично находим элементы столбца вектора A_2

Элементы столбца вектора A_5 вычисляем по правилу треугольника. Элемент 1-й строки указанного столбца равен 0-12•(1/8)=-3/2. Элемент, стоящий в 3-й строке данного столбца, равен $0-3\bullet(1/8)$ =-3/8.

По окончании расчета всех элементов табл.12 в ней получены опорный план и коэффициенты разложения векторов A_j (j = 1,6) через базисные векторы A_4,A_3,A_6 и значения Δ_j и F_o.

Новым опорным планом задачи является план F = (0; 0; 24; *11;* 0; 108) При данном плане производства изготовляется 24 изделия С и остается ней пользованным 72 кг сырья I вида и 108 кг сырья Ш вида. Стоимость всей изводимой при этом плане продукции равна 384 руб.

Изменились данные столбцов, а их экономическое содержание стало более сложным. Так, например, число 1/2 во 2-й строке столбца вектора *A?* показывает, на сколько следует уменьшить изготовление изделий С, если запланировать выпуск одного изделия В.

Числа 9 и 3/2 в 1-й и 3-й строках вектора A_2 показывают соответственно, сколько потребуется сырья I и II вида при включении в план производства одного изделия В, а число -2 в 4-й строке показывает, что если будет

запланирован выпуск одного изделия В, то это обеспечит увеличение выпуска продукции в стоимостном выражении на 2 руб.

Иными словами, если включить в план производства продукции одно изделие В, то это потребует уменьшения выпуска изделия С на 1/2 ед. и потребует дополнительных затрат 9 кг сырья I вида и 3/2 кг сырья Ш вида, а общая стоимость изготовляемой продукции в соответствии с новым оптимальным планом возрастет на 2 руб.

Таким образом, числа 9 и 3/2 выступают как бы новыми «нормами» затрат сырья 1 и Ш вида на изготовление одного изделия В (как видно из табл.11, ранее они были равны 15 и 3), что объяеняется уменьшением выпуска изделий С. Такой же экономический смысл имеют данные столбца вектора *А* / табл. 12.

Число 1/8 во 2-й строке столбца *А5,* показывает, что увеличение объемов сырья П вида на 1 кг позволили бы увеличить выпуск изделий С на 1/8 ед. Одновременно потребовалось бы дополнительно 3/2 кг сырья I вида и 3/8 кг сырья Ш вида. Увеличение выпуска изделий С на 1/8 ед. приведет к росту выпуска продукции на 2 руб. Из изложенного выше экономического содержания данных табл.12 следует, что найденный на II итерации план задачи не является оптимальным. Это видно и из 4-й строки табл.12, поскольку в столбце вектора A_2 этой строки стоит отрицательное число -2. Значит, в базис следует ввести вектор A_2, т.е. в новом плане следует предусмотреть выпуск изделий В

При определении возможного числа изготовления изделий В следует учитывать имеющееся количество сырья каждого вида, а именно: возможный выпуск изделий В определяется min (b *{) /a',)* для *а* ,2 > 0, т. е. находим

$$\theta_0 = \min\left(\frac{72}{9}; \frac{24 \cdot 2}{1}; \frac{108 \cdot 2}{3}\right) = \frac{72}{9} = 8.$$

Следовательно, исключению из базиса подлежит вектор Ад, иными словами, выпуск изделий В ограничен имеющимся в распоряжении предприятия сырьем I вида. С учетом имеющихся объемов этого сырья

предприятию следует изготовить 8 изделий В. Число 9 является разрешающим элементом, а столбец вектора A_2 и 1-я строка табл.12 являются направляющими. Составляем таблицу для III итерации (табл.. 13).

В табл.6.13 сначала заполняем элементы 1-й строки, которая представляет собой строку вновь вводимого в базис вектора A_2. Элементы этой строки получаем из элементов 1-й строки табл.12 делением последних на разрешающий элемент (т.е. на 9). При этом в столбце C_6 данной строки записываем Сг —10. Затем заполняем элементы столбцов векторов оазиса и по правилу; треугольника вычисляем элементы остальных столбцов.

Таблица 13

i	Базис	$C_б$	B	9	10	16	0	0	0
				x_1	x_2	x_3	x_4	x_5	x_6
1	x_2	10	8	1	1	0	1/9	-1/6	0
2	x_3	16	20	1/4	0	1	-1/18	5/24	0
3	x_6	0	96	5/4	0	0	-1/6	-1/8	1
4			400	5	0	0	2/9	5/3	0

В результате получаем новый опорный план X=(0;8; *20;0;0;96)* и коэффициенты разложения векторов А, (j=l,6) через базисные векторы A_2,$Л_3$,A_6 и соответствующие значения Aj и F_o".

В 4-й строке среди чисел А, нет отрицательных. Это означает, что найденный опорный план является оптимальным и F_{max} = 400

Следовательно, план выпуска продукции, включающий изготовление 8 изделий В и 20 изделий С, является оптимальным.

При данном плане выпуска изделий полностью используется сырье 1 и 2 видов и остается неиспользованным 96 кг сырья Ш вида, а стоимость произведенной продукции равна 400 руб.

Оптимальным планом производства продукции не предусматривается изготовление изделий А. Введение в план выпуска продукции изделий вида А привело бы к уменьшению указанной обшей стоимости.

Это видно из 4-й строки столбца вектора ,4/, где число 5 показывает, что при данном плане включение в него выпуска .единицы изделия А приводит лишь к уменьшению общей величины стоимости на 5 руб.

Решение данного примера симплексным методом можно было бы проводить, используя лишь одну таблицу (табл.14). В этой таблице последовательно записаны одна за другой все три итерации вычислительного процесса

16

L_i	Базис	C_{ii}	B	9	10	16	0	0	0
				X_1	*2	*3	*4	*5	*6
Первая итерация									
1	*4	0	360	18	15	12	1	0	0
2	X_2	0	192	6	4	8	0	1	0
3	X_3	0	180	5	3	3	0	0	0
4			$F_0=0$	-9	-10	-16	0	0	0
Вторая итерация									
1	*4	0	72	9	9	0	1	-3/2	0
2	x_2	16	24	3/4	1/2	1	0	1/8	0
3	X_3	0	108	11/4	3/2	0	0	-3/8	1
4			384	3	-2	0	0	2	0
Третья итерация									
1	*2	10	8	1	1	0	1/9	-1/6	0
2	x_2	16	20	1/4	0	1	-1/18	5/24	0
3	X_3	0	96	5/4	0	0	-1/6	-1/8	1
4			400	5	0	0	2/9	5/3	0

3 Задания для самостоятельной работы

1. **<u>Задание 1</u>** Решить задачу линейного программирования симплекс-методом.

$$f = 2X_1 + X_2 - 2X_3 \to \min$$

$$\begin{cases} X_1 + X_2 - X_3 \geq 8; \\ X_1 - X_2 + 2X_3 \geq 2; \\ -2X_1 - 8X_2 + 3X_3 \geq 1; \\ X_i \geq 0 \, (i = 1, 2, 3). \end{cases}$$

2. **<u>Задание 2</u>**

Решить задачу линейного программирования симплекс-методом

$Y = 12X_1 + X_2 - 2X_3 \to \max$

$-2X_1 - X_2 \geq -10$

$2X_1 - X_2 - 2X_3 \leq 8$

$-X_1 + X_3 \leq -2$

3. **<u>Задание 3</u>**

Решить задачу линейного программирования симплекс-методом

$$z = 4x_1 + 6x_2 \to \max$$

$$\begin{cases} 2x_1 + x_2 \le 64 \\ x_1 + 3x_2 \le 72 \\ x_2 \le 20 \end{cases}$$

$$x_i \ge 0,\ i = 1,2.$$

4. **Задание 4**

Решить задачу линейного программирования симплекс-методом

$$x_2 - 3x_3 + 2x_5 \Rightarrow \min$$

$$\begin{aligned} x_1 \quad & +3x_2 \quad -x_3 \quad & & +2x_5 \quad & = 7 \\ & -2x_2 \quad +4x_3 \quad +x_4 \quad & & & = 12 \\ & -4x_2 \quad +3x_3 \quad & & +8x_5 \quad +x_6 & = 10 \end{aligned}$$

$$x_i \ge 0,\ \ i = \overline{1,6}$$

Занятие 2. Решение транспортной задачи

1. Теоретическая часть

Общая постановка транспортной задачи состоит в определении оптимального плана перевозок некоторого однородного груза из *m* пунктов отправления A_1, A_2, A_3 ...A_n пунктов в n пунктов назначения B_1, B_2 B_n. При этом в качестве критерия оптимальности обычно берется либо минимальная стоимость перевозок всего груза, либо минимальное время его доставки .

Рассмотрим транспортную задачу, в качестве критерия оптимально которой взята минимальная стоимость перевозок всего груза. Обозначим через c_{ij} тарифы перевозки единицы груза из i-го пункта отправления в j-й пункт назначения, через a_i - запасы груза в i-м пункте отправления, через b_j - потребности в грузе в j-м пункте назначения, а через x_{ij} - количество единиц груза,: ввозимого из i-го пункта отправления j-й пункт назначения.

Тогда математическая постановка задачи состоит в определении минимального значения функции

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

при условиях

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}),$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}),$$

$$x_{ij} \ge 0 \quad (i = \overline{1,m};\ j = \overline{1,n}). \qquad (1)\text{-}(4)$$

Поскольку переменные x_{ij} (i - 1,m; j = 1,n) удовлетворяют системам линейных уравнений (2) и (3) и условию неотрицательности (4), то обеспечиваются доставка необходимого количества груза в каждый из пунктов назначения, вывоз имеющегося груза из всех пунктов отправления, а также исключаются обратные перевозки.

Всякое неотрицательное решение систем линейных уравнений (2) и (3), определяемое матрицей $X=(x_{ij});(i=1,m;\ j=1n)$, называется планом транспортной задачи.

План $X^* = (x^*_{ij})$ (i = 1, m; $j=1,n$), при котором функция (1.5) принимает свое минимальное значение, называется оптимальным планом транспортной .задачи.

Обычно исходные данные задачи записывают в виде табл.1

Исходные данные транспортной задачи

Пункты отправления	Пункты назначения					Запасы
	B_1	...	B_j	...	B_n	
A_1	c_{11} x_{11}	...	c_{1j} x_{1j}	...	c_{1n} x_{1n}	a_1
...	...	...	...	...	...	...
A_i	c_{i1} x_{i1}	...	c_{ij} x_{ij}	...	c_{in} x_{in}	a_i
...	...	...	...	...	...	...
A_m	c_{m1} x_{m1}	...	c_{mj} x_{mj}	...	c_{mn} x_{mn}	a_m
Потребности	b_1	...	b_j	...	b_n	

Общее наличие груза у поставщиков равно $\sum_{ш=1}^{b} a_i$ а общая потребность в грузе в пунктах назначения равна $\sum_{j=1}^{b} b_j$ единиц. Если общая потребность в грузе в пунктах назначения равна запасу груза в пунктах, отправления, т.е.

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j, \quad (5)$$

то модель такой транспортной задачи называется закрытой. Если же указанное условие не выполняется, то модель транспортной задачи называется открытой. Для разрешимости транспортной задачи необходимо и достаточно, чтобы запасы груза в пунктах отправления были равны потребностям в грузе в пунктах назначения, т.е. чтобы выполнялось равенство (5).

В случае превышения запаса над потребностью ($\sum_{ш=1}^{b} a_i > \sum_{j=1}^{b} b_j$) вводится фиктивный (n+1) пункт назначения с потребностью $b_{n+t} = \sum_{ш=1}^{b} a_i - \sum_{j=1}^{b} b_j$ и соответствующие тарифы считаются равными нулю: $c_{ij} = 0$ (i= 1,m). Полученная задача является транспортной задачей, для которой выполняется равенство (5).

Аналогично, при ($\sum_{ш=1}^{b} a_i < \sum_{j=1}^{b} b_j$) вводится фиктивный (m+1)-й пункт отправления с запасом груза $a_{m+1} = \sum_{j=1}^{b} b_j - \sum_{ш=1}^{b} a_i$, и тарифы полагаются равными нулю: $c_{m+lj} = 0$ (j = 1,n).

Этим задача сводится к обычной транспортной задаче, из оптимального плана которой получается оптимальный план исходной задачи.

В дальнейшем будем рассматривать закрытую модель транспортной задачи. Если же модель конкретной задачи является открытой, то, исходя из сказанного выше, перепишем таблицу условий задачи так, чтобы выполнялось равенство (5).

Число переменных x_v в транспортной задаче *с m* пунктами отправления и *n* пунктами назначения равно *nm,* а число уравнений в системах (2) и (3)' равно n + m. Так как мы предполагаем, что выполняется условие (5), то число линейно независимых уравнений равно *n* + *m* - 1. Следовательно, опорный план транспортной задачи может иметь не более n+ *m* - 1 отличных от нуля неизвестных.

Если в опорном плане число отличных от нуля компонент равно в точности n + *m* -1, то план является невырожденным, а если меньше - то вырожденным.

Для определения опорного плана существует несколько методов: метод северо-западного угла, метод минимального элемента, метод аппроксимации Фогеля и др. Как и для всякой задачи линейного программирования, оптимальный план транспортной задачи является и опорным планом., Для определения оптимального плана транспортной задачи можно использовать стандартные методы линейного программирования, например, симплекс-метод. Однако ввиду исключительной практической важности этой задачи и специфики ее ограничений (каждая неизвестная входит лишь в два уравнения систем (2) и (3) и коэффициенты при неизвестных равны единице) для определения оптимального плана транспортной задачи разработаны специальные методы: метод потенциалов, метод дифференциальных рент и др.

Определение опорного плана транспортной задачи

Поиск оптимального плана транспортной задачи начинают с нахождения какого-нибудь ее опорного плана. Опорный план находят последовательно за n+m-1 шагов, на каждом из которых в таблице условий задачи заполняют одну клетку, которую называют занятой.

Заполнение одной из клеток обеспечивает полностью либо удовлетворение потребности в грузе одного из пунктов назначения (того, в столбце которого находится заполненная клетка), либо вывоз груза из одного из пунктов отправления (из того, в строке которого находится заполняемая клетка).

В первом случае временно исключают из рассмотрения столбец, содержащий заполненную на данном шаге клетку, и рассматривают задачу, таблица условий которой содержит на один столбец меньше, чем было перед этим шагом, но то же количество строк и соответственно измененные запасы груза в одном из пунктов отправления (в том, за счет запаса которого была удовлетворена потребность в грузе пункта назначения на данном шаге).

Во втором случае временно исключают из рассмотрения строку, содержащую заполненную клетку, и считают, что таблица условий имеет на одну строку меньше при неизменном количестве столбцов и при соответствующем изменении потребности в грузе в пункте назначения, в столбце которого находится заполняемая клетка.

После того, как проделаны *n+m-2* описанных выше шагов, получают задачу с одним пунктом отправления и одним пунктом назначения. При этом останется свободной только одна клетка, а запасы оставшегося пункта отправления будут равны потребностям оставшегося пункта назначения. Заполнив эту клетку, тем самым делают (n+m-1.)-й шаг и получают искомый опорный план.

Следует заметить, что на некотором шаге (уо не на последнем) может оказаться, что потребности очередного пункта назначения равны запасам очередного пункта отправления.

В этом случае также временно исключают из рассмотрения либо столбец, либо строку (что-нибудь одно). В итоге, либо запасы соответствующего пункта отправления, либо потребности данного пункта назначения считают равными нулю. Этот нуль записывают в очередную заполняемую клетку.

Указанные выше условия гарантируют получение n+m-l занятых клеток, в которых стоят компоненты опорного плана, что является исходным условием для проверки последнего на оптимальность и нахождения оптимального плана.

При нахождении опорного плана транспортной задачи методом северо-западного угла на каждом шаге рассматривают первый из оставшихся пунктов отправления и первый из оставшихся пунктов назначения. Заполнение клеток таблицы условий начинается с левой верхней клетки для неизвестного x_{11} («северо-западный угол») и заканчивается клеткой для неизвестного x_{mn}, т.е. процесс заполнения клеток идет как бы по диагонали таблицы.

2. Практическая часть

Пример. Для транспортной задачи, исходные данные которой приведены в табл..4, найти оптимальный план.

Решение. Сначала, используя метод северо-западного угла, находим опорный план задачи. Этот план записан в табл.4.

Найденный опорный план проверяем на оптимальность. В связи с этим находим потенциалы пунктов отправления и назначения.

Найденные числа записываем в каждую из свободных клеток табл. .5.

Таблица.5

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	1 30	2 20	4	1	50
A_2	2	3 10	1 10	5 10	30
A_3	3	2	4	4 10	10
Потребности	30	30	10	20	90

Для определения потенциалов получаем систему, содержащую шесть уравнений с семью неизвестными

$$\beta_1 - \alpha_1 = 1, \quad \beta_2 - \alpha_1 = 2, \quad \beta_2 - \alpha_2 = 3,$$
$$\beta_3 - \alpha_2 = 1, \quad \beta_4 - \alpha_2 = 5, \quad \beta_4 - \alpha_3 = 4.$$

Полагая $\alpha_1 = 0$, находим $\beta_1 = 1$, $\beta_2 = 2$, $\alpha_2 = -1$, $\beta_3 = 0$, $\beta_4 = 4$, $\alpha_3 = 0$.
Для каждой свободной клетки вычисляем число $\Delta_{ij} = \beta_j - \alpha_i - c_{ij}$:

$$\Delta_{13} = -4, \quad \Delta_{14} = 3, \quad \Delta_{21} = \Delta_{32} = 0, \quad \Delta_{31} = -2, \quad \Delta_{33} = -4.$$

Так как среди чисел Δ_{ij} имеются положительные, то построенный план перевозок не является оптимальным и надо перейти к новому опорному плану. Наибольшим среди положительных чисел Δ_{ij} являются $\Delta_{14} = 3$, поэтому для данной свободной клетки строим цикл пересчета (табл..5) и производим сдвиг по этому циклу. Наименьшее из чисел в минусовых клетках равно 10. Клетка, в которой находится это число, становится свободной в новой табл. .6.

Таблица .6

Промежуточные результаты решения транспортной задачи

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	1 30	2 (-) 20	4 (-4)	1 (+) (+3)	50
A_2	2 (0)	3 (+) 10	1 10	5 (-) 10	30
A_3	3 (-2)	2 (0)	4 (-4)	4 10	10
Потребности	30	30	10	20	90

Другие числа в табл..5 получаются следующим способом. К числу 10 в плюсовой клетке A_2B_2, добавим 10 и вычтем 10 из числа 20, находящегося в минусовой клетке AiB_2. Клетка на пересечении строки A_2 и столбца B_4 становится свободной.

После этих преобразований получаем новый опорный план (табл. :7).

Таблица .7

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	1 30	2 (-) 10	4 [-2]	1 (+) 10	50
A_2	2 (0)	3 20	1 10	5 [-3]	30
A_3	3 [+1]	2 (+) [+3]	4 [-1]	4 (-) 10	10
Потребности	30	30	10	20	90

Этот план проверяем на оптимальности. Снова находим потенциалы пунктов отправления и назначения. Для этого составляем следующую систему уравнений:

Полагаем $\alpha_1 = 0$, получаем $\beta_1 = \beta_4 = 1$, $\beta_2 = 2$, $\beta_3 = 0$, $\alpha_2 = -3$, $\alpha_3 = -1$. Для каждой свободной клетки вычисляем число Δ_{ij}.

Имеем: $\Delta_{13} = -2$, $\Delta_{21} = 0$, $\Delta_{24} = -3$, $\Delta_{31} = 1$, $\Delta_{32} = 3$, $\Delta_{33} = 3$.

Видим, что данный план перевозок не является оптимальным. Поэтому переходим к новому опорному плану (табл. .8)

Улучшенный опорный план решения транспортной задачи

Пункты отправления	Пункты назначений				Запасы
	B_1	B_2	B_3	B_4	
A_1	1 30	2 (0)	4 [-4]	1 20	50
A_2	2 (0)	3 20	1 10	5 [-3]	30
A_3	3 [-2]	2 10	4 [-4]	4 [-3]	10
Потребности	30	30	10	20	90

Сравнивая разности $\Delta_{ij}=\beta j$ — α_i новых потенциалов, отвечающих свободным клеткам табл.7, с соответствующими числами c_{ij} видим, что указанные разности потенциалов для всех свободных клеток не превосходят чисел c_{ij}.

Следовательно, получен оптимальный план:

$$X^* = \begin{pmatrix} 30 & 0 & 0 & 20 \\ 0 & 20 & 10 & 0 \\ 0 & 10 & 0 & 0 \end{pmatrix}.$$

При данном плане стоимость перевозок составляет величину

S= 1*30 + 2*0+ 1*20+ 3*20+1*10 + 2*10=140.

3 Задания для самостоятельной работы

1.

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	24	8	34	5	520
A_2	36	22	16	8	300
A_3	43	11	25	12	600
Потребности	200	320	400	170	

2.

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	7	4	17	5	120
A_2	4	2	12	8	280
A_3	3	8	18	2	160
Потребности	130	220	60	70	

3.

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	4	8	4	5	420
A_2	6	2	6	8	230
A_3	3	11	5	12	560
Потребности	230	320	160	170	

4.

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	8	57	35	27	200
A_2	12	62	34	32	280
A_3	11	68	33	11	320
Потребности	300	220	130	200	

Тема 2.4 Динамическое программирование
Практическое занятие 3.
Решение задач методом динамического программирования

1 Теоретическая часть

Уравнения Беллмана. Вместо исходной задачи ДП с фиксированным числом шагов и и начальным состоянием s_0 рассмотрим последовательность задач, полагая последовательно $n=1,2, \ldots$ при различных s— одношаговую, двухшаговую и т.д., — используя принцип оптимальности.

Введем ряд новых обозначений. Обозначения в ДП несут большую информационную нагрузку, поэтому очень важно их четко усвоить.

На каждом шаге любого состояния системы S_{k-1} решение $X_к$ нужно выбирать "с оглядкой", так как этот выбор влияет на последующее состояние s_k и дальнейший процесс управления, зависящий от s_k. Это следует из принципа оптимальности.

Но есть один шаг, последний, который можно для любого состояния s_{k-1} планировать локально-оптимально, исходя только из соображений этого шага. Рассмотрим n-й шаг: s_{n-1} — состояние системы к началу n-го шага, $s_n = s$ — конечное состояние, X_n — управление на n-м шаге, *а* $f_n(s_{n-1}, X_n)$ — целевая функция (выигрыш) n-го шага.

Согласно принципу оптимальности, X_n нужно выбирать так, чтобы для любых состояний s_{n-1} получить максимум целевой функции на этом шаге.

Обозначим через $Z^*_n(s_{n-1})$ максимум целевой функции — показателя эффективности n-го шага при условии, что к началу последнего шага система S была в произвольном состоянии s_{n-1}, а на последнем шаге управление было оптимальным.

$Z^*_n(s_{n-1})$ называется условным максимумом целевой функции на n-м шаге.

Очевидно, что

$$Z_n^*(s_{n-1}) = \max_{(X_n)} f_n(s_{n-1}, X_n). \qquad (5)$$

Максимизация ведется по всем допустимым управлениям X_n.

Решение X_n при котором достигается $Z^*_n(s_{n-1})$ также зависит от s_{n-1} и называется *условным оптимальным управлением* на *n-м шаге.* Оно обозначается через X^*_n- Решив одномерную задачу локальной оптимизации по уравнению

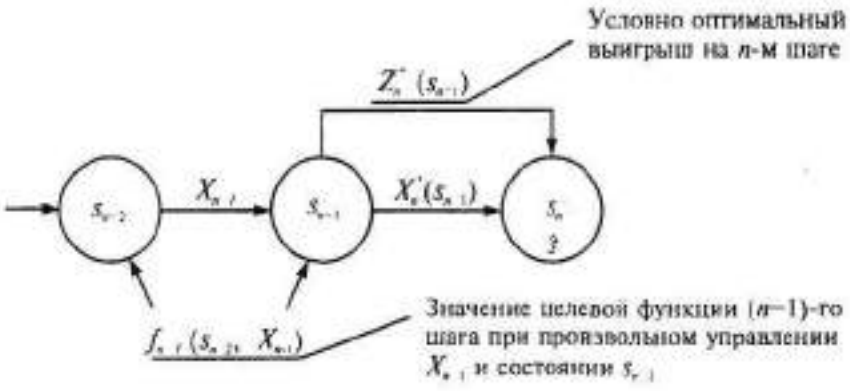


Рис. .2

(5), найдем для всех возможных состояний s_{n-1} две функции: $Z_N^*(s_{n-1})$ и $X_N^*(s_{n1})$.

Рассмотрим теперь двухшаговую задачу: присоединим к n-му шагу (n-1)-й (рис. 2).

Для любых состояний s_{n-2}, произвольных управлений X_{n-1} и оптимальном управлении на n-м шаге значение целевой функции на двух последних шагах равно:

$$f_{n-1}(s_{n-2}, X_{n-1}) + Z^*_n(s_{n-1}). \quad \textbf{(6)}$$

Согласно принципу оптимальности для любых s_n-i решение нужно выбирать так, чтобы оно вместе с оптимальным управлением на последнем (n-м) шаге приводило бы к максимуму целевой функции на двух последних шагах. Следовательно, нужно найти максимум выражения (6) по всем допустимым управлениям X_{n-1} Максимум этой суммы зависит от s_{n-2}, обозначается через $Z^*_n(s_{n-2})$ и называется *условным максимумом целевой функции при оптимальном управлении на двух последних шагах.* Соответствующее управление X_{n-1} на (n—1)-м шаге обозначается через

$X^*_{n-1}(s_n\text{-}2)$ и называется *условным оптимальным управлением* на (n-1)-м шаге.

$$Z^*_{n-1}(s_{n-2}) = \max_{[X_{n-1}]} \left\{ f_{n-1}(s_{n-2}, X_{n-1}) + Z^*_n(s_{n-1}) \right\}. \quad \textbf{(7)}$$

Следует обратить внимание на то, что выражение, стоящее в фигурных скобках (7), зависит только от S_{n-2} и X_{n-1}, так как можно найти из уравнения состояний (2) при $к=n—1$

$$s_{n-1} = \varphi_{n-1}(s_{n-2}, X_{n-1})$$

и подставить вместо s_{n-1} в функцию $Z_n^*(s_{n-1})$.

В результате максимизации только по одной переменной X_{n-1} согласно уравнению (7) вновь получаются две функции:

$$Z^*_{n-1}(s_{n-2}) \text{ и } X^*_{n-1}(s_{n-2}).$$

Далее рассматривается трехшаговая задача: к двум последним шагам присоединяется (n—2)-й и т. д.

Обозначим через $Z^*_k(s_{n-1})$ *условный максимум целевой функции, полученный при оптимальном управлении на n~к+1 шагах, начиная с*

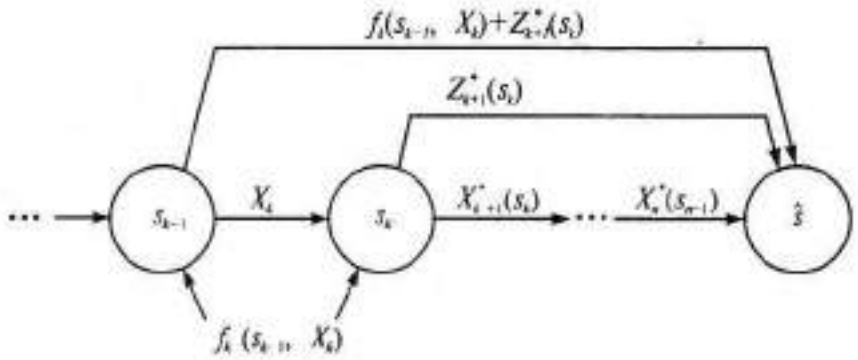
Рис. 3

k-го до конца, при условии, что к началу k-го шага система находилась в состоянии s_{k-1}. Фактически эта функция равна

$$Z_k^*(s_{k-1}) = \max_{\{(x_k,\dots,x_n)\}} \sum_{i=k}^{n} f_i(s_{i-1}, X_i).$$

Тогда

$$Z_{k+1}^*(s_k) = \max_{\{(x_{k+1},\dots,x_n)\}} \sum_{i=k+1}^{n} f_i(s_{i-1}, X_i).$$



Целевая функция на *n—к* последних шагах (рис. 3) при произвольном управлении $X_к$ на *к-м* шаге и оптимальном управлении на последующих *n—к* шагах равна

$$f_k(s_{k-1}, X_k) + Z^*_{k+1}(s_k).$$

Согласно принципу оптимальности, $X_к$ выбирается из условия максимума этой суммы, т.е.

$$Z_k^*(s_{k-1}) = \max_{\{X_k\}}\left\{ f_k(s_{k-1}, X_k) + Z^*_{k+1}(s_k)\right\}, \qquad \textbf{(8)}$$

$$k = n-1,\ n-2,\ \dots,\ 2,\ 1.$$

Управление $X_к$ на *к-м* шаге, при котором достигается максимум в (8), обозначается через $X^*_k(s_{k-1})$ и называется *условным оптимальным управлением на к-м шаге* (в правую часть уравнения(8) следует вместо s_k подставить выражение $s_k=\varphi_k(s_{k-1}, X_k)$, найденное из уравнений состояния).

Уравнения (8) называют *уравнениями Беллмана.* Это рекуррентные соотношения, позволяющие найти предыдущее значение функции, зная последующие. Если из (5) найти $Z^*_n(s_{n-1})$, то при *к=n*— 1 из (8) можно определить, решив задачу максимизации для всех возможных значений $s_{n~2}$, выражения для $Z^*_{n-1}(s_n-_2)$ и соответствующее $X^*_{n-1}(s_n-_2)$- Далее, зная $Z^*_{n-1}(s_n-_2)$, находим, используя (8) и (2), уравнения состояний.

Процесс решения уравнений (5) и (8) называется *условной оптимизацией.*

В результате условной оптимизации получаются две последовательности:

$$Z_n^*(s_{n-1}),\ Z_{n-1}^*(s_{n-2}),\ \dots,\ Z_2^*(s_1),\ Z_1^*(s_0) -$$

условные максимумы целевой функции на последнем, на двух последних, на ...n шагах и

$$X_n^*(s_{n-1}),\ X_{n-1}^*(s_{n-2}),\ \ldots,\ X_2^*(s_1),\ X_1^*(s_0) -$$

условные оптимальные управления на n-м, (n-1)-м, ..., 1-м шагах. Используя эти последовательности, можно найти решение задачи ДП при данных n и s0. По определению Z1*(s0) — условный максимум целевой функции за n шагов при условии, что к началу 1-го шага система была в состоянии s0, т.е.

$$Z_{max} = Z_1^*(s_0). \qquad (9)$$

Далее следует использовать последовательность условных оптимальных управлений и уравнения состояний (2).

При фиксированном s0 получаем X1*= X1*(s0). Далее из уравнений (2) находим s1*=φ(s0, X1*) и подставляем это выражение в последовательность условных оптимальных управлений:

2. Практическая часть

. Задача о распределении средств между предприятиями

Планируется деятельность четырех промышленных предприятий (системы) на очередной год. Начальные средства: s0 =5 усл. ед.

x	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
1	8	6	3	4
2	10	9	4	6
3	11	11	7	8
4	12	13	11	13
5	18	15	18	16

Размеры вложения в каждое предприятие кратны 1 усл. ед. Средства x, выделенные k:-му предприятию *(к=1,* 2, 3, 4), приносят в конце года прибыль $f_k(x)$. Функции $f_k(x)$ заданы таблично (табл. 1). Принято считать, что:

а) прибыль $f_k(x)$ не зависит от вложения средств в другие предприятия;

б) прибыль от каждого предприятия выражается в одних условных единицах;

в) суммарная прибыль равна сумме прибылей, полученных от каждого предприятия.

Определить, какое количество средств нужно выделить каждому предприятию, чтобы суммарная прибыль была наибольшей.

Таблица 1

x	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
1	8	6	3	4
2	10	9	4	6
3	11	11	7	8
4	12	13	11	13
5	18	15	18	16

Решение. Обозначим через x* количество средств, выделенных *к-му* предприятию. (Нумерацию предприятий 1, 2, 3, 4 сохраняем в процессе решения неизменной.)

Суммарная прибыль равна

$$Z=\sum_{k=1}^{4} f_k(x_k). \qquad \textbf{(1)}$$

Переменные $x_к$ удовлетворяют ограничениям:

$$\sum_{k=1}^{4} x_k = 5,$$
$$x_k \geq 0,\ k = 1,2,3,4.$$

Требуется найти переменные x_1, x_2, x_3, x_4, удовлетворяющие системе ограничений (2) и обращающие в максимум функцию 1).

Особенности модели. Ограничения линейные, но переменные целочисленные, а функции $f_r(x)$ заданы таблично] поэтому нельзя применить методы целочисленного линейного программирования .

Схема решения задачи методом ДП имеет следующий вид: процесс решения распределения средств s_0 =5 МОЖНО рассматривать как шаговый, номер шага совпадет с номером предприятия; выбор временных x_1, x_2, x_3, x_4,— управление соответственно на I, II, III**,** *V* шагах, *s* — конечное состояние процесса распределения — равно нулю, так как все средства должны быть вложены в производство, *s=0* Схема распределения показана на рис. 1.

Уравнения состояний (12.2) в данной задаче имеют вид:

$$s_k=s_{k-1}-x_k,\ \ k=1,\ 2,\ 3,\ 4, \qquad (3)$$

Сравнивая подчеркнутые числа, получим Z_1^* (5)=24 усл. ед. = Zmax при $x_1^*=x_1^*(5)= 1$.

Используя уравнения (3), получим s_1^* =5-1=4, а по табл. 2 в столбце 9 находим $x_2^* = x_2^*(4)$ - 2. Далее находим $s_2^*=4-2=2$, а по табл. 2 в столбце 6 — $x_3^*= x_3^*\{2\} = 1$. Наконец, $s_3^*=2-1=1$ и $x_4^*= x_4^*\{1\} = 1$ т.е. $X^*(1;2;1;1)$.

Максимум суммарной прибыли равен 24 усл. ед. средств при условии, что 1-му предприятию выделено 1 усл. ед.; 2-му предприятию — 2 усл. ед.; 3-му предприятию — 1 усл. ед.; 4-му предприятию — 1 усл.ед. **(2)**

Замечание 1. Решение четырехмерной задачи на определение условного экстремума сведено фактически к решению четырех одномерных задач: на каждом шаге определялась одна переменная *x.*

Замечание 2. На разобранной задаче видно, что метод ДП безразличен к виду и способу задания функции: $f_k(x)$ были заданы таблично, поэтому и $Z^*_k(s)$ и $X^*_k(s)$ принимали дискретные значения, представленные в табл. 2.

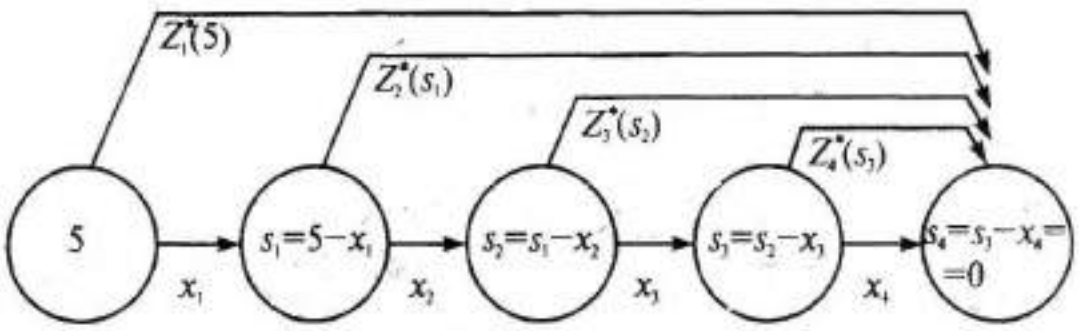


Рис.1

Таблица 2

s_{k-1}	x_k	s_k	$k=3$			$k=2$			$k=1$		
			$f_3(x_3)+Z_4^*(s_3)$	$Z_3^*(s_2)$	$x_3^*(s_2)$	$f_2(x_2)+Z_3^*(s_2)$	$Z_2^*(s_1)$	$x_2^*(s_1)$	$f_1(x_1)+Z_2^*(s_1)$	$Z_1^*(s_0)$	$x_1^*(s_0)$
1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0+1=1			0+4=4			0+6=6	8	1
	1	0	3+0=3	4	0	6+0=6	6	1	8+0=8		
2	0	2	0+6=6			0+7=7			0+10=10		
	1	1	3+4=7	7	1	6+4=10	10	1	8+6=14	14	1
	2	0	4+0=4			9+0=9			10+0=10		
3	0	3	0+8=8			0+9=9			0+13=13		
	1	2	3+6=9	9	1	6+7=13	13	1	8+10=18	18	1
	2	1	4+4=8			9+4=13		2	10+6=16		

s_{k-1}	x_k	s_k	$k=3$			$k=2$			$k=1$		
			$f_3(x_3)+Z_4^*(s_3)$	$Z_3^*(s_2)$	$x_3^*(s_2)$	$f_2(x_2)+Z_3^*(s_2)$	$Z_2^*(s_1)$	$x_2^*(s_1)$	$f_1(x_1)+Z_2^*(s_1)$	$Z_1^*(s_0)$	$x_1^*(s_0)$
1	2	3	4	5	6	7	8	9	10	11	12
	3	0	7+0=7			11+0=11			11+0=11		
4	0	4	0+13=13			0+13=13			0+16=16		
	1	3	3+8=11			6+9=15			8+13=21		
	2	2	4+6=10	13	0	9+7=16	16	2	10+10=20	21	1
	3	1	7+4=11			11+4=15			11+6=17		
	4	0	11+0=11			13+0=13			12+0=12		
5	0	5	0+16=16			0+18=18			0+19=19		
	1	4	3+13=16			6+13=19			8+16=24		
	2	3	4+8=12	18	5	9+9=18	19	1	10+13=23	24	1
	3	2	7+6=13			11+7=18			11+10=21		
	4	1	11+4=15			13+4=17			12+6=18		
	5	0	18+0=18			15+0=15			18+0=18		

Введем в рассмотрение функцию $Z_k^*(s_{k-1})$ — условную оптимальную прибыль, полученную от *k-го,* (k+1)-го, ..., 4-го предприятий, если между ними распределялись оптимальным образом средства $s_{k-1}(0< s_{k-1}<5)$. Допустимые управления на *к-м* шаге удовлетворяют условию: $0< X_k < s_{k-1}$ (либо *к-му* предприятию ничего не выделяем, $x_k=0$, либо не больше того, что имеем к *к-му* шагу,

Уравнения (5) и (8) имеют вид:

$$k=4,\ s_4 = 0 \Rightarrow Z_4^*(s_3) = \max_{0\le x_4 \le s_3} f_4(x_4), \qquad \text{(а)}$$

$$Z_3^*(s_2) = \max_{0\le x_3 \le s_2} \{f_3(x_3) + Z_4^*(s_3)\}, \qquad \text{(б)}$$

$$Z_2^*(s_1) = \max_{0\le x_2 \le s_1} \{f_2(x_2) + Z_3^*(s_2)\}, \qquad \text{(в)}$$

$$Z_1^*(5) = \max_{0\le x_1 \le 5} \{f_1(x_1) + Z_2^*(s_1)\}. \qquad \text{(г)}$$

Последовательно решаем записанные уравнения, проводя условную оптимизацию *(см.* рис. 1) каждого шага.

IV ш а г. В табл. 1 $f_4\{x)$ прибыли монотонно возрастают, поэтому все средства, оставшиеся к IV шагу, следует вложить в 4-е предприятие. При этом для возможных значений $s_з=0, 1, \ldots, 5$ получим:

$$Z_4^*(s_3) = f_4(s_3) \text{ и } x_4^*(s_3) = s_3.$$

I l l шаг. Делаем все предположения относительно остатка средств s_2 к III шагу (т.е. после выбора X_1 и x_2). S_2 может принимать значения 0, 1, 2, 3, 4, 5 (например, $s_2 = 0$, если все средства отданы 1-му и 2-му предприятиям, $s_2=5$, если 1-е и 2-е предприятия ничего не получили, и т.д.). В зависимости от этого выбираем $0<x_з<s_2>$ находим $S_3=\sim S_2—x_з$ и сравниваем для разных x_3, при фиксированном s_2 значения суммы $f_3(x_3) + Z_4^*(s_3)$. Для каждого s_2 наибольшее из этих значений есть $Z_3^*(s_2)$ — условная оптимальная прибыль, полученная при оптимальном распределении средств s_2 между 3-м и 4-м предприятиями. Оптимизация дана в табл. 2 при k=3. Для каждого значения s_2 $Z_3^*(s_2)$ и $X_3^*(s_2)$ помещены в графах 5 и 6 соответственно.

II шаг. Условная оптимизация, согласно уравнению (в), проведена в табл. 2 при *k=2.* Для всех возможных значений s_1 значения $Z_2^*(s_1)$ и $X_2^*(s_1)$ находятся в столбцах 8 и 9 соответственно; первые слагаемые в столбце 7 — значения $f_2\{x_2)$, взяты из табл. 1 а вторые слагаемые взяты из столбца 5 табл. 2 при $s_2 = s_1 — x_1$.

I шаг. Условная оптимизация (уравнение (г) проведена в табл. 2 при *k=1* для $s_0=5$. Поясним решение подробно: если $x_1=0$, то $s_1=5$, прибыль, полученная от четырех предприятий при условии, что $s_1=5$ ед. средств между оставшимися тремя предприятиями будут распределены оптимально, равна $f_1(0)+Z_2^*(5)=0+19=19$ *($Z_2^*(5)$* взято из столбца 9 табл. 2 при $s_1=5$). Если $x_1=1$, то $s_2=4$. Суммарная прибыль при условии, что $s_2=4$ ед. средств между оставшимися тремя предприятиями будут распределены оптимально, равна $f_1(l)+Z_2^*(4)=8+16=24$ $f_1(l)$ взято из табл. 1, а $Z_2^*(4)$ — из столбца 9 табл. 2.) Аналогично при $x_1,=2$, $s_2=3$ $F_1(2)+Z_2(3)=10+13=23$;

при $x_1=3$, $s_2=2$ и $f_1(3)+Z_2^*(3)=11+10=\underline{21}$;

при $x_1=4$, $s_2=1$ и $f_1(4)+Z_2^*(1)=12+16=\underline{28}$;

при $x_1=5$, $s_2=0$ и $f_1(5)+Z_2^*(0)=18+0=\underline{18}$.

В результате найдены два оптимальных решения: $X^{(1)}$*(1; 1; 1; 1) и $X^{(2)}$* (1; 2; 0; 1). Если начальные средства увеличились, например, на 1 усл. ед., s_o=6, а функции прибыли $f_{k(}x)$ остались прежними, то в табл. 2 достаточно добавить раздел для $s_o=6$ при k=3, 2, 1; этот фрагмент расчетов помещен в табл. 3.

Таблица 3

1	2	3	4	5	6	7	8	9	10	11	12
	0	6	0+16=16			0+22=22			0+24=24		
	1	5	3+16=19			6+18=24			8+19=27		
6	2	4	4+13=17	22	5	9+13=22	24	1	10+16=26	27	1
	3	3	7+8=15			11+9=20			11+13=24		
	4	2	11+6=17			13+7=20			12+10=22		
	5	1	18+4=22			15+4=19			18+6=24		

Получаем $Z_{max}=27$, $x_1^*=1 \rightarrow s_1^*=6-1=5 \Rightarrow x_2^*=1 \rightarrow s_2^*=5-1=$
$=4 \Rightarrow x_3^*=0 \rightarrow s_3^*=4-0=4 \Rightarrow x_4^*=4$.

Оптимальное решение X (1; 1; 0; 4).

Если принято решение распределить средства s_o=5 между 2-, 3- и 4-м предприятиями, то задача уже решена в табл.2. В разделе $k=2$ таблицы находим $Z_{max}= Z_2$*(5)=19 при условии, **что $x^*_2=1$, x_3*=0 $x^*_4=4$.**

Наконец, если увеличилось количество предприятий (число шагов), то схему можно дополнить, присоединяя шаги с номерами k=0, 1, … и т. д. Например, пусть средства ξ,o = 6 распределяются между пятью предприятиями. Функция прибыли для пятого предприятия задана формулой f(x)=3x+1, если x≠0 и f(0)=0. Присвоим 5-му предприятию номер k=0, тогда x_0 — средства, выделенные этому предприятию. Обозначим через Z_Q^*(6) оптимальную прибыль, полученную от пяти предприятий:

$$Z_0^*(6) = \max_{0\le x_0\le 6}\left\{f_0(x_0)+Z_1^*(s_1)\right\},$$

а $s_1=6—X_Q$.

Условная оптимизация 0-го шага дана в табл. .4

Таблица 4

x_0	0	1	2	3	4	5	6
$s_1=6-x_0$	6	5	4	3	2	1	0
$f(0)=0$	0	4	7	10	13	16	19
$Z_1^*(s_1)$ (взята из табл. 12.2 и 12.3 при $k=1$)	27	24	21	18	14	8	0
$f(x_0)+ Z_1(s_1)$	27	28	28	28	27	24	19

Следовательно, $Z_{max}=28$, а оптимальных решений четыре: X_1^*(1; 2; **1** 1), X_2^* (2\ **1;** 1; 1; 1), X_3^* (2; 1; 2; **0;** 1), X_4^* (3; 1; 1; 0; 1).

Замечание К недостаткам метода по-прежнему следует отнести возникновение технических трудностей при вычислениях в случае увеличения размерности. Если каждое управление X_k^* будет зависеть от r переменных, а состояние s_k^* — от p параметров, то на каждом шаге возникает р-мерная задача оптимизации

3 Задания для самостоятельной работы

Задача 1. Для двух предприятий выделено a единиц средств. Как распределить все средства в течение 4 лет, чтобы доход был наибольшим, если известно, что доход от x единиц средств, вложенных в первое предприятие, равен $f_1(x)$, а доход от y единиц средств, вложенных во второе предприятие, равен $f_2(y)$. Остаток средств к концу года составляет $g_1(x)$ для первого предприятия и $g_2(y)$ для второго предприятия. Задачу решить методом динамического программирования.

a	f_1	g_1	f_2	g_2
1000	3x	0,1x	2y	0,5y

Задача 2. Планируется распределение начальной суммы X_0 млн. р. Между четырьмя предприятиями некоторого объединения. Средства выделяются только в размерах кратных $a=80$ млн. р. Функции прироста продукции от вложенных средств на каждом предприятии заданы таблично. Требуется так распределить вложения между предприятиями, чтобы общий прирост продукции (в млн. р.) был максимальным. Решить задачу на основе функционального уравнения Беллмана.

X_0	Вкладываемые средства X	Функции прироста продукции на предприятии			
		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
400	0	10	15	13	14
	80	13	20	17	16
	160	16	22	21	23
	240	21	25	26	25
	320	25	30	28	27
	400	25	32	30	32

Тема 2.5 Алгоритмы на графах
Занятие 4 Решение задач с использованием графов
1 Теоретическая часть

Алгоритм Флойда (поиск всех кратчайших путей в графе)

Алгоритм Флойда является одним из методов поиска кратчайших путей в графе. В отличии от алгоритма Дейкстры, который позволяет при доведении до конца построить ориентированное дерево кратчайших путей от некоторой вершины, метод Флойда позволяет найти длины всех кратчайших путей в графе. Конечно эта задача может быть решена и многократным применением алгоритма Дейкстры (каждый раз последовательно выбираем вершину от первой до N-ной, пока не получим кратчайшие пути от всех вершин графа), однако реализация подобной процедуры потребовала бы значительных вычислительных затрат.

Прежде чем представлять алгоритмы, необходимо ввести некоторые обозначения. Перенумеруем вершины исходного графа целыми числами от 1 до N. Обозначим через $d_{i,j}^m$ длину кратчайшего пути из вершинм i в вершину j, который в качестве промежуточных может содержать только первые m вершин графа. (Напомним, что промежуточной вершиной пути является любая принадлежащая ему вершина, не совпадающая с его начальной или конечной вершинами.) Если между вершинами i и j не существует ни одного пути указанного типа, то условно будем считать, что $d_{i,j}^m=\infty$. Из данного определения величин $d_{i,j}^m$ следует, что величина $d_{i,j}^0$, представляет длину кратчайшего пути из вершины i в вершину j, не имеющего промежуточных вершин, т. е. длину кратчайшей дуги, соединяющей i с j (если такие дуги присутствуют в графе). для любой вершины i положим $d_{i,i}^m= 0$. Отметим далее, что величина $d_{i,j}^m$представляет длину кратчайшего пути между вершинами i и j.

Обозначим через D^m матрицу размера NxN, элемент (i, j) которой совпадает с $d_{i,j}^m$. Если в исходном графе нам известна длина каждой дуги, то мы можем сформировать матрицу D^0. Наша цель состоит в определении матрицы D^N, представляющей кратчайшие пути между всеми вершинами рассматриваемого графа.

В алгоритме Флойда в качестве исходной выступает матрица D^0. Вначале из этой матрицы вычисляется матрица D^1. Затем по матрице D^1 вычисляется матрицав D^2 и т. д. Процесс повторяется до тех пор, пока по матрице D^{N-1} не будет вычислена матрица D^N.

Рассмотрим основную идею, лежащую в основе алгоритма Флойда. Суть алгоритма Флойда заключается в проверке того, не окажется ли путь из вершины i в вершину j короче, если он будет проходить через некоторую промежуточную вершину m. Предположим, что нам известны:

1. кратчайший путь из вершины i в вершину m, в котором в качестве промежуточных допускается использование только первых (m - 1) вершин;
2. кратчайший путь из вершины m в вершину j, в котором в качестве промежуточных допускается использование только первых (m - 1) вершин;
3. кратчайший путь из вершины i в вершину j, в котором в качестве промежуточных допускается использование только первых (m - 1) вершин.

Поскольку по предположению исходный граф не может содержать контуров отрицательной длины, один из двух путей — путь, совпадающий с представленным в пункте 3, или путь, являющийся объединением путей из пунктов 1 и 2 — должен быть кратчайшим путем из вершины i в вершину j, в котором в качестве промежуточных допускается использование только первых m вершин. Таким образом,

$$\mathbf{d_{i,j}^m = min\{ d_{i,m}^{m-1} + d_{m,j}^{m-1}; d_{i,j}^{m-1}\}}$$

Из соотношения видно, что для вычисления элементов матрицы D^m необходимо располагать лишь элементами матрицы D^{m-1}. Более того, соответствующие вычисления могут быть проведены без обращения к исходному графу. Теперь мм в состоянии датьформальное описание алгоритма Флойда для нахождения на графе кратчайших путей между всеми парами вершин.

Алгоритм

1. Перенумеровать вершины графа от 1 до N целыми числами, определить матрицу D^0, каждый элемент $d_{i,j}$ которой есть длина кратчайшей дуги между вершинами i и j. Если такой дуги нет, положить значение элемента равным ∞. Кроме того, положить значения диагонального элемента $d_{i,i}$равным 0.
2. Для целого m, последовательно принимающего значения 1...N определить по элементам матрицы D^{m-1} элементы D^m
3. Алгоритм заканчивается получением матрицы всех кратчайших путей D^N, N – число вершин графа.

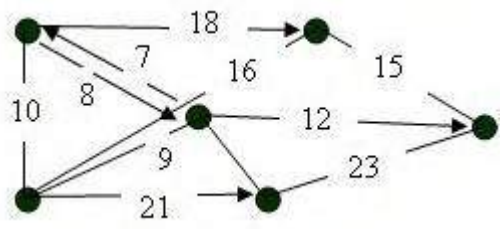
Напомним, для определения по известным элементам матрицы D^{m-1} элементов матрицы D^m в алгоритме Флойда применяется рекурсивное соотношение:

$$\mathbf{d_{i,j}^m = min\{ d_{i,m}^{m-1} + d_{m,j}^{m-1}; d_{i,j}^{m-1}\}}$$

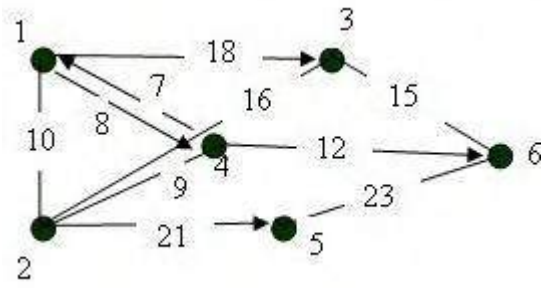
$d_{i,j}^m$ – элемент матрицы D^m, $d_{i,j}^{m-1}$ – элементы матрицы D^{m-1} найденой на предыдущем шаге алгоритма.

2 Практическая часть

Необходимо найти кратчайшие пути между каждой парой вершин в графе, представленом на рисунке:



Пронумеруем все вершины графа, и составим матрицу длин кратчайших дуг D^0, в случае, если дуги между вершиной i и j не существует, элементу $d_{i,j}$ матрицы присваивается значение ∞. Исходный граф с пронумерованными вершинами представлен на рисунке ниже.



Матрица D^0:

$D^0=$	0	10	18	8	∞	∞
	10	0	16	9	21	∞
	∞	16	0	∞	∞	15
	7	9	∞	0	∞	12
	∞	∞	∞	∞	0	23
	∞	∞	15	∞	23	0

На основании матрицы D^0, вычислим последовательно все элементы матрицы D^1. Для этого мы используем рекуррентное соотношение $d_{i,j}^1 = \min\{ d_{i,1}^0 + d_{1,j}^0; d_{i,j}^0\}$.

$d_{1,1}^1=\min\{d_{1,1}^0+d_{1,1}^0 d_{1,1}^0\}=\min\{0+0;0\}=0$
$d_{1,2}^1=\min\{d_{1,1}^0+d_{1,2}^0 d_{1,2}^0\}=\min\{0+10;10\}=10$
$d_{1,3}^1=\min\{d_{1,1}^0+d_{1,3}^0 d_{1,3}^0\}=\min\{0+18;18\}=18$
$d_{1,4}^1=\min\{d_{1,1}^0+d_{1,4}^0 d_{1,4}^0\}=\min\{0+8;8\}=8$
$d_{1,5}^1=\min\{d_{1,1}^0+d_{1,5}^0 d_{1,5}^0\}=\min\{0+\infty;\infty\}=\infty$
$d_{1,6}^1=\min\{d_{1,1}^0+d_{1,6}^0 d_{1,6}^0\}=\min\{0+\infty;\infty\}=\infty$
$d_{2,1}^1=\min\{d_{2,1}^0+d_{1,1}^0 d_{2,1}^0\}=\min\{10+0;10\}=10$
$d_{2,2}^1=\min\{d_{2,1}^0+d_{1,2}^0 d_{2,2}^0\}=\min\{10+10;0\}=0$
$d_{2,3}^1=\min\{d_{2,1}^0+d_{1,3}^0 d_{2,3}^0\}=\min\{10+18;16\}=16$
$d_{2,4}^1=\min\{d_{2,1}^0+d_{1,4}^0 d_{2,4}^0\}=\min\{10+8;9\}=9$
$d_{2,5}^1=\min\{d_{2,1}^0+d_{1,5}^0 d_{2,5}^0\}=\min\{10+\infty;21\}=21$
$d_{2,6}^1=\min\{d_{2,1}^0+d_{1,6}^0 d_{2,6}^0\}=\min\{10+\infty;\infty\}=\infty$
$d_{3,1}^1=\min\{d_{3,1}^0+d_{1,1}^0 d_{3,1}^0\}=\min\{\infty+0;\infty\}=\infty$
$d_{3,2}^1=\min\{d_{3,1}^0+d_{1,2}^0 d_{3,2}^0\}=\min\{\infty+10;16\}=16$
$d_{3,3}^1=\min\{d_{3,1}^0+d_{1,3}^0 d_{3,3}^0\}=\min\{\infty+18;0\}=0$
$d_{3,4}^1=\min\{d_{3,1}^0+d_{1,4}^0 d_{3,4}^0\}=\min\{\infty+8;\infty\}=\infty$
$d_{3,5}^1=\min\{d_{3,1}^0+d_{1,5}^0 d_{3,5}^0\}=\min\{\infty+\infty;\infty\}=\infty$
$d_{3,6}^1=\min\{d_{3,1}^0+d_{1,6}^0 d_{3,6}^0\}=\min\{\infty+\infty;15\}=15$
$d_{4,1}^1=\min\{d_{4,1}^0+d_{1,1}^0 d_{4,1}^0\}=\min\{7+0;7\}=7$
$d_{4,2}^1=\min\{d_{4,1}^0+d_{1,2}^0 d_{4,2}^0\}=\min\{7+10;9\}=9$
$d_{4,3}^1=\min\{d_{4,1}^0+d_{1,3}^0 d_{4,3}^0\}=\min\{7+18;\infty\}=25$
$d_{4,4}^1=\min\{d_{4,1}^0+d_{1,4}^0 d_{4,4}^0\}=\min\{7+8;0\}=0$
$d_{4,5}^1=\min\{d_{4,1}^0+d_{1,5}^0 d_{4,5}^0\}=\min\{7+\infty;\infty\}=\infty$
$d_{4,6}^1=\min\{d_{4,1}^0+d_{1,6}^0 d_{4,6}^0\}=\min\{7+\infty;12\}=12$
$d_{5,1}^1=\min\{d_{5,1}^0+d_{1,1}^0 d_{5,1}^0\}=\min\{\infty+0;\infty\}=\infty$
$d_{5,2}^1=\min\{d_{5,1}^0+d_{1,2}^0 d_{5,2}^0\}=\min\{\infty+10;\infty\}=\infty$
$d_{5,3}^1=\min\{d_{5,1}^0+d_{1,3}^0 d_{5,3}^0\}=\min\{\infty+18;\infty\}=\infty$
$d_{5,4}^1=\min\{d_{5,1}^0+d_{1,4}^0 d_{5,4}^0\}=\min\{\infty+8;\infty\}=\infty$
$d_{5,5}^1=\min\{d_{5,1}^0+d_{1,5}^0 d_{5,5}^0\}=\min\{\infty+\infty;0\}=0$
$d_{5,6}^1=\min\{d_{5,1}^0+d_{1,6}^0 d_{5,6}^0\}=\min\{\infty+\infty;23\}=23$
$d_{6,1}^1=\min\{d_{6,1}^0+d_{1,1}^0 d_{6,1}^0\}=\min\{\infty+0;\infty\}=\infty$
$d_{6,2}^1=\min\{d_{6,1}^0+d_{1,2}^0 d_{6,2}^0\}=\min\{\infty+10;\infty\}=\infty$
$d_{6,3}^1=\min\{d_{6,1}^0+d_{1,3}^0 d_{6,3}^0\}=\min\{\infty+18;15\}=15$
$d_{6,4}^1=\min\{d_{6,1}^0+d_{1,4}^0 d_{6,4}^0\}=\min\{\infty+8;\infty\}=\infty$
$d_{6,5}^1=\min\{d_{6,1}^0+d_{1,5}^0 d_{6,5}^0\}=\min\{\infty+\infty;23\}=23$
$d_{6,6}^1=\min\{d_{6,1}^0+d_{1,6}^0 d_{6,6}^0\}=\min\{\infty+\infty;0\}=0$

Представим матрицу D^1, включив в нее рассчитанные элементы.

$D^1=$						
	0	10	18	8	∞	∞
	10	0	16	9	21	∞
	∞	16	0	∞	∞	15
	7	9	25	0	∞	12
	∞	∞	∞	∞	0	23
	∞	∞	15	∞	23	0

На основании матрицы D^1, вычислим последовательно все элементы матрицы D^2. Для этого мы используем рекуррентное соотношение $d_{i,j}^2=\min\{ d_{i,2}^1+ d_{2,j}^1; d_{i,j}^1\}$.
$d_{1,1}^2=\min\{d_{1,2}^1+d_{2,1}^1 d_{1,1}^1\}=\min\{10+10;0\}=0$
$d_{1,2}^2=\min\{d_{1,2}^1+d_{2,2}^1 d_{1,2}^1\}=\min\{10+0;10\}=10$
$d_{1,3}^2=\min\{d_{1,2}^1+d_{2,3}^1 d_{1,3}^1\}=\min\{10+16;18\}=18$
$d_{1,4}^2=\min\{d_{1,2}^1+d_{2,4}^1 d_{1,4}^1\}=\min\{10+9;8\}=8$
$d_{1,5}^2=\min\{d_{1,2}^1+d_{2,5}^1 d_{1,5}^1\}=\min\{10+21;\infty\}=31$
$d_{1,6}^2=\min\{d_{1,2}^1+d_{2,6}^1 d_{1,6}^1\}=\min\{10+\infty;\infty\}=\infty$
$d_{2,1}^2=\min\{d_{2,2}^1+d_{2,1}^1 d_{2,1}^1\}=\min\{0+10;10\}=10$
$d_{2,2}^2=\min\{d_{2,2}^1+d_{2,2}^1 d_{2,2}^1\}=\min\{0+0;0\}=0$
$d_{2,3}^2=\min\{d_{2,2}^1+d_{2,3}^1 d_{2,3}^1\}=\min\{0+16;16\}=16$
$d_{2,4}^2=\min\{d_{2,2}^1+d_{2,4}^1 d_{2,4}^1\}=\min\{0+9;9\}=9$
$d_{2,5}^2=\min\{d_{2,2}^1+d_{2,5}^1 d_{2,5}^1\}=\min\{0+21;21\}=21$
$d_{2,6}^2=\min\{d_{2,2}^1+d_{2,6}^1 d_{2,6}^1\}=\min\{0+\infty;\infty\}=\infty$
$d_{3,1}^2=\min\{d_{3,2}^1+d_{2,1}^1 d_{3,1}^1\}=\min\{16+10;\infty\}=26$
$d_{3,2}^2=\min\{d_{3,2}^1+d_{2,2}^1 d_{3,2}^1\}=\min\{16+0;16\}=16$
$d_{3,3}^2=\min\{d_{3,2}^1+d_{2,3}^1 d_{3,3}^1\}=\min\{16+16;0\}=0$
$d_{3,4}^2=\min\{d_{3,2}^1+d_{2,4}^1 d_{3,4}^1\}=\min\{16+9;\infty\}=25$
$d_{3,5}^2=\min\{d_{3,2}^1+d_{2,5}^1 d_{3,5}^1\}=\min\{16+21;\infty\}=37$
$d_{3,6}^2=\min\{d_{3,2}^1+d_{2,6}^1 d_{3,6}^1\}=\min\{16+\infty;15\}=15$
$d_{4,1}^2=\min\{d_{4,2}^1+d_{2,1}^1 d_{4,1}^1\}=\min\{9+10;7\}=7$
$d_{4,2}^2=\min\{d_{4,2}^1+d_{2,2}^1 d_{4,2}^1\}=\min\{9+0;9\}=9$
$d_{4,3}^2=\min\{d_{4,2}^1+d_{2,3}^1 d_{4,3}^1\}=\min\{9+16;25\}=25$
$d_{4,4}^2=\min\{d_{4,2}^1+d_{2,4}^1 d_{4,4}^1\}=\min\{9+9;0\}=0$
$d_{4,5}^2=\min\{d_{4,2}^1+d_{2,5}^1 d_{4,5}^1\}=\min\{9+21;\infty\}=30$
$d_{4,6}^2=\min\{d_{4,2}^1+d_{2,6}^1 d_{4,6}^1\}=\min\{9+\infty;12\}=12$
$d_{5,1}^2=\min\{d_{5,2}^1+d_{2,1}^1 d_{5,1}^1\}=\min\{\infty+10;\infty\}=\infty$
$d_{5,2}^2=\min\{d_{5,2}^1+d_{2,2}^1 d_{5,2}^1\}=\min\{\infty+0;\infty\}=\infty$
$d_{5,3}^2=\min\{d_{5,2}^1+d_{2,3}^1 d_{5,3}^1\}=\min\{\infty+16;\infty\}=\infty$
$d_{5,4}^2=\min\{d_{5,2}^1+d_{2,4}^1 d_{5,4}^1\}=\min\{\infty+9;\infty\}=\infty$
$d_{5,5}^2=\min\{d_{5,2}^1+d_{2,5}^1 d_{5,5}^1\}=\min\{\infty+21;0\}=0$
$d_{5,6}^2=\min\{d_{5,2}^1+d_{2,6}^1 d_{5,6}^1\}=\min\{\infty+\infty;23\}=23$
$d_{6,1}^2=\min\{d_{6,2}^1+d_{2,1}^1 d_{6,1}^1\}=\min\{\infty+10;\infty\}=\infty$
$d_{6,2}^2=\min\{d_{6,2}^1+d_{2,2}^1 d_{6,2}^1\}=\min\{\infty+0;\infty\}=\infty$
$d_{6,3}^2=\min\{d_{6,2}^1+d_{2,3}^1 d_{6,3}^1\}=\min\{\infty+16;15\}=15$
$d_{6,4}^2=\min\{d_{6,2}^1+d_{2,4}^1 d_{6,4}^1\}=\min\{\infty+9;\infty\}=\infty$
$d_{6,5}^2=\min\{d_{6,2}^1+d_{2,5}^1 d_{6,5}^1\}=\min\{\infty+21;23\}=23$
$d_{6,6}^2=\min\{d_{6,2}^1+d_{2,6}^1 d_{6,6}^1\}=\min\{\infty+\infty;0\}=0$
Представим матрицу D^2, включив в нее рассчитанные элементы.

$D^2=$	0	10	18	8	31	∞
	10	0	16	9	21	∞
	26	16	0	25	37	15

	7	9	25	0	30	12
	∞	∞	∞	∞	0	23
	∞	∞	15	∞	23	0

На основании матрицы D^2, вычислим последовательно все элементы матрицы D^3. Для этого мы используем рекуррентное соотношение $d_{i,j}^3=\min\{ d_{i,3}^2+ d_{3,j}^2; d_{i,j}^2\}$.
$d_{1,1}^3=\min\{d_{1,3}^2+d_{3,1}^2 d_{1,1}^2\}=\min\{18+26;0\}=0$
$d_{1,2}^3=\min\{d_{1,3}^2+d_{3,2}^2 d_{1,2}^2\}=\min\{18+16;10\}=10$
$d_{1,3}^3=\min\{d_{1,3}^2+d_{3,3}^2 d_{1,3}^2\}=\min\{18+0;18\}=18$
$d_{1,4}^3=\min\{d_{1,3}^2+d_{3,4}^2 d_{1,4}^2\}=\min\{18+25;8\}=8$
$d_{1,5}^3=\min\{d_{1,3}^2+d_{3,5}^2 d_{1,5}^2\}=\min\{18+37;31\}=31$
$d_{1,6}^3=\min\{d_{1,3}^2+d_{3,6}^2 d_{1,6}^2\}=\min\{18+15;\infty\}=33$
$d_{2,1}^3=\min\{d_{2,3}^2+d_{3,1}^2 d_{2,1}^2\}=\min\{16+26;10\}=10$
$d_{2,2}^3=\min\{d_{2,3}^2+d_{3,2}^2 d_{2,2}^2\}=\min\{16+16;0\}=0$
$d_{2,3}^3=\min\{d_{2,3}^2+d_{3,3}^2 d_{2,3}^2\}=\min\{16+0;16\}=16$
$d_{2,4}^3=\min\{d_{2,3}^2+d_{3,4}^2 d_{2,4}^2\}=\min\{16+25;9\}=9$
$d_{2,5}^3=\min\{d_{2,3}^2+d_{3,5}^2 d_{2,5}^2\}=\min\{16+37;21\}=21$
$d_{2,6}^3=\min\{d_{2,3}^2+d_{3,6}^2 d_{2,6}^2\}=\min\{16+15;\infty\}=31$
$d_{3,1}^3=\min\{d_{3,3}^2+d_{3,1}^2 d_{3,1}^2\}=\min\{0+26;26\}=26$
$d_{3,2}^3=\min\{d_{3,3}^2+d_{3,2}^2 d_{3,2}^2\}=\min\{0+16;16\}=16$
$d_{3,3}^3=\min\{d_{3,3}^2+d_{3,3}^2 d_{3,3}^2\}=\min\{0+0;0\}=0$
$d_{3,4}^3=\min\{d_{3,3}^2+d_{3,4}^2 d_{3,4}^2\}=\min\{0+25;25\}=25$
$d_{3,5}^3=\min\{d_{3,3}^2+d_{3,5}^2 d_{3,5}^2\}=\min\{0+37;37\}=37$
$d_{3,6}^3=\min\{d_{3,3}^2+d_{3,6}^2 d_{3,6}^2\}=\min\{0+15;15\}=15$
$d_{4,1}^3=\min\{d_{4,3}^2+d_{3,1}^2 d_{4,1}^2\}=\min\{25+26;7\}=7$
$d_{4,2}^3=\min\{d_{4,3}^2+d_{3,2}^2 d_{4,2}^2\}=\min\{25+16;9\}=9$
$d_{4,3}^3=\min\{d_{4,3}^2+d_{3,3}^2 d_{4,3}^2\}=\min\{25+0;25\}=25$
$d_{4,4}^3=\min\{d_{4,3}^2+d_{3,4}^2 d_{4,4}^2\}=\min\{25+25;0\}=0$
$d_{4,5}^3=\min\{d_{4,3}^2+d_{3,5}^2 d_{4,5}^2\}=\min\{25+37;30\}=30$
$d_{4,6}^3=\min\{d_{4,3}^2+d_{3,6}^2 d_{4,6}^2\}=\min\{25+15;12\}=12$
$d_{5,1}^3=\min\{d_{5,3}^2+d_{3,1}^2 d_{5,1}^2\}=\min\{\infty+26;\infty\}=\infty$
$d_{5,2}^3=\min\{d_{5,3}^2+d_{3,2}^2 d_{5,2}^2\}=\min\{\infty+16;\infty\}=\infty$
$d_{5,3}^3=\min\{d_{5,3}^2+d_{3,3}^2 d_{5,3}^2\}=\min\{\infty+0;\infty\}=\infty$
$d_{5,4}^3=\min\{d_{5,3}^2+d_{3,4}^2 d_{5,4}^2\}=\min\{\infty+25;\infty\}=\infty$
$d_{5,5}^3=\min\{d_{5,3}^2+d_{3,5}^2 d_{5,5}^2\}=\min\{\infty+37;0\}=0$
$d_{5,6}^3=\min\{d_{5,3}^2+d_{3,6}^2 d_{5,6}^2\}=\min\{\infty+15;23\}=23$
$d_{6,1}^3=\min\{d_{6,3}^2+d_{3,1}^2 d_{6,1}^2\}=\min\{15+26;\infty\}=41$
$d_{6,2}^3=\min\{d_{6,3}^2+d_{3,2}^2 d_{6,2}^2\}=\min\{15+16;\infty\}=31$
$d_{6,3}^3=\min\{d_{6,3}^2+d_{3,3}^2 d_{6,3}^2\}=\min\{15+0;15\}=15$
$d_{6,4}^3=\min\{d_{6,3}^2+d_{3,4}^2 d_{6,4}^2\}=\min\{15+25;\infty\}=40$
$d_{6,5}^3=\min\{d_{6,3}^2+d_{3,5}^2 d_{6,5}^2\}=\min\{15+37;23\}=23$
$d_{6,6}^3=\min\{d_{6,3}^2+d_{3,6}^2 d_{6,6}^2\}=\min\{15+15;0\}=0$
Представим матрицу D^3, включив в нее рассчитанные элементы.

$D^3=$	0	10	18	8	31	33
	10	0	16	9	21	31
	26	16	0	25	37	15
	7	9	25	0	30	12
	∞	∞	∞	∞	0	23
	41	31	15	40	23	0

На основании матрицы D^3, вычислим последовательно все элементы матрицы D^4. Для этого мы используем рекуррентное соотношение $d_{i,j}^4=\min\{ d_{i,4}^3+ d_{4,j}^3; d_{i,j}^3\}$.

$d_{1,1}^4=\min\{d_{1,4}^3+d_{4,1}^3 d_{1,1}^3\}=\min\{8+7;0\}=0$
$d_{1,2}^4=\min\{d_{1,4}^3+d_{4,2}^3 d_{1,2}^3\}=\min\{8+9;10\}=10$
$d_{1,3}^4=\min\{d_{1,4}^3+d_{4,3}^3 d_{1,3}^3\}=\min\{8+25;18\}=18$
$d_{1,4}^4=\min\{d_{1,4}^3+d_{4,4}^3 d_{1,4}^3\}=\min\{8+0;8\}=8$
$d_{1,5}^4=\min\{d_{1,4}^3+d_{4,5}^3 d_{1,5}^3\}=\min\{8+30;31\}=31$
$d_{1,6}^4=\min\{d_{1,4}^3+d_{4,6}^3 d_{1,6}^3\}=\min\{8+12;33\}=20$
$d_{2,1}^4=\min\{d_{2,4}^3+d_{4,1}^3 d_{2,1}^3\}=\min\{9+7;10\}=10$
$d_{2,2}^4=\min\{d_{2,4}^3+d_{4,2}^3 d_{2,2}^3\}=\min\{9+9;0\}=0$
$d_{2,3}^4=\min\{d_{2,4}^3+d_{4,3}^3 d_{2,3}^3\}=\min\{9+25;16\}=16$
$d_{2,4}^4=\min\{d_{2,4}^3+d_{4,4}^3 d_{2,4}^3\}=\min\{9+0;9\}=9$
$d_{2,5}^4=\min\{d_{2,4}^3+d_{4,5}^3 d_{2,5}^3\}=\min\{9+30;21\}=21$
$d_{2,6}^4=\min\{d_{2,4}^3+d_{4,6}^3 d_{2,6}^3\}=\min\{9+12;31\}=21$
$d_{3,1}^4=\min\{d_{3,4}^3+d_{4,1}^3 d_{3,1}^3\}=\min\{25+7;26\}=26$
$d_{3,2}^4=\min\{d_{3,4}^3+d_{4,2}^3 d_{3,2}^3\}=\min\{25+9;16\}=16$
$d_{3,3}^4=\min\{d_{3,4}^3+d_{4,3}^3 d_{3,3}^3\}=\min\{25+25;0\}=0$
$d_{3,4}^4=\min\{d_{3,4}^3+d_{4,4}^3 d_{3,4}^3\}=\min\{25+0;25\}=25$
$d_{3,5}^4=\min\{d_{3,4}^3+d_{4,5}^3 d_{3,5}^3\}=\min\{25+30;37\}=37$
$d_{3,6}^4=\min\{d_{3,4}^3+d_{4,6}^3 d_{3,6}^3\}=\min\{25+12;15\}=15$
$d_{4,1}^4=\min\{d_{4,4}^3+d_{4,1}^3 d_{4,1}^3\}=\min\{0+7;7\}=7$
$d_{4,2}^4=\min\{d_{4,4}^3+d_{4,2}^3 d_{4,2}^3\}=\min\{0+9;9\}=9$
$d_{4,3}^4=\min\{d_{4,4}^3+d_{4,3}^3 d_{4,3}^3\}=\min\{0+25;25\}=25$
$d_{4,4}^4=\min\{d_{4,4}^3+d_{4,4}^3 d_{4,4}^3\}=\min\{0+0;0\}=0$
$d_{4,5}^4=\min\{d_{4,4}^3+d_{4,5}^3 d_{4,5}^3\}=\min\{0+30;30\}=30$
$d_{4,6}^4=\min\{d_{4,4}^3+d_{4,6}^3 d_{4,6}^3\}=\min\{0+12;12\}=12$
$d_{5,1}^4=\min\{d_{5,4}^3+d_{4,1}^3 d_{5,1}^3\}=\min\{\infty+7;\infty\}=\infty$
$d_{5,2}^4=\min\{d_{5,4}^3+d_{4,2}^3 d_{5,2}^3\}=\min\{\infty+9;\infty\}=\infty$
$d_{5,3}^4=\min\{d_{5,4}^3+d_{4,3}^3 d_{5,3}^3\}=\min\{\infty+25;\infty\}=\infty$
$d_{5,4}^4=\min\{d_{5,4}^3+d_{4,4}^3 d_{5,4}^3\}=\min\{\infty+0;\infty\}=\infty$
$d_{5,5}^4=\min\{d_{5,4}^3+d_{4,5}^3 d_{5,5}^3\}=\min\{\infty+30;0\}=0$
$d_{5,6}^4=\min\{d_{5,4}^3+d_{4,6}^3 d_{5,6}^3\}=\min\{\infty+12;23\}=23$
$d_{6,1}^4=\min\{d_{6,4}^3+d_{4,1}^3 d_{6,1}^3\}=\min\{40+7;41\}=41$
$d_{6,2}^4=\min\{d_{6,4}^3+d_{4,2}^3 d_{6,2}^3\}=\min\{40+9;31\}=31$
$d_{6,3}^4=\min\{d_{6,4}^3+d_{4,3}^3 d_{6,3}^3\}=\min\{40+25;15\}=15$
$d_{6,4}^4=\min\{d_{6,4}^3+d_{4,4}^3 d_{6,4}^3\}=\min\{40+0;40\}=40$
$d_{6,5}^4=\min\{d_{6,4}^3+d_{4,5}^3 d_{6,5}^3\}=\min\{40+30;23\}=23$

$d_{6,6}^4=\min\{d_{6,4}^3+d_{4,6}^3 d_{6,6}^3\}=\min\{40+12;0\}=0$

Представим матрицу D^4, включив в нее рассчитанные элементы.

$D^4=$	0	10	18	8	31	20
	10	0	16	9	21	21
	26	16	0	25	37	15
	7	9	25	0	30	12
	∞	∞	∞	∞	0	23
	41	31	15	40	23	0

На основании матрицы D^4, вычислим последовательно все элементы матрицы D^5. Для этого мы используем рекуррентное соотношение $d_{i,j}^5=\min\{ d_{i,5}^4+ d_{5,j}^4;\ d_{i,j}^4\}$.

$d_{1,1}^5=\min\{d_{1,5}^4+d_{5,1}^4 d_{1,1}^4\}=\min\{31+\infty;0\}=0$

$d_{1,2}^5=\min\{d_{1,5}^4+d_{5,2}^4 d_{1,2}^4\}=\min\{31+\infty;10\}=10$

$d_{1,3}^5=\min\{d_{1,5}^4+d_{5,3}^4 d_{1,3}^4\}=\min\{31+\infty;18\}=18$

$d_{1,4}^5=\min\{d_{1,5}^4+d_{5,4}^4 d_{1,4}^4\}=\min\{31+\infty;8\}=8$

$d_{1,5}^5=\min\{d_{1,5}^4+d_{5,5}^4 d_{1,5}^4\}=\min\{31+0;31\}=31$

$d_{1,6}^5=\min\{d_{1,5}^4+d_{5,6}^4 d_{1,6}^4\}=\min\{31+23;20\}=20$

$d_{2,1}^5=\min\{d_{2,5}^4+d_{5,1}^4 d_{2,1}^4\}=\min\{21+\infty;10\}=10$

$d_{2,2}^5=\min\{d_{2,5}^4+d_{5,2}^4 d_{2,2}^4\}=\min\{21+\infty;0\}=0$

$d_{2,3}^5=\min\{d_{2,5}^4+d_{5,3}^4 d_{2,3}^4\}=\min\{21+\infty;16\}=16$

$d_{2,4}^5=\min\{d_{2,5}^4+d_{5,4}^4 d_{2,4}^4\}=\min\{21+\infty;9\}=9$

$d_{2,5}^5=\min\{d_{2,5}^4+d_{5,5}^4 d_{2,5}^4\}=\min\{21+0;21\}=21$

$d_{2,6}^5=\min\{d_{2,5}^4+d_{5,6}^4 d_{2,6}^4\}=\min\{21+23;21\}=21$

$d_{3,1}^5=\min\{d_{3,5}^4+d_{5,1}^4 d_{3,1}^4\}=\min\{37+\infty;26\}=26$

$d_{3,2}^5=\min\{d_{3,5}^4+d_{5,2}^4 d_{3,2}^4\}=\min\{37+\infty;16\}=16$

$d_{3,3}^5=\min\{d_{3,5}^4+d_{5,3}^4 d_{3,3}^4\}=\min\{37+\infty;0\}=0$

$d_{3,4}^5=\min\{d_{3,5}^4+d_{5,4}^4 d_{3,4}^4\}=\min\{37+\infty;25\}=25$

$d_{3,5}^5=\min\{d_{3,5}^4+d_{5,5}^4 d_{3,5}^4\}=\min\{37+0;37\}=37$

$d_{3,6}^5=\min\{d_{3,5}^4+d_{5,6}^4 d_{3,6}^4\}=\min\{37+23;15\}=15$

$d_{4,1}^5=\min\{d_{4,5}^4+d_{5,1}^4 d_{4,1}^4\}=\min\{30+\infty;7\}=7$

$d_{4,2}^5=\min\{d_{4,5}^4+d_{5,2}^4 d_{4,2}^4\}=\min\{30+\infty;9\}=9$

$d_{4,3}^5=\min\{d_{4,5}^4+d_{5,3}^4 d_{4,3}^4\}=\min\{30+\infty;25\}=25$

$d_{4,4}^5=\min\{d_{4,5}^4+d_{5,4}^4 d_{4,4}^4\}=\min\{30+\infty;0\}=0$

$d_{4,5}^5=\min\{d_{4,5}^4+d_{5,5}^4 d_{4,5}^4\}=\min\{30+0;30\}=30$

$d_{4,6}^5=\min\{d_{4,5}^4+d_{5,6}^4 d_{4,6}^4\}=\min\{30+23;12\}=12$

$d_{5,1}^5=\min\{d_{5,5}^4+d_{5,1}^4 d_{5,1}^4\}=\min\{0+\infty;\infty\}=\infty$

$d_{5,2}^5=\min\{d_{5,5}^4+d_{5,2}^4 d_{5,2}^4\}=\min\{0+\infty;\infty\}=\infty$

$d_{5,3}^5=\min\{d_{5,5}^4+d_{5,3}^4 d_{5,3}^4\}=\min\{0+\infty;\infty\}=\infty$

$d_{5,4}^5=\min\{d_{5,5}^4+d_{5,4}^4 d_{5,4}^4\}=\min\{0+\infty;\infty\}=\infty$

$d_{5,5}^5=\min\{d_{5,5}^4+d_{5,5}^4 d_{5,5}^4\}=\min\{0+0;0\}=0$

$d_{5,6}^5=\min\{d_{5,5}^4+d_{5,6}^4 d_{5,6}^4\}=\min\{0+23;23\}=23$

$d_{6,1}^5=\min\{d_{6,5}^4+d_{5,1}^4 d_{6,1}^4\}=\min\{23+\infty;41\}=41$

$d_{6,2}^5=\min\{d_{6,5}^4+d_{5,2}^4 d_{6,2}^4\}=\min\{23+\infty;31\}=31$

$d_{6,3}^5=\min\{d_{6,5}^4+d_{5,3}^4 d_{6,3}^4\}=\min\{23+\infty;15\}=15$
$d_{6,4}^5=\min\{d_{6,5}^4+d_{5,4}^4 d_{6,4}^4\}=\min\{23+\infty;40\}=40$
$d_{6,5}^5=\min\{d_{6,5}^4+d_{5,5}^4 d_{6,5}^4\}=\min\{23+0;23\}=23$
$d_{6,6}^5=\min\{d_{6,5}^4+d_{5,6}^4 d_{6,6}^4\}=\min\{23+23;0\}=0$
Представим матрицу D^5, включив в нее рассчитанные элементы.

$D^5=$	0	10	18	8	31	20
	10	0	16	9	21	21
	26	16	0	25	37	15
	7	9	25	0	30	12
	∞	∞	∞	∞	0	23
	41	31	15	40	23	0

На основании матрицы D^5, вычислим последовательно все элементы матрицы D^6. Для этого мы используем рекуррентное соотношение $d_{i,j}^6=\min\{ d_{i,6}^5+ d_{6,j}^5; d_{i,j}^5\}$.
$d_{1,1}^6=\min\{d_{1,6}^5+d_{6,1}^5 d_{1,1}^5\}=\min\{20+41;0\}=0$
$d_{1,2}^6=\min\{d_{1,6}^5+d_{6,2}^5 d_{1,2}^5\}=\min\{20+31;10\}=10$
$d_{1,3}^6=\min\{d_{1,6}^5+d_{6,3}^5 d_{1,3}^5\}=\min\{20+15;18\}=18$
$d_{1,4}^6=\min\{d_{1,6}^5+d_{6,4}^5 d_{1,4}^5\}=\min\{20+40;8\}=8$
$d_{1,5}^6=\min\{d_{1,6}^5+d_{6,5}^5 d_{1,5}^5\}=\min\{20+23;31\}=31$
$d_{1,6}^6=\min\{d_{1,6}^5+d_{6,6}^5 d_{1,6}^5\}=\min\{20+0;20\}=20$
$d_{2,1}^6=\min\{d_{2,6}^5+d_{6,1}^5 d_{2,1}^5\}=\min\{21+41;10\}=10$
$d_{2,2}^6=\min\{d_{2,6}^5+d_{6,2}^5 d_{2,2}^5\}=\min\{21+31;0\}=0$
$d_{2,3}^6=\min\{d_{2,6}^5+d_{6,3}^5 d_{2,3}^5\}=\min\{21+15;16\}=16$
$d_{2,4}^6=\min\{d_{2,6}^5+d_{6,4}^5 d_{2,4}^5\}=\min\{21+40;9\}=9$
$d_{2,5}^6=\min\{d_{2,6}^5+d_{6,5}^5 d_{2,5}^5\}=\min\{21+23;21\}=21$
$d_{2,6}^6=\min\{d_{2,6}^5+d_{6,6}^5 d_{2,6}^5\}=\min\{21+0;21\}=21$
$d_{3,1}^6=\min\{d_{3,6}^5+d_{6,1}^5 d_{3,1}^5\}=\min\{15+41;26\}=26$
$d_{3,2}^6=\min\{d_{3,6}^5+d_{6,2}^5 d_{3,2}^5\}=\min\{15+31;16\}=16$
$d_{3,3}^6=\min\{d_{3,6}^5+d_{6,3}^5 d_{3,3}^5\}=\min\{15+15;0\}=0$
$d_{3,4}^6=\min\{d_{3,6}^5+d_{6,4}^5 d_{3,4}^5\}=\min\{15+40;25\}=25$
$d_{3,5}^6=\min\{d_{3,6}^5+d_{6,5}^5 d_{3,5}^5\}=\min\{15+23;37\}=37$
$d_{3,6}^6=\min\{d_{3,6}^5+d_{6,6}^5 d_{3,6}^5\}=\min\{15+0;15\}=15$
$d_{4,1}^6=\min\{d_{4,6}^5+d_{6,1}^5 d_{4,1}^5\}=\min\{12+41;7\}=7$
$d_{4,2}^6=\min\{d_{4,6}^5+d_{6,2}^5 d_{4,2}^5\}=\min\{12+31;9\}=9$
$d_{4,3}^6=\min\{d_{4,6}^5+d_{6,3}^5 d_{4,3}^5\}=\min\{12+15;25\}=25$
$d_{4,4}^6=\min\{d_{4,6}^5+d_{6,4}^5 d_{4,4}^5\}=\min\{12+40;0\}=0$
$d_{4,5}^6=\min\{d_{4,6}^5+d_{6,5}^5 d_{4,5}^5\}=\min\{12+23;30\}=30$
$d_{4,6}^6=\min\{d_{4,6}^5+d_{6,6}^5 d_{4,6}^5\}=\min\{12+0;12\}=12$
$d_{5,1}^6=\min\{d_{5,6}^5+d_{6,1}^5 d_{5,1}^5\}=\min\{23+41;\infty\}=64$
$d_{5,2}^6=\min\{d_{5,6}^5+d_{6,2}^5 d_{5,2}^5\}=\min\{23+31;\infty\}=54$
$d_{5,3}^6=\min\{d_{5,6}^5+d_{6,3}^5 d_{5,3}^5\}=\min\{23+15;\infty\}=38$
$d_{5,4}^6=\min\{d_{5,6}^5+d_{6,4}^5 d_{5,4}^5\}=\min\{23+40;\infty\}=63$
$d_{5,5}^6=\min\{d_{5,6}^5+d_{6,5}^5 d_{5,5}^5\}=\min\{23+23;0\}=0$

$d_{5,6}^{6}=\min\{d_{5,6}^{5}+d_{6,6}^{5}d_{5,6}^{5}\}=\min\{23+0;23\}=23$
$d_{6,1}^{6}=\min\{d_{6,6}^{5}+d_{6,1}^{5}d_{6,1}^{5}\}=\min\{0+41;41\}=41$
$d_{6,2}^{6}=\min\{d_{6,6}^{5}+d_{6,2}^{5}d_{6,2}^{5}\}=\min\{0+31;31\}=31$
$d_{6,3}^{6}=\min\{d_{6,6}^{5}+d_{6,3}^{5}d_{6,3}^{5}\}=\min\{0+15;15\}=15$
$d_{6,4}^{6}=\min\{d_{6,6}^{5}+d_{6,4}^{5}d_{6,4}^{5}\}=\min\{0+40;40\}=40$
$d_{6,5}^{6}=\min\{d_{6,6}^{5}+d_{6,5}^{5}d_{6,5}^{5}\}=\min\{0+23;23\}=23$
$d_{6,6}^{6}=\min\{d_{6,6}^{5}+d_{6,6}^{5}d_{6,6}^{5}\}=\min\{0+0;0\}=0$
Представим матрицу D^6, включив в нее рассчитанные элементы.

$D^6=$	0	10	18	8	31	20
	10	0	16	9	21	21
	26	16	0	25	37	15
	7	9	25	0	30	12
	64	54	38	63	0	23
	41	31	15	40	23	0

В результате, нами получена матрица длин кратчайших путей между каждой парой вершин графа. Ниже представлена **таблица путей**. Каждый элемент Cij таблицы, это путь из вершины i в вершину j:

Таблица путей						
i \ j	**1**	**2**	**3**	**4**	**5**	**6**
1	-	d_{1-2}=1-2	d_{1-3}=1-3	d_{1-4}=1-4	d_{1-5}=1-2-5	d_{1-6}=1-4-6
2	d_{2-1}=2-1	-	d_{2-3}=2-3	d_{2-4}=2-4	d_{2-5}=2-5	d_{2-6}=2-4-6
3	d_{3-1}=3-2-1	d_{3-2}=3-2	-	d_{3-4}=3-2-4	d_{3-5}=3-2-5	d_{3-6}=3-6
4	d_{4-1}=4-1	d_{4-2}=4-2	d_{4-3}=4-1-3	-	d_{4-5}=4-2-5	d_{4-6}=4-6
5	d_{5-1}=5-6-3-2-1	d_{5-2}=5-6-3-2	d_{5-3}=5-6-3	d_{5-4}=5-6-3-2-4	-	d_{5-6}=5-6
6	d_{6-1}=6-3-2-1	d_{6-2}=6-3-2	d_{6-3}=6-3	d_{6-4}=6-3-2-4	d_{6-5}=6-5	-

3 Задания для самостоятельной работы

1.. Методом Флойда определить кратчайшие пути между всеми парами вершин в нагруженном орграфе, заданном матрицей длин дуг.

$$\begin{bmatrix} 0 & 7 & 2 & 3 & - \\ - & 0 & - & 2 & - \\ 3 & - & 0 & 5 & 4 \\ - & - & 2 & 0 & 1 \\ - & 3 & 4 & - & 0 \end{bmatrix}$$

2. Методом Флойда определить кратчайшие пути между всеми парами вершин в нагруженном орграфе, заданном матрицей длин дуг.

$$\begin{bmatrix} 0 & 4 & - & - & 5 & 1 & - \\ 3 & 0 & 2 & 1 & - & - & - \\ 1 & 1 & 0 & - & - & - & 3 \\ - & 3 & 1 & 0 & 1 & - & - \\ - & - & 2 & - & 0 & 1 & 5 \\ - & 3 & - & 2 & 2 & 0 & - \\ - & - & 2 & - & - & 2 & 0 \end{bmatrix}$$

3. Методом Флойда определить кратчайшие пути между всеми парами вершин в нагруженном орграфе, заданном матрицей длин дуг.

$$\begin{bmatrix} 0 & 5 & - & 3 & 4 \\ 5 & 0 & 2 & - & 3 \\ - & 2 & 0 & 4 & 1 \\ 3 & - & 4 & 0 & 2 \\ 4 & 3 & 1 & 2 & 0 \end{bmatrix}$$

Тема 3.2 Системы массового обслуживания

Занятие 5 Одноканальная СМО с отказами

1 Теоретическая часть

В качестве показателей эффективности СМО с отказами будем рассматривать:

A — абсолютную пропускную способность СМО, т.е. среднее число заявок, обслуживаемых в единицу времени;

Q — относительную пропускную способность, т.е. среднюю долю пришедших заявок, обслуживаемых системой;

$P_{отк}$ — *вероятность отказа,* т.е. того, что заявка покинет СМО необслуженной;

к — среднее число занятых каналов (для многоканальной системы).

Одноканальная система с отказами. Рассмотрим задачу.

Имеется один канал, на который поступает поток заявок с интенсивностью *X*. Поток обслуживании имеет интенсивность ц[1]. Найти предельные вероятности состояний системы и показатели ее эффективности.

Система *S* (СМО) имеет два состояния: S_0 — канал свободен, S_1— канал занят. Размеченный граф состояний представлен на рис. 6.

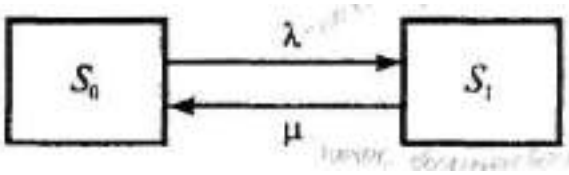


Рис. 15.6

В предельном, стационарном режиме система алгебраических уравнений для вероятностей состояний имеет вид

$$\begin{cases} \lambda p_0 = \mu p_1, \\ \mu p_1 = \lambda p_0, \end{cases} \qquad (18)$$

т.е. система вырождается в одно уравнение. Учитывая нормировочное условие $p_o + p_1 = 1$ найдем из (18) предельные вероятности состояний

$$p_0 = \frac{\mu}{\lambda + \mu}, \; p_1 = \frac{\lambda}{\lambda + \mu}, \qquad (19)$$

которые выражают среднее относительное время пребывания сие темы в состоянии S_0 (когда канал свободен) и S_1 (когда канал занят), т.е. определяют соответственно относительную пропускную способность *Q* системы и вероятность отказа $P_{тк}$:

$$Q = \frac{\mu}{\lambda + \mu}, \qquad (20)$$

$$P_{\text{отк.}} = \frac{\lambda}{\lambda + \mu}.$$ (21) Абсолютную пропускную способность найдем, умножив относительную пропускную способность Q на интенсивность потока отказов

$$A = \frac{\lambda\mu}{\lambda + \mu}.$$ (22)

2 Практическая часть

Задача 1Известно, что заявки на телефонные переговоры в телевизионном ателье поступают с интенсивностью λ, равной 90 заявок в час, а средняя продолжительность разговора по телефону $t_{об.}$=2 мин. Определить показатели эффективности работы СМО (телефонной связи) при наличии одного телефонного номера.

Решение. Имеем λ=90 (1/ч), $t_{ОБС}$=2 мин. Интенсивность потока обслуживании $\mu=1/t_{обс}$==1/2=0,5 (1/мин)=30 (1/ч). По (20) относительная пропускная способность СМО Q=30/(90+30)=0,25, т.е. в среднем только 25% поступающих заявок осуществят переговоры по телефону. Соответственно вероятность отказа в обслуживании составит $Р_{отк}$=0,75 (см. (21)). Абсолютная пропускная способность СМО по (20) А=90*0,25=22,5, т.е. в среднем в час будут обслужены 22,5 заявки на переговоры. Очевидно, что при наличии только одного телефонного номера СМО будет плохо справляться с потоком заявок.

3 Задания для самостоятельной работы

1. Рассматривается круглосуточная работа пункта проведения профилактического осмотра автомашин с одним каналом (одной группой проведения осмотра). На осмотр и выявление дефектов каждой машины затрачивается в среднем 0,5 ч. На осмотр поступает в среднем 36 машин в сутки. Потоки заявок и обслуживании — простейшие. Если машина, прибывшая в пункт осмотра, не застает ни одного канала свободным, она покидает пункт осмотра необслуженной. Определить вероятности состояний и характеристики обслуживания профилактического пункта осмотра.

2. Анализируется работа междугородного переговорного пункта в небольшом городке. Пункт имеет один телефонный аппарат для переговоров. В среднем за сутки поступает 240 заявок на переговоры. Средняя длительность переговоров (с учетом вызова абонентов в другом городе) составляет 5 мин. Никаких ограничений на длину очереди нет. Потоки заявок и обслуживания простейшие. Определить предельные вероятности состояний и характеристики обслуживания переговорного пункта в стационарном режиме.

Занятие 6 Одноканальная СМО с ожиданиями

1 Теоретическая часть

СМО с ограниченным временем ожидания. На практике часто встречаются СМО с так называемыми "нетерпеливыми" заявками. Такие заявки могут уйти из очереди, если время ожидания превышает некоторую величину. В частности, такого рода заявки возникают в различных технологических системах, в которых задержка с началом обслуживания может привести к потере качества продукции, в системах оперативного управления, когда срочные сообщения теряют ценность (или даже смысл), если они не поступают на обслуживание в течение определенного времени.

2 Практическая часть

Задание 1. 2 В порту имеется один причал для разгрузки судов. Интенсивность потока судов равна 0,4 (судов в сутки). Среднее время разгрузки одного судна составляет 2 суток. Предполагается, что очередь может быть неограниченной длины. Найти показатели эффективности работы причала, а также вероятность того, что ожидают разгрузки не более чем 2 судна.

Решение. Имеем $\rho = \lambda/\mu = \lambda t_{обс} = 0{,}4*2 = 0{,}8$. Так как $\rho = 0{,}8 < 1$, то очередь на разгрузку не может бесконечно возрастать и предельные вероятности существуют. Найдем их.

Вероятность того, что причал свободен, по (33) p_o=1—0,8=0,2, а вероятность того, что он занят, $P_{зан}$=1—0,2=0,8. По формуле (34) вероятности того, что у причала находятся 1, 2, 3 судна (т.е. ожидают разгрузки 0, 1, 2 судна), равны $p_1 = 0{,}8(1-0{,}8) = 0{,}16$; $p_2 = 0{,}8^2*(1-0{,}8) = 0{,}128$; $p_3 = 0{,}8^3(1-0{,}8) = 0{,}1024$.

Вероятность того, что ожидают разгрузку не более чем 2 судна, равна По формуле (40) среднее число судов, ожидающих разгрузки,

$$L_{оч.} = 0{,}8^2/(1-0{,}8) = 3{,}2,$$

а среднее время ожидания разгрузки по формуле

$$T_{оч.} = \frac{1}{\lambda} L_{оч.}.$$

Среднее число судов, находящихся у причала, $L_{сист.} = 0{,}8/(1-0{,}8) = 4$ (сутки) ($L_{сист} = 3{,}2+0{,}8 = 4$ (сутки), а среднее время пребывания судна у причала по формуле (41) $L_{сист} = 4/0{,}8 = 5$ (сутки).

Очевидно, что эффективность разгрузки судов невысокая. Для ее повышения необходимо уменьшение среднего времени, разгрузки судна $t_{обс}$ либо увеличение числа причалов n.

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 $au + bv = d$.

3 Задания для самостоятельной работы

1. В торговом павильоне покупателей обслуживает один продавец. Площадь павильона составляет 24 м^2, причем 10 м^2 приходится на торговый зал, вместимость которого ограничена. Поэтому если очередь на обслуживание составляет 10 человек, то потенциальный покупатель туда не входит, что свидетельствует об отказе в обслуживании и как следствие, снижении товарооборота и ухудшению "показателей работы коммерческого предприятия. Дать оценку системы массового обслуживания и определить рекомендации по созданию оптимального режима работы, если интенсивность прихода покупателей составляет 120 человек в час. а среднее время обслуживания одного покупателя 3 мин.
2. Мини-маркет с одним контролером-кассиром обслуживает покупателей, входящий поток которых подчиняется закону Пуассона с параметром 20 покупателей в час. Время обслуживания подчиняется показательному закону с параметром 25 покупателей в час. Определить: вероятность простоя контролера-кассира, среднюю длину очереди. среднее число покупателей в мини-маркете. среднее время ожидания обслуживания, среднее время пребывания покупателей в мини-маркете и дать оценку работы системы.

Занятие 7 Многоканальная СМО с ожиданиями

1 Теоретическая часть

Рассмотрим задачу. Имеется n-канальная СМО с неограниченной очередью. Поток заявок, поступающих в СМО, имеет интенсивностьλ, а поток обслуживании — интенсивность μ. Необходимо найти предельные вероятности состояний СМО и показатели ее эффективности.

Система может находиться в одном из состояний $S_0, S_1, S_2, ..., S_k ..., S_n, ...$, нумеруемых по числу заявок, находящихся в СМО: S_o — в системе нет заявок (все каналы свободны); S_1— занят один канал, остальные свободны; S_2 — заняты два канала, остальные свободны; ..., S_k — занято k каналов, остальные свободны; ..., S_n — заняты все n каналов (очереди нет); S_{n+1} — заняты все n каналов, в очереди одна заявка; ..., S_{n+r} — заняты все n каналов, r заявок стоит в очереди,....

Граф состояний системы показан на рис. 9. Обратим внимание на то, что в отличие от предыдущей СМО, интенсивность потока обслуживании (переводящего систему из одного состояния в другое справа налево) не остается постоянной, а по мере увеличения числа заявок в СМО от 0 до n увеличивается от величины μ до nμ, так как соответственно увеличивается число каналов обслуживания. При числе заявок в СМО большем, чем я, интенсивность потока обслуживании сохраняется равной $n\mu$

Можно показать, что при p/n < 1 предельные вероятности существуют.

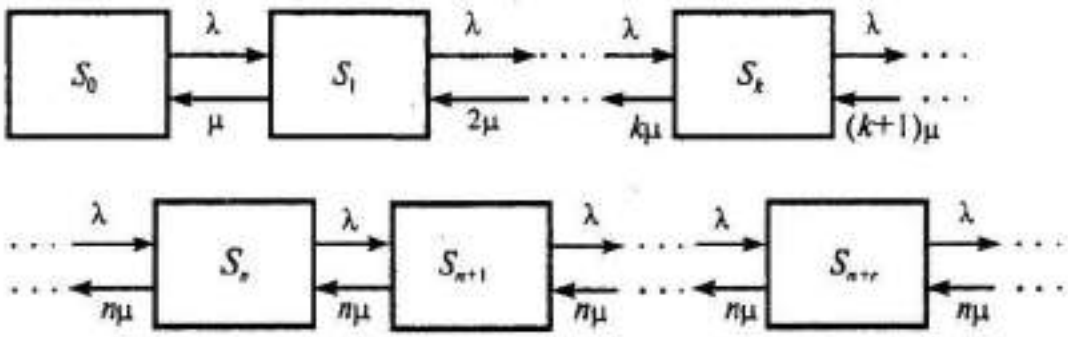


Рис. 9

Если p/n≥1, очередь растет до бесконечности. Используя формулы (16) и (17) для процесса гибели и размножения, можно получить следующие формулы для предельных вероятностей состояний n-канальной СМО с неограниченной очередью

$$p_0 = \left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \ldots + \frac{\rho^n}{n!} + \frac{\rho^{n+1}}{n!(n-\rho)}\right)^{-1}, \qquad \textbf{(45)}$$

$$p_1 = \frac{\rho}{1!}p_0, \ldots, p_k = \frac{\rho^k}{k!}p_0, \ldots, p_n = \frac{\rho^n}{n!}\rho_0, \qquad \textbf{(46)}$$

$$p_{n+1} = \frac{\rho^{n+1}}{n \cdot n!}p_0, \ldots, p_{n+r} = \frac{\rho^{n+r}}{n^r \cdot n!}p_0, \ldots . \qquad \textbf{(47)}$$

Вероятность того, что заявка окажется в очереди,

$$P_{оч.} = \frac{\rho^{n+1}}{n!(n-\rho)}p_0 . \qquad \textbf{(48)}$$

Для n-канальной СМО с неограниченной очередью, используя прежние приемы, можно найти: среднее число занятых каналов

$$\bar{k} = \frac{\lambda}{\mu} = \rho, \qquad \textbf{(49)}$$

среднее число заявок в очереди

$$L_{оч.} = \frac{\rho^{n+1} p_0}{n \cdot n! \left(1 - \frac{\rho}{n}\right)^2}, \qquad \textbf{(50)}$$

среднее число заявок в системе

$$L_{сист.} = L_{оч.} + \rho.$$

2 Практическая часть

Задача 1В универсаме к узлу расчета поступает поток покупателей с интенсивностью λ=81чел. в час. Средняя продолжительность обслуживания контролером-кассиром одного покупателя $t_{об.}$=2 мин. Определить:

а. Минимальное количество контролеров-кассиров n_{min}, при котором очередь не будет расти до бесконечности, и соответствующие характеристики обслуживания при $n=n_{min}$

б. Оптимальное количество $n_{опт}$ контролеров-кассиров, при котором относительная величина затрат $C_{оти}$, связанная с издержками на содержание каналов обслуживания и с пребыванием в очереди покупателей, задаваемая, например, как $C_{оти} = \frac{1}{\lambda} n + 3t_{оч.}$ будет минимальна, и сравнить характеристики обслуживания при $n=n_{min}$ и $n=n_{опт}$.

в. Вероятность того, что в очереди будет не более трех покупателей.

Решение. а. По условию λ=81(1/ч)=81/60=1,35(1/мин.). По формуле (24) $\rho = \lambda/\mu = \lambda t_{об}$=1,35*2 =2,7. Очередь не будет возрастать до бесконечности при условии $\rho/n < 1$, т.е. при $n > \rho = 2{,}7$. Таким образом, минимальное количество контролеров-кассиров $n_{min} = 3$.

Найдем характеристики обслуживания СМО при $n = 3$.

Вероятность того, что в узле расчета отсутствуют покупатели, по формуле (45) $P_0=(1+2{,}7+2{,}7^2/2!+2{,}7^3/3!+2{,}7^4/3!(3-2{,}7)^{-1} = 0{,}025$, т.е. в среднем 2,5% времени контролеры-кассиры будут простаивать.

Вероятность того, что в узле расчета будет очередь, по (48)

$$P_{оч.} = (2{,}7^4/3!(3-2{,}7))0{,}025 = 0{,}735.$$

Среднее число покупателей, находящихся в очереди, по (50)

$$L_{оч} = (2{,}7^4/3 \cdot 3!(1-2{,}7/3)^2)0{,}025 = 7{,}35.$$

Среднее время ожидания в очереди по (15.42)

$$T_{оч.} = 7{,}35/1{,}35 = 5{,}44 \text{ (мин)}.$$

Среднее число покупателей в узле расчета по (51)

$$L_{\text{сист.}} = 7{,}35+2{,}7 = 10{,}05.$$

Среднее время нахождения покупателей в узле расчета по (41)

$$T_{\text{сист.}} = 10{,}05/1{,}35 = 7{,}44 \text{ (мин)}.$$

Среднее число контролеров-кассиров, занятых обслуживанием покупателей, по (49) $к$ = 2,7.

Коэффициент (доля) занятых обслуживанием контролеров-кассиров

$$\bar{k} = \rho/n = 2{,}7/3 = 0{,}9.$$

Абсолютная пропускная способность узла расчета A = 1,35 (1/мин), или 81 (1/ч), т.е. 81 покупатель в час.

Анализ характеристик обслуживания свидетельствует о значительной перегрузке узла расчета при наличии трех контролеров-кассиров.

б. Относительная величина затрат при n = 3

$$C_{\text{отн.}} = \frac{1}{\lambda} n + 3T_{\text{оч.}} = 3/1{,}35+3\cdot 5{,}44 = 18{,}54.$$

Рассчитаем относительную величину затрат при других значениях n (табл. 2).

Таблица 2

Характеристика обслуживания	Число контролеров-кассиров				
	3	4	5	6	7
Вероятность простоя контролеров-кассиров p_0	0,025	0,057	0,065	0,067	0,067
Среднее число покупателей в очереди $T_{оч.}$	5,44	0,60	0,15	0,03	0,01
Относительная величина затрат $C_{отн.}$	18,54	4,77	4,14	4,53	5,22

Как видно из табл. 2, минимальные затраты получены при n = $n_{опт}$=5 контролерах-кассирах.

Определим характеристики обслуживания узла расчета при n= $n_{опт}$=5. Получим $P_{оч}$ = 0.091; $L_{оч}$= 0,198; $T_{оч}$.=0,146 (мин);$L_{сист.}$ = 2,90; $T_{сист.}$ = 2,15 (мин); $к$ = 2,7; $k_З$ = 0,54.

Как видим, при n =5 по сравнению с n= 3 существенно уменьшились вероятность возникновения очереди $P_{оч}$, длина очереди $L_{оч}$ и среднее время пребывания в очереди $T_{оч}$ и соответственно среднее число покупателей $L_{сист}$ и среднее время нахождения в узле расчета $T_{сист.}$, а также доля занятых обслуживанием контролеров $k_З$ Но среднее число занятых обслуживанием контролеров-кассиров $к$ и абсолютная пропускная способность узла расчета A естественно не изменились.

в. Вероятность того, что в очереди будет не более 3 покупателей, определится как

$$P(r \le 3) = \underbrace{p_1+p_2+p_3+p_4+p_5}_{\text{(когда заняты от 1 до 5 контролеров-кассиров)}} + \underbrace{p_{5+1}+p_{5+2}+p_{5+3}}_{\text{(когда в очереди стоят от 1 до 3 покупателей)}} =$$

$= 1-P_{оч.}+p_{5+1}+p_{5+2}+p_{5+3}$, где каждое слагаемое найдем по формулам (15.45)—(15.48). Получим при $n = 5$:

$$P(r\le 3) = 1-\frac{2,7^6}{5!(5-2,3)}0,065+\frac{2,7^6}{5\cdot 5!}0,065+$$

$$+\frac{2,7^7}{5^2\cdot 5!}0,065+\frac{2,7^8}{5^3\cdot 5!}0,065=0,986.$$

(Заметим, что в случае n=3 контролеров-кассиров та же вероятность существенно меньше: $P(r\le 3)=0,464$)

Задача 2. Железнодорожная касса с двумя окошками продает билеты в два пункта *A* и *B*. Интенсивность потока пассажиров, желающих купить билеты, для обоих пунктов одинакова: $\lambda_A = \lambda_B = 0,45$ (пассажиров в минуту). На обслуживание пассажиров кассир тратит в среднем 2 мин. Рассматриваются два варианта продажи билетов: первый — билеты продаются в одной кассе с двумя окошками одновременно в оба пункта *A* и *B* второй — билеты продаются в двух специализированных кассах (по одному окошку в каждой), одна только в пункт *A,* другая — только в пункт *B*. Необходимо:

а. Сравнить два варианта продажи билетов по основным характеристикам обслуживания.

б. Определить, как надо изменить среднее время обслуживания одного пассажира, чтобы по второму варианту продажи пассажиры затрачивали на приобретение билетов в среднем меньше времени, чем по первому варианту.

Решение, а**. По первому варианту** имеем двухканальную СМО, на которую поступает поток заявок интенсивностью $\lambda = 0,45+0,45 = 0,9$; интенсивность потока обслуживании $\mu = 1/2 = 0,5$; $p/n = 1.8/2 = 1,8$. Так как $p/n = 1,8/2 = 0,9 < 1$, то предельные вероятности существуют.

Вероятность простоя двух кассиров по (45)

$$p_0 = \left(1+\frac{1,8}{1!}+\frac{1,8^2}{2!}+\frac{1,8^3}{2!(2-1,8)}\right)^{-1} = 0,0526.$$

Среднее число пассажиров в очереди по (15.50)

$$L_{оч.} = 1,8^3/2\cdot 2!(1-1,8/2)\cdot 0,0526 = 7,67.$$

Среднее число пассажиров у кассы по (15.51)

$$L_{сист.} = 7,67+1,8 = 9,47.$$

Среднее время на ожидание в очереди и покупку билетов равно соответственно (по формулам (42) и (41)): $T_{оч.}$ = 7,67/0,9= 8,52 (мин) и $T_{сист.}$ = 9,47/0,9 =10,5 (мин).

По **второму варианту** имеем две одноканальные СМО (два специализированных окошка); на каждую поступает поток заявок с интенсивностью λ= 0,45. По-прежнему μ= 0,5; ρ = λ/μ = 0,9 < 1, предельные вероятности существуют. По формулам (40), (36), (42), (41)

$$L_{оч.} = 0{,}9^2/(1-0{,}9) = 8{,}1;\ L_{сист.} = 0{,}9/(1-0{,}9) = 9{,}0;$$

$$T_{оч.} = 8{,}1/0{,}45 = 18{,}0\ (мин),\ T_{сист.} = 9{,}0/0{,}45 = 20{,}0\ (мин).$$

Итак, по второму варианту увеличились и длина очереди, и среднее время ожидания в ней и в целом на покупку билетов. Такое различие объясняется тем, что в первом варианте (двухканальная СМО) меньше средняя доля времени, которую простаивает каждый из двух кассиров: если он не занят обслуживанием пассажира, покупающего билет в пункт *A,* он может заняться обслуживанием пассажира, покупающего билет в пункт *B,* и наоборот. Во втором варианте такой взаимозаменяемости нет.

Можно заметить, что среднее время на покупку билетов по второму варианту увеличилось более чем в 2 раза. Такое значительное увеличение связано с тем, что СМО работает на пределе своих возможностей (ρ = 0,9): достаточно незначительно увеличить среднее время обслуживания $T_{об}$, т.е. уменьшить μ, и ρ превзойдет 1, т.е. очередь начнет неограниченно возрастать.

б. Выше было получено, что по первому варианту продажи билетов при среднем времени обслуживания одного пассажира $t_{об.}$ = 2 (мин) среднее время на покупку билетов составит $T_{сист1}$ = 10,5 (мин). По условию для второго варианта продажи $T_{сист2} < T_{сист1}$ или с учетом (36) и (41): $\frac{1}{\lambda}\frac{\rho}{1-\rho} < T_{сист1.}$ Полагая $\rho=\lambda/\mu=\lambda_{tоб.}$ Получим $-\frac{t_{об.}}{1-\lambda t_{об.}} < T_{сист1,}$ откуда найдем

$$t_{об.} < \frac{T_{сист1}}{1+\lambda T_{сист1}} \text{ или } t_{об.} < \frac{10{,}5}{1+0{,}45*10{,}5} = 1{,}83((мин))$$

Итак, средние затраты времени на покупку билетов по второму варианту продажи уменьшатся, если среднее время обслуживания одного пассажира уменьшится более чем на 0,17 мин, или более чем на 8,5%

Тема 3.3 Имитационное моделирование
Занятие 8 Применение метода Монте-Карло
1 Теоретическая часть

Во многих задачах исходные данные носят случайный характер, поэтому для их решения должен применяться статистико-вероятностный подход. На основе таких подходов построен ряд численных методов, которые учитывают случайный характер вычисляемых или измеряемых величин. К ним принадлежит метод статистических испытаний, называемый также методом Монте -Карло, который применяется к решению некоторых задач вычислительной математики, в том числе вычисления интегралов.

Рассмотрим идею применения метода Монте-Карло на примере вычисления определённого интеграла от функции, зависящей от одной переменной. Пусть необходимо вычислить интеграл от некоторой заданной функции *f(x)* на интервале [*a*,*b*]. Рассмотрим прямоугольник высотой *H* и длинной (*b*-*a*), такой, что функция *f(x)* целиком лежит внутри данного прямоугольника (рис 3.1).

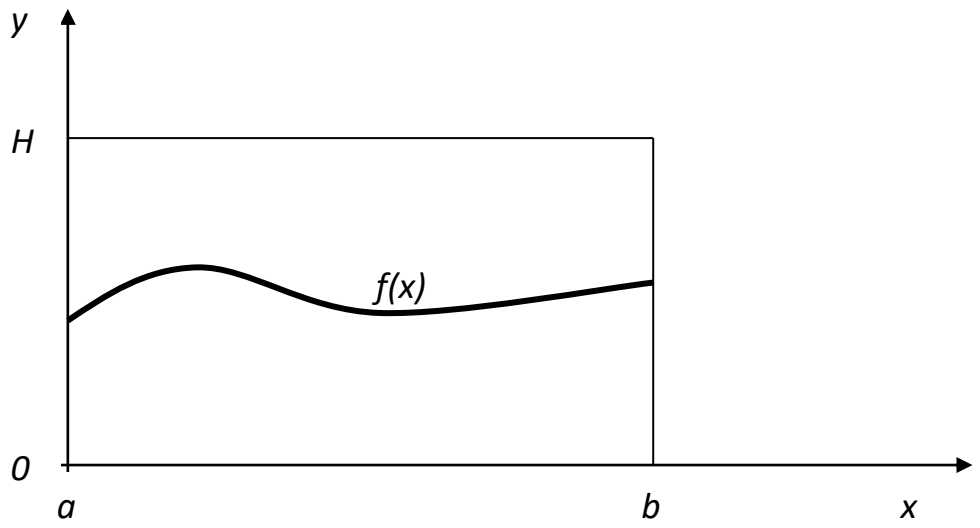


Рис. 3.1 График функции *f(x)*.

Сгенерируем *N* пар случайных чисел (x_i, y_i), $i=1,2,\ldots,N$, равномерно распределённых в данном прямоугольнике: $a \le x_i \le b$, $0 \le y_i \le H$. Тогда доля точек (x_i, y_i), удовлетворяющих условию $y_i \le f(x_i)$, является оценкой отношения интеграла от функции *f(x)* к площади *S* рассматриваемого прямоугольника. Следовательно, оценка интеграла в данном методе может быть получена по формуле:

$$I_N = S\frac{n_S}{N}, \qquad (3.1)$$

где

S - площадь прямоугольника;

n_S - количество точек, удовлетворяющих условию $y_i \le f(x_i)$;

N - полное количество точек.

Можно использовать другой способ вычисления определённого интеграла, рассматривая его как среднее значение функции $f(x)$ на отрезке $[a,b]$:

$$I_N = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad (3.2)$$

где

x_i - последовательность случайных чисел с равномерным законом распределения на отрезке $[a,b]$.

Метод Монте-Карло также может быть использован и для вычисления кратных интегралов.

2 Практическая часть

Задача: Методом Монте-Карло вычислить интеграл

$$I = \iint\limits_G (x^2 + y^2)dxdy$$

где область G определятся следующими неравенствами

1/2≤x≤1 0≤y≤1

Решение

Область интегрирования принадлежит единичному квадрату 0≤x≤1 0≤y≤1

Для вычисления интеграла воспользуемся таблицей случайных чисел при этом каждые два последовательных числа из этой таблицы примем за координаты случайной точки M(x,y)

Табл. Случайные числа, равномерно распределенные на отрезке [0.1]

0,57705	0,35483	0,11578	0,65339
0,71618	0,09393	0,93045	0,93382
0,73710	0,30304	0,93011	0,05758
0,70131	0,55186	0,42844	0,00336
0,16961	0,64003	0,52906	0,88222
0,53324	0,20514	0,09461	0,98585
0,43166	0,00188	0,99602	0,52103
0,26275	0,55709	0,69962	0,91827
0,05926	0,86977	0,31311	0,07069
0,66289	0,31303	0,27004	0,13928

Записываем координаты x и y случайных точек в таблицу 2 округляя до трех знаков после запятой, и выбираем те из них, которые принадлежат области интегрирования.

П о р я д о к з а п о л н е н и я таблицы 2

1) Среди всех значений x выделяем те, которые заключены между $\underset{\cdot\cdot}{x} = 0.5$ è ẍ=1. Для этих значений полагаем ε_1=1, для всех остальных ε_1=0. Например, для x=0,577 имеем ε_1=1, для x=0,170 имеем ε_1=0.

2) Среди всех значений y, соответствующих выделенным x, выбираем те, которые заключены между $\underline{y(x)}=0$ è $\overline{y(x)}=2x-1$. Для этих значений полагаем

$\varepsilon_2=1$, для всех остальных $\varepsilon_2=0$. Например, для н=0,701 имеем $\varepsilon_2=0$, для x=0,205 имеем $\varepsilon_2=1$

x	$\underline{x}$	$\overline{x}$	ε_1	y	$\underline{y(x)}$	$\overline{y(x)}$	ε_2	ε	f(x,y)
0.577	0.500	1.000	1	0.716	0	0.154	0	0	
0.737	0.500	1.000	1	0.701	0	0.474	0	0	
0.170	0.500	1.000	0	0.533			0	0	
0.432	0.500	1.000	0	0.263			0	0	
0.059	0.500	1.000	0	0.663			0	0	
0.355	0.500	1.000	0	0.094			0	0	
0.303	0.500	1.000	0	0.552			0	0	
0.640	0.500	1.000	1	0.205	0	0.280	1	1	0.452
0.002	0.500	1.000	0	0.557			0	0	
0.870	0.500	1.000	1	0.323	0	0.740	1	1	0.855
0.116	0.500	1.000	0	0.930			0	0	
0.930	0.500	1.000	1	0.428	0	0.860	1	1	1.048
0.529	0.500	1.000	1	0.095	0	0.058	0	0	
0.996	0.500	1.000	1	0.700	0	0.992	1	1	1.482
0.313	0.500	1.000	0	0.270			0	0	
0.653	0.500	1.000	1	0.934	0	0.306	0	0	
0.058	0.500	1.000	0	0.003			0	0	
0.882	0.500	1.000	1	0.986	0	0.764	0	0	
0.521	0.500	1.000	1	0.918	0	0.042	0	0	
0.071	0.500	1.000	0	0.139			0	0	
							Суммы	4	3,837

.

1) Вычисляем $\varepsilon=\varepsilon_1*\varepsilon_2$. Области интегрирования принадлежат только те точки, для которых ε=1. Например для точки M(0,640б0,205) имеем ε=1.
2) Вычисляем значение подынтегральной функции в полученных точках. После заполнения табл.2 вычисляем области интегрирования и по формуле (3) находим

$$I \approx \frac{1}{4}*\frac{1}{4}(0.452+0.855+1.048+1.482)=\frac{1}{16}*3.837=0.240$$

Точное значение интеграла I=0.21875. Результат имеет сравнительно небольшую точность потому, что N=20 не достаточно велико.

$$V_G=\frac{1}{2}*1*\frac{1}{2}=\frac{1}{4}$$

3 Задания для самостоятельной работы

Вычислить с указанной точностью ε несобственные интегралы методом усечения

1. $\int_1^{\infty} \frac{arctgx}{1+x^2} dx$ ε=10^{-2} 2. $\int_1^{\infty} \frac{xe^{-x^2}}{2+\sin x} dx$ ε=10^{-2}

Занятие 10 Использование алгоритмов псевдогенераторов случайных чисел

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* .

Для получения псевдослучайных чисел предлагалось использовать также квадратичные и кубические генераторы:

$$k_i = (a_1^2 \times k_{i-1} + a_2 \times k_{i-1} + b) \bmod c$$

$$k_i = (a_1^3 \times k_{i-1} + a_2^2 \times k_{i-1} + a_3 \times k_{i-1} + b) \bmod c$$

Однако такие генераторы тоже оказались непригодными для целей криптографии по той же самой причине «предсказуемости».

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:

$$k_6 = (5 * 1 + 3) \bmod 11 = 8.$$

А значения k_7 и k_8 будут равны 10 и 9 соответственно:

$$k_7 = (5 * 8 + 3) \bmod 11 = 10;$$

$$k_8 = (5 * 10 + 3) \bmod 11 = 9.$$

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

Определите последовательность из первых десяти чисел и период линейного конгруэнтного генератора ПСЧ для различных параметров *a*, *b* и *c*:

а) *a* =11, *b* =7 и *c* = 16; б) *a* =21, *b* =19 и *c* = 32;

в) *a* =19, *b* =7 и *c* = 32; г) *a* = 11, *b* =3 и *c* = 32.

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Тульский государственный университет»

Технический колледж имени С.И. Мосина

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

по дисциплине

«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ»

для специальности

09.02.03 «Программирование в компьютерных системах»

(углубленный уровень)

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий информационных технологий
Протокол от « 14 » апреля 2020 г. № 6
Председатель цикловой комиссии И.В.Миляева

Составитель Воронцова Н.В., И.В.Миляева,

Методические указания по выполнению курсовой работы по дисциплине «Математическое моделирование». Тула, Издательство Технического колледжа им. С.И. Мосина ТулГУ, 2020, 17с.

Методические указания по выполнению курсовой работы по дисциплине «Математическое моделирование» предназначены для студентов, обучающихся по специальности 09.02.03 «Программирование в компьютерных системах», для приобретения опыта решения оптимизационных задач и их программирования.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4

1 ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ 6

2 ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ 7

2.1 Тематика курсовой работы 7

2.2 Исходные данные к курсовой работе 7

2.2 Разработка плана курсовой работы 7

2.3 Выполнение курсовой работы 8

2.4 Оформление пояснительной записки 8

2.5 Защита курсовой работы 9

2.6 Методические рекомендации к курсовой работе 10

3 БИБЛИОГРАФИЧЕСКИЙ СПИСОК 11

ПРИЛОЖЕНИЕ А 12

ПРИЛОЖЕНИЕ Б 13

Образец задания на курсовую работу 13

ПРИЛОЖЕНИЕ В 14

График выполнения курсовой работы 14

ПРИЛОЖЕНИЕ Г 15

Пример оформления титульного листа 15

ПРИЛОЖЕНИЕ Д 16

Примеры оформления списка литературы 16

ВВЕДЕНИЕ

Курсовая работа является заключительным этапом обучения студентов по дисциплине "Математическое моделирование" и должна способствовать:

- закреплению, углублению, обобщению и систематизации знаний, полученных студентами за время изучения дисциплины;
- приобретению опыта моделирования и аналитического решения оптимизационных задач;
- усвоению современных методов и технологий программирования;
- формирования умений и навыков самостоятельного умственного труда, комплексной проверки уровня знаний и умений;
- развитию навыков работы с технической документацией, научной и справочной литературой,
- формированию у студентов общих (ОК) и профессиональных (ПК) компетенций.

.

Код	**Наименование результата обучения**
ОК 1	Понимать сущность и социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности в области обеспечения информационной безопасности
ОК 2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество
ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность
ОК 4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития
ОК 5	Использовать информационно-коммуникационные технологии в профессиональной деятельности
ОК 6	Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями
ОК 7	Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.
ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации
ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности
ОК 10	Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).
ОК 11	Применять проектный подход в профессиональной деятельности

Код	**Наименование результата обучения**
ПК 1.1	Выполнять разработку спецификаций отдельных компонент.
ПК 1.2	Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.4	Выполнять тестирование программных модулей.
ПК 1.5	Осуществлять оптимизацию программного кода модуля.
ПК 5.3	Выполнять работы по модификации отдельных компонент программного обеспечения.
ПК 5.4	Обеспечивать защиту программного обеспечения компьютерных систем.

В процессе подготовки курсовой работы студенты решают оптимизационные задачи различной тематики. Курсовую работу студенты выполняют по индивидуальным заданиям.

Обязательным условием выполнения курсовой работы студентом является творческое отношение к делу в сочетании с самостоятельностью выполнения порученной работы.

1 ЦЕЛЬ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Целью курсовой работы является решение оптимизационной задачи как аналитическими методами, так разработка программного обеспечения его реализации.

В процессе курсового проектирования студенты должны:

- создать математическую модель поставленной задачи;
- выбрать алгоритм решения задачи;
- описать алгоритм решения задачи;
- аналитически решить задачу;
- разработать программное обеспечение для реализации выбранного алгоритма;
- сравнить и проанализировать полученные результаты.

2 ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

Основными требованиями к курсовой работе являются обязательное выполнение в процессе оформления требований ЕСКД и ЕСПД (1-2), использование при разработке системы современных методов и приемов решения оптимизационных задач, строгое планирование процесса проектирования, составления и отладки программного продукта.

2.1 Тематика курсовой работы

Тема курсовой работы должна предполагать освоение определенного метода решения оптимизационной задачи, указанного в приложение А, и применение его при решении конкретного практического задания.

Тема курсовой работы может выдаваться группе студентов (не больше трех человек), при этом практические задания у каждого студента индивидуальные.

2.2 Исходные данные к курсовой работе

Курсовая работа выполняется в соответствии с индивидуальным заданием на курсовую работу, бланк которого приведен в приложении Б.

Курсовая работа ориентирована на создание программного продукта, реализующего заданный метод. Выбор программных средств зависит от решаемой задачи и подлежит согласованию с преподавателем.

По согласованию с руководителем курсовой работы студент может предлагать свою тему курсовой работы.

2.2 Разработка плана курсовой работы

Студент под руководством преподавателя-руководителя составляет рабочий план курсовой работы, т.е. определяет ее основные элементы.

Курсовая работа носит практический характер и содержит:

- титульный лист;
- содержание, которое включает наименование всех разделов и подразделов работы с указанием номеров страниц, на которых размещается начало разделов и подразделов;
- введение, в котором раскрывается актуальность и значение темы, формулируется цель работы;
- основная часть, которая состоит из двух разделов:

  - в первом разделе содержаться теоретические основы разрабатываемой темы;

  - вторым разделом является практическая часть, которая представлена расчетами, таблицами, схемами, программами;

- заключение, в котором содержатся выводы и рекомендации о возможности практического применения материалов работы;
- список использованных источников;
- приложения.

В соответствии с рабочим планом составляется график написания курсовой работы, т.е. намечаются сроки проработки каждого раздела и представления материалов для просмотра руководителю. Пример оформления графика выполнения курсовой работы приведен в приложении В.

2.3 Выполнение курсовой работы

Подготовка и написание курсовой работы включает ряд последовательных этапов:

1 Выбор темы и получение от руководителя первоначальных рекомендаций по методам исследования и источникам необходимой информации.

2 Изучение литературы и ее анализ, составление библиографического списка использованных источников.

3 Разработка плана курсовой работы.

4 Формулирование основных теоретических положений.

5 Аналитическое решение практического задания.

6 Разработка алгоритма реализации заданного метода с применением вычислительной техники.

7 Создание программного продукта, реализующего разработанный алгоритм.

8 Формулировка выводов, рекомендаций.

9 Оформление курсовой работы и составление отзыва на нее руководителем.

10 Защита курсовой работы.

2.4 Оформление пояснительной записки

По результатам курсового проектирования составляется пояснительная записка, которая состоит из описания всех этапов работы, и приложения, включающие распечатки кода машинной реализации задачи. Образец титульного листа пояснительной записки приведен в приложении Г.

Пояснительная записка должна содержать следующие разделы:

ВВЕДЕНИЕ

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Общая постановка задачи

1.2 Алгоритм решения задачи

2 ПРАКТИЧЕСАКАЯ ЧАСТЬ

2.1 Постановка задачи

2.2 Аналитическое решение задачи
2.3 Решение задачи с применение средств вычислительной техники
2.4 Сравнение полученных результатов
ЗАКЛЮЧЕНИЕ
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ
ПРИЛОЖЕНИЯ

Объем курсовой работы должен быть не менее 15-20 страниц печатного текста на одной стороне писчей бумаги формата А4.

Пояснительная записка оформляется в соответствии с ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам.

Размеры полей при оформлении пояснительной записки: левое поле – не менее 20 мм, верхнее поле – не менее 20 мм, правое поле – не менее 20 мм, нижнее поле – не менее 10 мм.

Для компьютерного набора текста используется гарнитура «Times New Roman» размером кегля 12 пунктов с полуторным межстрочным интервалом или 14 пунктов с одинарным межстрочным интервалом. Нумерация страниц сквозная и проставляется в правом верхнем углу страницы. Первой страницей является титульный лист, на котором номер страницы не проставляется (приложение Г). Вторым листом работы является лист задания.

Рисунки и схемы выполняются вручную черной тушью или с помощью компьютера. Схемы алгоритмов выполняются в точном соответствии с ГОСТ. Рисунки нумеруются последовательно арабскими цифрами или в пределах раздела и могут сопровождаться пояснительными подписями. На все рисунки должны быть ссылки в тексте. Рисунки помещаются после первого упоминания в тексте.

Цифровой 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. Затем следует заголовок таблицы. При ссылке на таблицу указывается ее номер, например: (таблица 1).

Записка подписывается студентом и руководителем.

2.5 Защита курсовой работы

В установленные сроки выполненная курсовая работа представляется руководителю для составления отзыва, в котором определяется соответствие работы заявленной теме, уровень, качество и полнота разработки поставленных вопросов, оценивается теоретическая и практическая значимость курсовой работы и делается вывод о готовности студента к защите курсовой работы.

Защита курсовой работы включает:

- сообщение студента по содержанию работы в течение 10 минут, используя необходимый графический материал (презентацию) и демонстрируя работу программы на компьютере;
- вопросы руководителя студенту по теме работы;

- высказывание руководителя по поводу качества выполнения работы;
- выставление оценки.

Оценка за курсовую работу (проект) выставляется с учетом качества ее выполнения и оформления.

Основными критериями оценки курсовой работы (проекта) являются:

- актуальность и степень разработки темы;
- творческий подход и самостоятельность в анализе, расчетах, разработки программного обеспечения, обобщениях, выводах;
- правильность проведенных расчетов и работы программы;
- новизна и практическая значимость выводов и рекомендаций, их обоснованность;
- соблюдение требований к оформлению работы, аккуратности исполнения.

Курсовая работа оценивается по пятибалльной системе. Положительная оценка выставляется при успешной защите курсовой работы на оценку не ниже оценки «удовлетворительно».

2.6 Методические рекомендации к курсовой работе

2.6.1 Теоретическая часть

В первом разделе содержаться теоретические основы разрабатываемой темы. Автор работы должен изложить общую постановку задачи и привести общий алгоритм решения задачи выбранным методом в соответствии с темой курсовой работы.

2.6.2 Практическая часть

Второй раздел основной части пояснительной записки должен включать постановку практической задачи, в случае необходимости описание построения ее математической модели.

В подразделе «Аналитическое решение задачи» необходимо привести ход решения практической задачи с помощью таблиц, систем уравнений, описывая каждый шаг.

Подраздел «Решение задачи с применение средств вычислительной техники» должен содержать обоснование выбора языка программирования, описание структуры программы реализации выбранного метода, технологию работы с программой.

В подразделе «Сравнение полученных результатов» необходимо проанализировать результаты решения практической задачи аналитическим способом и с использованием компьютера.

3 БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных систем. Условные обозначения и правила выполнения.

2 ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» [Текст]. . – Введ. 2001-05-22. – М. : Госстандарт России : Изд-во стандартов, 2017.

3 Зализняк, В. Е. Введение в математическое моделирование : учебное пособие для среднего профессионального образования / В. Е. Зализняк, О. А. Золотов. — Москва : Издательство Юрайт, 2020. — 133 с. — (Профессиональное образование). — ISBN 978-5-534-13307-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/457484

4 Кудрявцев, К. Я. Методы оптимизации : учебное пособие для вузов / К. Я. Кудрявцев, А. М. Прудников. — 2-е изд. — Москва : Издательство Юрайт, 2020. — 140 с. — (Высшее образование). — ISBN 978-5-534-08523-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/455825

Дополнительные источники:

1 Рейзлин, В. И. Математическое моделирование : учебное пособие для вузов / В. И. Рейзлин. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 126 с. — (Высшее образование). — ISBN 978-5-534-08475-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451402

2 Горлач, Б. А. Математическое моделирование. Построение моделей и численная реализация : учебное пособие / Б. А. Горлач, В. Г. Шахов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2018. — 292 с. — ISBN 978-5-8114-2168-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/103190

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/

ПРИЛОЖЕНИЕ А

Примерные темы курсовых работ

1. Реализация метода искусственного базиса
2. Реализация модифицированного симплекс метода
3. Реализация симплекс метода в случае положительных свободных членов
4. Реализация симплекс метода в случае отрицательных свободных членов
5. Геометрическое истолкование задачи линейного программирования
6. Решение транспортной задачи. Определение опорного плана методом минимального элемента (наименьших затрат).
7. Транспортная задача и ее решение методом северо-западного угла
8. Реализация задачи о нахождении кратчайшего пути в графах
9. Реализация задачи о нахождении максимального потока в графе
10. Реализация задачи о замене оборудования.
11. Реализация задачи о распределении средств между предприятиями.
12. Реализация задачи о нахождении оптимального плана игры (игра по выбору)
13. Реализация задачи о теории расписаний.
14. Решение уравнения Колмогорова.
15. Реализация задач имитационного моделирования.
16. Реализация задачи прогнозирования потребительского спроса.
17. Реализация задачи теории принятия решений.
18. Реализация задачи метода Форда.
19. Реализация задачи метода Флойда
20. Реализация распределительной задачи.
21. Реализация задачи оптимального ассортимента.
22. Реализация балансовой задачи.
23. Реализация задачи оптимального графика производства.

ПРИЛОЖЕНИЕ Б

Образец задания на курсовую работу

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

Задание для курсовой работы

По дисциплине (МДК)_____
студенту _____ курса, группы _____
специальность_____
_____
_____
(фамилия, имя, отчество студента)

на тему:

_____
_____
_____

Отзыв на курсовую работу

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Дата выдачи «___» _____ 20__г.

Срок сдачи «___» _____ 20__г.

Руководитель _____ _____
(подпись) *(И.О.Фамилия)*

ПРИЛОЖЕНИЕ В

Образец графика выполнения курсовой работы

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

График выполнения курсовой работы

_____

(*фамилия, имя, отчество студента*)

Студента(ки)_____курса _____группы _____
специальность_____
_____

Тема курсовой работы:_____
_____
_____

Наименование	**Сроки**	**Отметка о выполнении**
1 Разработка содержания курсовой работы:		
Введение		
Раздел 1		
1.1		
1.2		
Раздел 2		
2.1		
2.2		
2.3		
Заключение		
Подготовка приложений		
2 Оформление курсовой работы		
Оформление пояснительной записки		
Оформление приложений		
Оформление графической части		

Руководитель _____ _____
(подпись) *(И.О.Фамилия)*

ПРИЛОЖЕНИЕ Г

Пример оформления титульного листа

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

КУРСОВАЯ РАБОТА

по дисциплине «Математическое моделирование»

на тему: «Нахождение кратчайшего пути»

Автор работы,
студентка гр.З-090203 **А.А.Петрова**

Руководитель,
преподаватель **П.П.Иванова**

Тула 2020

ПРИЛОЖЕНИЕ Д

Примеры оформления списка литературы

Вид издания	Оформление по ГОСТ Р 7.0.5-2008
Книга одного автора	Аверин А.К. Приспособления для металлорежущих станков: справочник. 7-изд., перераб. М.: Машиностроение, 1991. 303с.
Книга двух авторов	Болотин Ч.Л., Костромин С.П. Станочные приспособления: учеб. Пособие. 5-е изд. перераб. и доп. М.: Высш.шк.,1992.344с.
Книга трех авторов	Зверев М.П., Рыжков Э.В., Аверченков А.В. Технологическая оснащенность в машиностроении. Минск: Наука и техника. 1992.443с.
Книга четырех и более авторов	Дипломное проектирование по технологии машиностроения: учеб. пособие/ В.В. Бабук [и др.]. Минск: Высш.шк.,1992. 464с.
Том многотомного издания	Савельев И.В. Курс общей физики: в 3 т. Т.1. Механика. Молекулярная физика: учеб. пособие. 21-е изд., перераб. и доп. М.: Наука, 1992. 432с.
Журнал	Вестник Тульского Государственного университета. Серия «Современные проблемы законодательства России, юридических наук и правоохранительной деятельности. Вып.3 / ТулГУ: редкол. Т.Н. Толстухин и др.. Тула, 2000. 200с.
Статья из журнала	Дементьева А.А. Эффективность научных исследований // Изв. вузов. Машиностроение. 1991. №6. С. 4-9
ГОСТ	ГОСТ 7.53-2001. Издания. Международная стандартная нумерация книг. Взамен ГОСТ 7.53-86 ; введ. 2000-07-01. М.: Изд-во стандартов, 2001.75с.
Книга на иностранном языке	Sosodia M.N. Microwave circuits and passive devices. New York: Wiley, 1991. 240p.
Статья из иностранного журнала	Parker Susan N/ What s new in metallcutin research // Amer/ Mach. 1992. Vol. 129/ № 7 P. 75-77
Статья из сборника трудов	Михайлов А.И. Методика расчета оптимального режима резания// Труды / Горьков. политех. ин-т. Горький, 1992. Т.: 128. С.75-77
Патент	Приемопередающее устройство: пат. 218788 Рос. Федерация . № 2000131736/09; заявл. 18.12.00, опубл. 20.08.02. Бюлл. № 23(II ч.) 3с.

Воронцова Наталья Вадимовна

Миляева Ирина Вилорьевна

Методические указания по выполнению курсовой работы
по дисциплине «Математическое моделирование»

для студентов специальности
09.02.03
Программирование в компьютерных системах

Ответственный за выпуск Миляева И.В.

Издательство Технического колледжа им. С.И. Мосина ТулГУ
18 проезд, д. 94, п.Мясново, г.Тула

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

Цикловая комиссия информационных технологий

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

по дисциплине
«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ»

специальности
09.02.03 Программирование в компьютерных системах

(углубленная подготовка)

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий информационных технологий
Протокол от «13» января 2020 г. № 6
Председатель цикловой комиссии И.В.Миляева

Составитель Воронцова Н.В.

Методические указания по выполнению лабораторных работ по дисциплине «Математическое моделирование».Тула, Издательство Технического колледжа им. С.И. Мосина ТулГУ, 2020, 10 с.

Методические указания по выполнению лабораторных работ по дисциплине «Математическое моделирование». предназначены для студентов, обучающихся по специальности 09.02.03 Программирование в компьютерных системах (углубленная подготовка), для приобретения опыта создания информационного обеспечения будущей профессиональной деятельности с применением информационных технологий.

СОДЕРЖАНИЕ

Лабораторная работа № 1
Решение оптимизационной задачи с помощью инструментальных средств,,,…………… 4

Тема 2.2 Линейное программирование

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема работы: Решение оптимизационной задачи с помощью инструментальных средств

Цель работы: *уметь:*

- подбирать аналитические методы исследования математических моделей;
- использовать численные методы исследования математических моделей;
- работать с пакетами прикладных программ аналитического и числового исследования математических моделей;

формировать общие и профессиональные компетенции:

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности

ПК 1.2 Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

Материально-техническое оснащение: ПК, операционная система Windows

Количество часов: 2 часа.

I.Теоретическая часть

Исходные данные транспортной задачи приведены схематически: внутри прямоугольника заданы удельные транспортные затраты на перевозку единицы груза *(c_{ij})*, слева указаны мощности поставщиков (а,), а сверху - мощности потребителей *(b_j)*. Найти оптимальный план закрепления поставщиков за потребителями *(x_{ij})*.

Мощности поставщиков	Мощности потребителей			
	250	100	150	50
80	6	6	1	4
320	8	30	6	5
100	5	4	3	30
50	9	9	9	9

В данной задаче суммарные запасы равны суммарным потребно-

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \$50.$$

Транспортная задача, в которой суммарные запасы и потребности совпадают, является закрытой.

Ввод условий задачи состоит из следующих основных шагов:

1. Создание формы для ввода условий задачи.
2. Ввод исходных данных.
3. Ввод зависимостей из математической модели.
4. Назначение целевой функции.
5. Ввод ограничений и граничных условий.

Матрица перевозок(изменямые ячейки)

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

исходные данные

	250	100	150	50
80	6	6	1	4
320	8	30	6	5
100	5	4	3	30
50	9	9	9	9

Рис 1 . Создание формы для ввода условий задачи.

Изменяемые ячейки **B3:B6.** В эти ячейки будет записан оптимальный план перевозки x_{ij}. Введены исходные данные задачи.

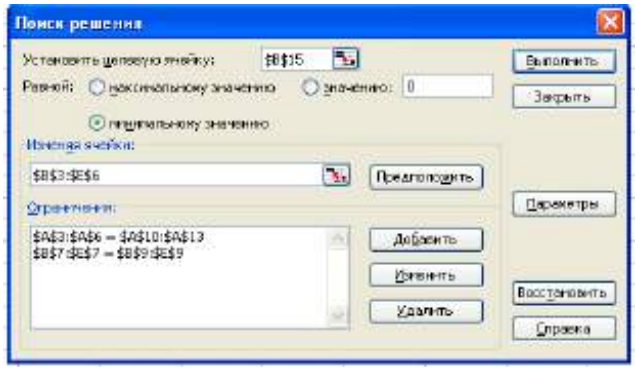


Рис.2 Введены зависимости из математической модели. Выражение для вычисления целевой функции получено с помощью функции СУММПРОИЗВ(B3:B6БВ10:E13)

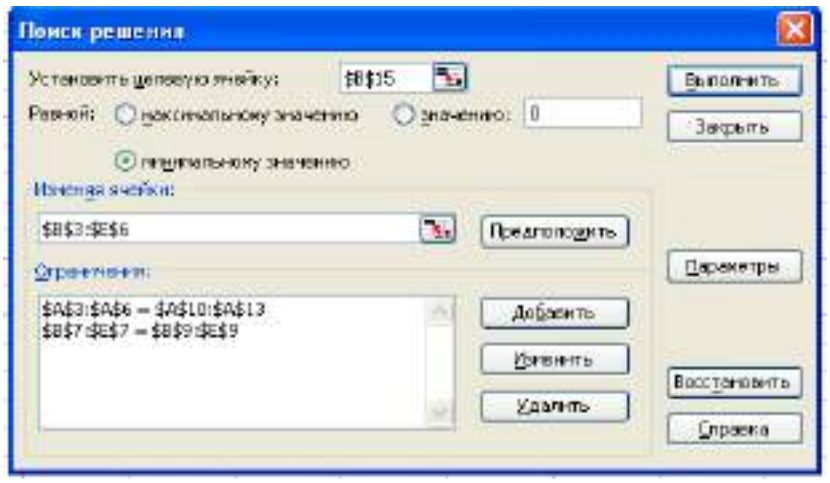


Рис.3 На экране диалоговое окно *Поиск решения*

После вызова *Поиска решения* курсор подвести в поле «Установить целевую ячейку» и ввести адрес: B15. Ввести направление целевой функции *«минимальному значению».* Поместить курсор в поле «Изменяя ячейки». Ввести адреса изменяемых ячеек B3:E6. Далее следует *добавить ограничения.*

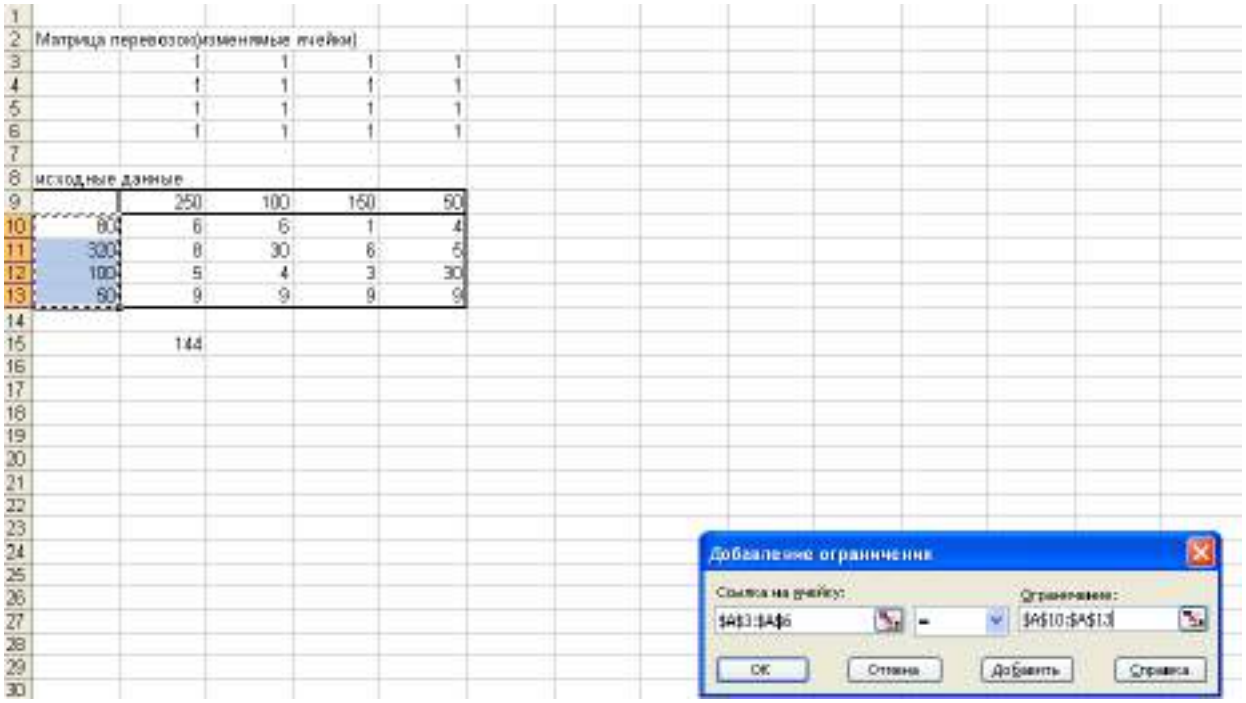


Рис 4. На экране диалоговое окно добавление ограничения

Все грузы должны быть перевезены, т.е.

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1,\ldots,m - \mathbf{A3:A6 = A10:A13.}$$

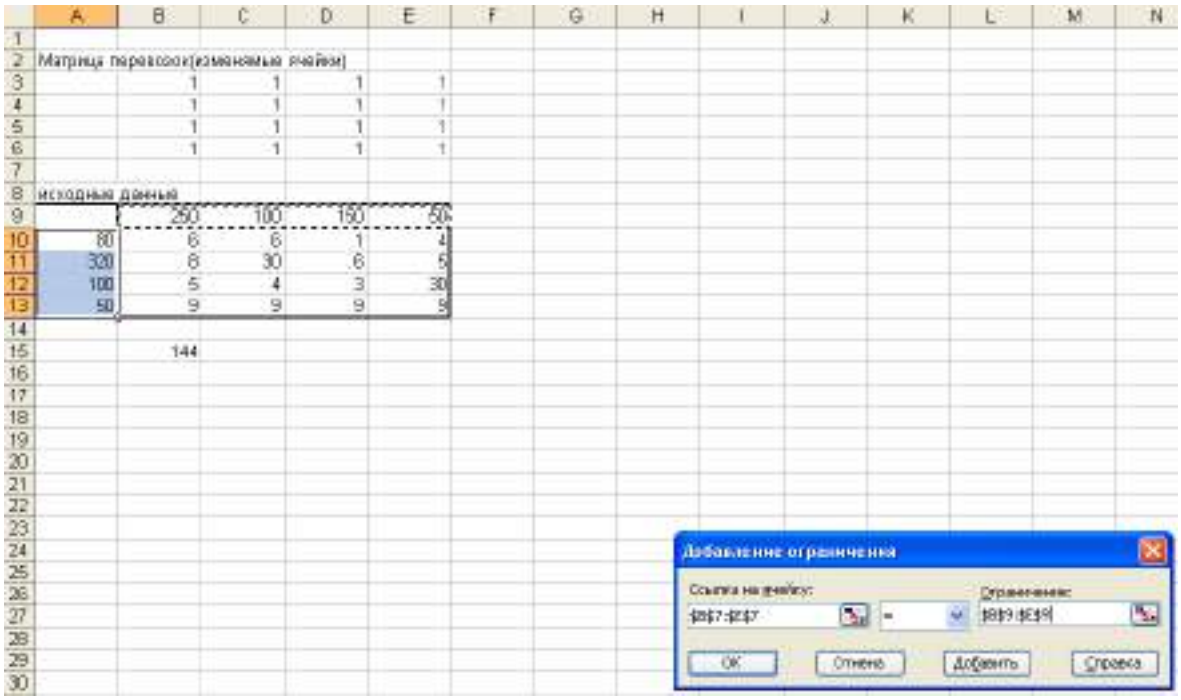


Рис. 5. На экране диалоговое окно *Добавление ограничения.*

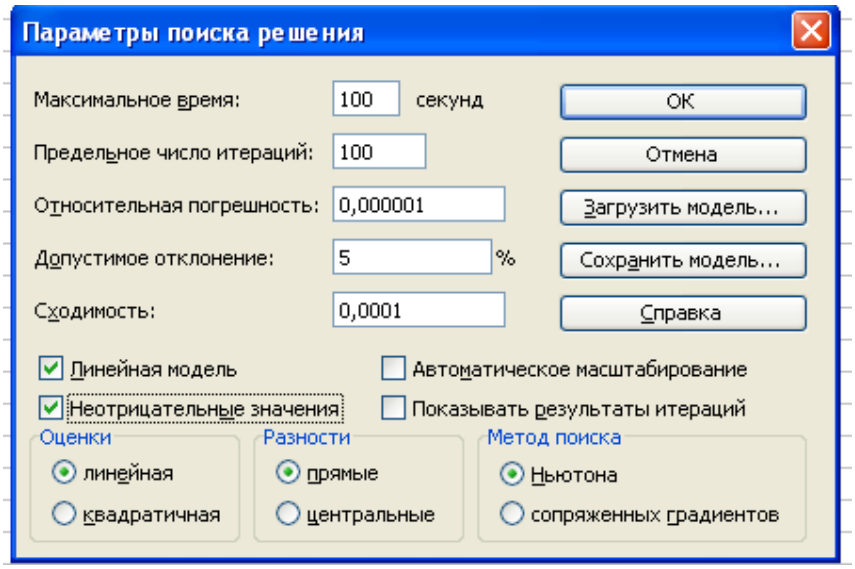
Все потребности должны быть удовлетворены, т.е.

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,\ldots,n - \mathbf{B7:E7 = B9:E9.}$$

После ввода последнего ограничения вместо *добавить* ввести ОК. На экране появится окно *Поиск решения* с введенными ограничениями (см. рис. 2.).

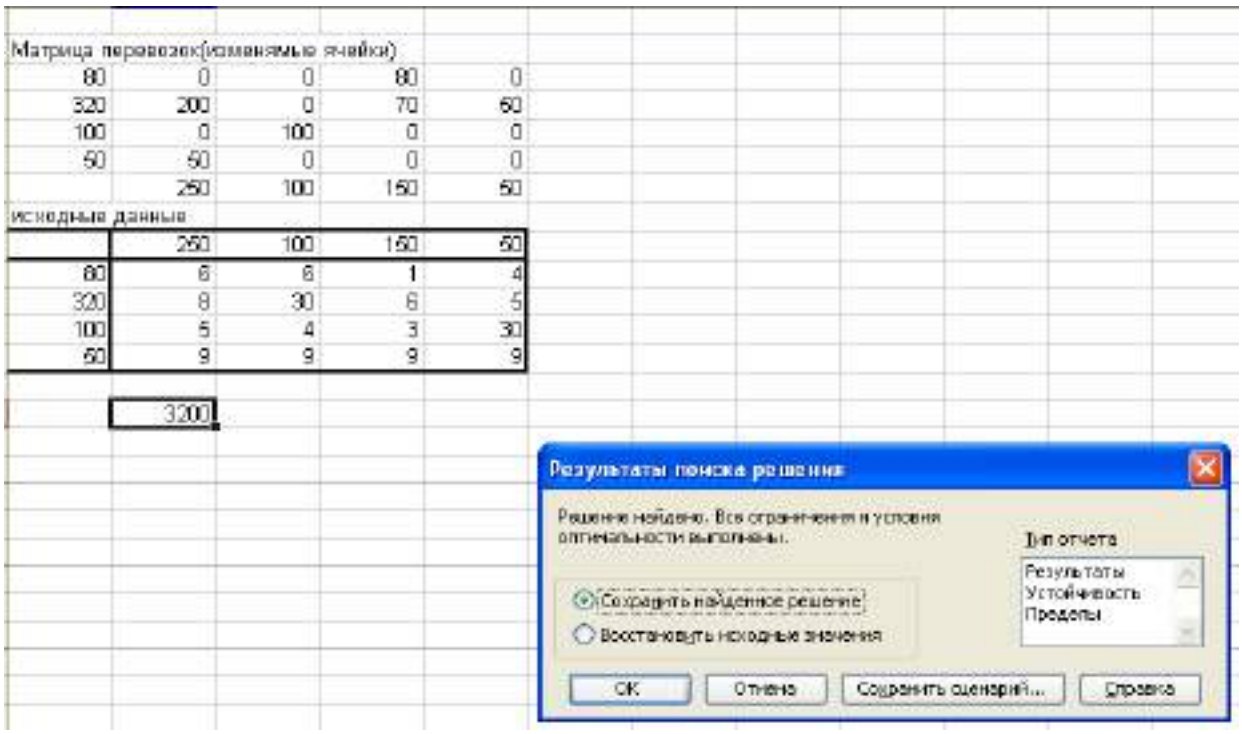
Решение задачи

Решение задачи производится сразу же после ввода данных, когда на экране находится диалоговое окно *Поиск решения* (рис. 2.4.3). С помощью окна *Параметры* можно вводить условия для решения оптимизационных задач. В нашей задаче следует установить флажок *«неотрицательные значения»* и флажок *«линейная модель»* (рис. 2.4.6). Нажать кнопку ОК. Опять появится диалоговое окно *Поиск решения.*



В результате решения получен оптимальный план перевозок:

Матрица перевозок (изменяемые				
80	**0**	**0**	**so**	0
320	**200**	**0**	70	50
100	**0**	**100**	0	0
50	**50**	**2.13E-X4**	0	0
550	250	100	150	50



Хв = 80 ед. груза следует перевезти от 1-го поставщика 3-му потребителю;

Х21 = 200 ед. груза следует перевезти от 2-го поставщика 1-му потребителю;

X_2з = 70 ед. груза следует перевезти от 2-го поставщика 3-му потребителю;

Х24 = 50 ед. груза следует перевезти от 2-го поставщика 4-му потребителю;

Хзг = 100 ед. груза следует перевезти от 3-го поставщика 2-му потребителю;

$X_4$1 = 50 ед. груза следует перевезти от 4-го поставщика 1 -му потребителю;

Х42 = 0 ед. груза следует перевезти от 4-го поставщика 2-му потребителю.

Общая стоимость перевозок = 3200.

III. Контрольные вопросы

1. Какая модель транспортной задачи является открытой?
2. Какая модель транспортной задачи является закрытой?
3. . Какие ячейки называются зависимыми?
4. Как работает надстройка Поиск решения.
5. Как работает функция СУММПРОИЗВ?

IV. Оформление отчёта

Отчет по лабораторно-практической работе составляется по следующей структуре:

1. Наименование лабораторной работы.
2. Цель работы.
3. Описание назначения команд и их параметров.
4. Результаты работы - изображение структуры каталогов.
5. Ответы на контрольные вопросы.
6. Вывод по работе.

Воронцова Наталья Вадимовна

Методические указания по выполнению лабораторных работ по дисциплине «Математическое моделирование», обучающихся по специальности ***09.02.03*** Программирование в компьютерных системах (углубленная подготовка**)**

Ответственный за выпуск Миляева И.В.

Издательство Технического колледжа им. С.И. Мосина ТулГУ

18 проезд, д. 94, п.Мясново, г.Тула

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

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РЕФЕРАТА

по дисциплине «Математическое моделирование»

специальности 09.02.03 Программирование в компьютерных системах

(углубленная подготовка)

Тула 2020

УТВЕРЖДЕНЫ
Цикловой комиссий информационных технологий
Протокол от «13» января 2020 г. № 6
Председатель цикловой комиссии И.В.Миляева

Авторы: Воронцова Н .В., канд. техн. наук

СОДЕРЖАНИЕ

ВВЕДЕНИЕ……..... 4

1 СТРУКТУРА РЕФЕРАТА…… 5

2 ОФОРМЛЕНИЕ РЕФЕРАТА…… 6

3 ПРИМЕРНАЯ ТЕМАТИКА РЕФЕРАТА…… 8

БИБЛИОГРАФИЧЕСКИЙ СПИСОК……... 9

ВВЕДЕНИЕ

Реферат - краткое изложение в письменном виде или в форме публичного выступления содержания книги, научной работы, результатов изучения научной проблемы.

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, по сути, нужно пересказать его (желательно своими словами). В реферате не требуется наличия большого фактического материала, глубокого анализа, фундаментальных выводов.

1 СТРУКТУРА РЕФЕРАТА

Реферат должен включать оглавление, введение, несколько глав (от 2 до 5), заключение и список использованных источников.

Структура обычного реферата:

- содержание;
- введение;
- несколько глав (от 2 до 5);
- заключение;
- список литературы (или библиографический список).

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 параграфы. Главы можно заканчивать выводами.

В заключении должны быть ответы, на поставленные во введении задачи и дан общий вывод. Объем заключения реферата - 1-1,5 страницы.

Список использованных источников для реферата обычно должен включать 4-12 позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы.

У реферата могут быть приложения - картинки, схемы и прочие.

2 ОФОРМЛЕНИЕ РЕФЕРАТА

Размеры полей при оформлении реферата: левое поле – не менее 20 мм, верхнее поле – не менее 20 мм, правое поле – не менее 20 мм, нижнее поле – не менее 10 мм.

Для компьютерного набора текста используется гарнитура «Times New Roman» размером кегля 12 пунктов с полуторным межстрочным интервалом или 14 пунктов с одинарным межстрочным интервалом. Нумерация страниц сквозная и проставляется в правом верхнем углу страницы. Первой страницей является титульный лист, на котором номер страницы не проставляется (приложение).

Каждая из частей реферата начинается с новой страницы. Заголовки каждой части реферата пишутся заглавными буквами и размещаются по центру строки. Между заголовком и последующим текстом должна быть пустая строка.

Главы реферата могут делиться на параграфы (если реферат небольшой, то лучше этого не делать). Заголовок параграфа пишется строчными буквами с заглавной, размещается «по ширине страницы» и с отступом красной строки. Пропуска строки между заголовком параграфа и последующим текстом не делается. Главы и параграфы реферата нумеруются. Точка после номера не ставится. Номер параграфа 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.

Текст реферата размещается с центрированием «по ширине страницы». Абзацы выделяются красной строкой с отступом не менее 1,27 см.

Рисунки нумеруются последовательно арабскими цифрами или в пределах главы: в каждой главе начинается заново (тогда номер рисунка перед собственно своим номером через точку содержит номер главы). Рисунки могут сопровождаться пояснительными подписями (*Пример подписи рисунка: Рисунок 1 – Схема кодирования*). На все рисунки должны быть ссылки в тексте. Рисунки помещаются после первого упоминания в тексте.

Цифровой 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).

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.

Список использованных источников для реферата обычно должен включать 4-12 позиций - нормативные акты, книги, печатную периодику, интернет-ресурсы. Впереди идут нормативные акты, потом книги, далее печатная периодика, источники с электронных носителей (например, «Консультант Плюс» или CD-издания), далее интернет-источники.

Очень желательно, чтобы в реферате были ссылки. Количество ссылок для реферата - от 2 до 10. Ставить ссылки можно двумя способами: за текстом номер ссылки в верхнем регистре - и внизу страницы название источника; за текстом в квадратных скобках с указанием номера источника по списку литературы. Ссылки безусловны на все точные числовые данные и на все прямые цитаты.

3 ПРИМЕРНАЯ ТЕМАТИКА РЕФЕРАТА

1. Применение аналитических моделей.
2. Применение модели потока заявок.
3. Применение вероятностных моделей процессов.
4. Применение статистических моделей.
5. Применение имитационных моделей.
6. Применение метода статистических испытаний.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основные источники:

1 Зализняк, В. Е. Введение в математическое моделирование : учебное пособие для среднего профессионального образования / В. Е. Зализняк, О. А. Золотов. — Москва : Издательство Юрайт, 2020. — 133 с. — (Профессиональное образование). — ISBN 978-5-534-13307-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/457484

2 Кудрявцев, К. Я. Методы оптимизации : учебное пособие для вузов / К. Я. Кудрявцев, А. М. Прудников. — 2-е изд. — Москва : Издательство Юрайт, 2020. — 140 с. — (Высшее образование). — ISBN 978-5-534-08523-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/455825

Дополнительные источники:

1 Рейзлин, В. И. Математическое моделирование : учебное пособие для вузов / В. И. Рейзлин. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 126 с. — (Высшее образование). — ISBN 978-5-534-08475-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451402

2 Горлач, Б. А. Математическое моделирование. Построение моделей и численная реализация : учебное пособие / Б. А. Горлач, В. Г. Шахов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2018. — 292 с. — ISBN 978-5-8114-2168-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/103190

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/

ПРИЛОЖЕНИЕ

Пример оформления титульного листа

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

РЕФЕРАТ

по дисциплине «Математическое моделирование»

на тему: «Кодирование информации»

Автор работы,
студентка гр.3-090203 **А.А.Петрова**

Руководитель,
преподаватель **П.П.Иванова**

Тула 2021

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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к практическим и самостоятельным работам

по дисциплине «Основы информатики»

для специальности 09.02.03 Программирование в компьютерных системах

Тула 2020

УТВЕРЖДЕНА
Цикловой комиссий информационных технологий
Протокол от «15» января 2020 г. №6

Председатель цикловой комиссии И.В.Миляева

Авторы: Миляева И.В., преподаватель

Тема 1.3. Статистическая мера информации
Практическое занятие: «Определение количества информации в сообщении»
Теоретическая часть

Логарифмическая мера позволяет вычислять количество информации, - бит;

$$I = \log_2 N, \text{ где}$$

N – количество возможных равновероятностных событий (формула Хартли).

Величина, вычисляемая по формуле Хартли, является максимальным количеством информации.

Встатистической теории информации вводится более общая мера количества информации, в соответствии с которой рассматривается не само событие, а информация о нем. Этот вопрос глубоко проработан К.Шенноном в работе «Избранные труды по теории информации». Если появляется сообщение о часто встречающемся событии, вероятность появления которого близка к единице, то такое сообщение для получателя малоинформативно. Столь же малоинформативны сообщения о событиях, вероятность появления которых близка к нулю.

Событияможно рассматривать как возможные исходы некоторого опыта, причем все исходы этого опыта составляют ансамбль, или полную группу событий. К. Шеннион ввел понятие неопределенности ситуации, возникающей в процессе опыта, назвав ее энтропией. *Энтропия* ансамбля есть количественная мера его неопределенности и, следовательно, информативности, количественно выражаемая как средняя функция множества вероятностей каждого из возможных исходов опыта.

Средняя информация, доставляемая одним опытом$I_{\text{ср}}$ вычисляется по формуле Шеннона

$$I_{\text{ср}} = -\sum_{i=1}^{k} p_i \log_2 p_i = H, \text{ где}$$

k- разныхтиповисхода опыта,

p_i – вероятность появления i-го типа исхода (формула Шеннона).

Данная величина называется энтропией и обозначается обычно буквой H.

Практическая часть

Пример 1. Рассчитать количество информации, которое получится по окончании опыта, состоящего изпроцедуры бросаниямонеты, пирамиды, кубика, рулетки.События (исходы) опытов равновероятны, т.е. монета, пирамидка и кубик симметричны и однородны, а «честная» рулетка имеет 64 сектора.

Решение.

Количество возможных событий для монеты N=2, для пирамиды N=4, для кубика N=6, для рулетки N=64.

Рассчитываем количество информации, которое получится по окончании опытапо формуле Хартли:

для монеты $I = \log_2 2 = 1$ бит;
для пирамиды $I = \log_2 4 = 2$ бит;
для кубика $I = \log_2 6 \approx 2{,}6$ бит;
для рулетки $I = \log_2 64 = 6$ бит.

Ответ: Количество информации

для монеты $I = 1$ бит;
для пирамиды $I = 2$ бит;
для кубика $I \approx 2{,}6$ бит;
для рулетки $I = 6$ бит.

Пример 2. Определить количество информации, связанное с появлением каждого символа в сообщениях, записанных на русском языке. Считать, что русский алфавит состоит из 33 букв и знака «пробел» для разделения слов.

Решение.

1) Найдем количество возможных событий N,

$$N=33+1=34$$

2) Рассчитываем количество информации по формуле Хартли:

$$I = \log_2 34 \approx 5 \text{ бит.}$$

Ответ: Количество информации$I \approx 5$ бит.

Пример 3. Определить количество информации, связанное с появлением каждого символа в сообщениях, записанных на русском языкес учетом того, что в словах русского языка (равно как и в словах других языков) различные буквы встречаются неодинаково часто. В таблице 1 приведены вероятности частоты употребления различных знаков русского алфавита, полученные на основе анализа очень больших по объему текстов.

Решение.

Рассчитываем количество информации по формуле Шеннона:

$$I = -\sum_{i=1}^{34} p_i \log_2 p_i \approx 4{,}72 \text{ бит.}$$

Таблица 1 Частотность букв русского языка

i	Символ	*P(i)*	*i*	Символ	*P(i)*	*i*	Символ	*P(i)*
1	Пробел	0,175	13	К	0,028	24	Г	0,012
2	О	0,090	14	М	0,026	25	Ч	0,012
3	Е	0,072	15	Д	0,025	26	Й	0,010
4	Ё	0,072	16	П	0,023	27	Х	0,009
5	А	0,062	17	У	0,021	28	Ж	0,007
6	И	0,062	18	Я	0,018	29	Ю	0,006
7	Т	0,053	19	Ы	0,016	30	Ш	0,006
8	Н	0,053	20	З	0,016	31	Ц	0,004
9	С	0,045	21	Ь	0,014	32	Щ	0,003
10	Р	0,040	22	Ъ	0,014	33	Э	0,003
11	В	0,038	23	Б	0,014	34	Ф	0,002
12	Л	0,035						

Ответ: Количество информации$I \approx 4{,}72$ бит.

Задания для самостоятельной работы

1. Сравните полученные результаты примеров 2 и 3 и дайте объяснение.

2. Определить количество информации, связанное с появлением каждого символа в сообщениях, записанных на языках, использующих латинский алфавит, - английского, немецкого, французского и др. (26 различных букв и «пробел»).

3. Определить количество информации, связанное с появлением каждого символа в сообщениях, составив таблицы частотности букв для каждого сообщения:

а) Для получения дополнительного кода отрицательных чисел можно воспользоваться следующим образом: сначала инвертируется цифровая часть исходного числа, в результате получается его обратный код; затем добавляется единица в младший разряд цифровой части числа и тем самым получается дополнительный код этого изображения.

б) В простейшем виде табличный метод чисел из одной системы счисления в другую заключается в следующем: имеется таблица всех чисел одной системы с соответствующими эквивалентами из другой системы; задача перевода сводится к нахождению соответствующей строки таблицы и выбору из нее эквивалента. Такая таблица очень громоздка и требует большой емкости памяти для хранения.

в)Естественная форма представления числа в цифровом автомате характеризуется тем, что положение его разрядов в автоматном изображении остается всегда постоянным независимо от величины самого числа. Существует также другое название этой формы записи чисел —

представление чисел с фиксированной запятой или с фиксированной точкой.

г) Входные величины независимо от количества значащих цифр могут содержать грубые ошибки, возникающие из-за опечаток, ошибочных отсчетов показаний каких-либо приборов, некорректной постановки задачи или отсутствия более полной и точной информации. При переводе чисел из одной системы счисления в другую неизбежно возникают погрешности, оценить которые нетрудно, если известны истинные значения входных чисел.

Замечание: при выполнении задания заполните таблицу:

Символ	n_i	p_i	$log_2 p_i$	$p_i log_2 p_i$
…		…	…	…
				Σ

Тема 1.6. Формы представления информации
Практическое занятие: «Применение помехоустойчивого кодирования»
Теоретическая часть

Из опыта известно, что высоконадежное оборудование нелегко построить.

В случаях, когда возможно повторение, часто бывает достаточно лишь обнаружить ошибку. После обнаружения ошибок, сообщение передается повторно и при некоторой удаче второе (или, возможно, третье) повторение будет верным.

Нельзя обнаружить ошибку, если любой принятый символ может служить сообщением. Ошибки можно обнаружить только в том случае, если на возможные сообщения наложены некоторые ограничения.

Задача состоит в том, чтобы свести ограничения на возможные сообщения к достаточно простым (легко вычислимым).

Кодыс обнаружением ошибок

Простые проверки на четность

Простейший способ кодирования двоичного сообщения для обнаружения ошибки состоит в подсчете числа 1 в сообщении и добавлении еще одного с тем, чтобы во всем сообщении содержалось четное число 1,таким образом, к (n-1) символам сообщения добавляется n-я позиция для проверки на четность. На приемном конце производится подсчет числа 1, и нечетное число 1 во всех n позициях указывает на появление, по крайней мере, одной ошибки.

Ясно, что такой подход не может обнаружить двойной ошибки. Однако нечетное число ошибок обнаружить можно.

Подсчет числа 1 и выбор четного их числа эквивалентен использованию *арифметики по модулю 2.* Модуль 2 означает, что каждое число делится на 2 (основание) и заменяется остатком. В этой арифметике (в которой последовательный счет идет так: 0,1,0,1,...) производится подсчет по модулю 2 числа единиц в первых (n-1) позициях сообщения, и затем результат помещается в n-ю позицию. Таким образом, среди n посылаемых символов содержится четное число единиц.

Обычно длинные сообщения, состоящие из двоичных символов, разбиваются на отрезки (блоки) по (n—1) символов в каждом, и j к каждому отрезку добавляется по одному символу; таким образом, получается передаваемый блок длиной *n.* При необходимости последний блок дополняется нулями. Этот метод приводит к избыточности, равной *n/ (n—1)=1+1/ (n—1).* Избыточность определяется как отношение числа используемых двоичных символов минимально необходимому числу символов. *Чистая избыточность* равна 1/(n—1).

Ясно, что для получения низкой избыточности следует применять длинные сообщения. Однако для получения высокой надежности предпочтительнее короткие сообщения. Таким образом, выбор длины

*n*посылаемых блоков сообщения является результатом компромисса между двумя противоположными силами.

Простые коды для обнаружения пакетов ошибок

Ошибки (шум) в принятом сообщении чаще группируются в пакеты, а не в отдельные изолированные позиции. Вспышки молнии, флуктуации питающего напряжения, дефекты магнитной поверхности являются типичными причинами, приводящими к возникновению пакетов ошибок.

Предположим, экспериментально получена максимальная длина L пакета ошибок, который должен быть обнаружен. Для простоты примем, что длина пакета L совпадает с длиной передаваемого слова. В этих условиях нужно выбрать подходящий код с обнаружением ошибок (или код с исправлением ошибок), и вместо вычисления проверок на четность по битам следует вычислить проверку на четность по словам. В самом деле, работа со словами, а не с битами приводит к L независимым (перемежающимся) кодам, по одному для каждого двоичного символа в слове.

Если пакет перекрывает конец одного слова и начало другого, то по-прежнему ни в каком коде не встретятся две ошибки, поскольку, *по* предположению, длина k пакета удовлетворяет условию $0 \leq k \leq L$.

Таким образом, можно передавать сообщения по каналу с пакетами ошибок, если есть возможность определить конфигурацию щумов и соответственно организовать проверки.

Практическая часть

Пример 1.Сформировать отправляемое сообщение кодом четности.
Исходные сообщения
а) 00101101
б)01001100
Решение.
а) Сложим значения битов исходного сообщения по модулю 2.
$0+0+1+0+1+1+0+1 = 4 \pmod 2 = 0$
Таким образом, в конец сообщения добавиться бит равный 0:
001011010
б) Сложим значения битов исходного сообщения по модулю 2.
$0+1+0+0+1+1+0+0 = 3 \pmod 2 = 1$
Следовательно, в конец сообщения добавиться бит равный 1:
010011001

*Ответ:*Закодированные сообщения 001011010, 010011001.

Пример 2.Проверить принятое сообщение на наличие ошибки.
Принятые сообщения
а) 01001011011011
б) 11010100110100

Решение.

а) Сложим значения битов исходного сообщения.

0+1+0+0+1+0+1+1+0+1+1+0+1+1= 8

Сумма получилась четной, следовательно, в принятом сообщении ошибка отсутствует.

б) Сложим значения битов исходного сообщения.

1+1+0+1+0+1+0+0+1+1+0+1+0+0= 7

Сумма получилась нечетной, следовательно, в принятом сообщении ошибка присутствует.

Ответ: в принятом сообщении 01001011011011 ошибка отсутствует,
в принятом сообщении 1101010011010 ошибка присутствует.

Пример 3. Необходимо отправить сообщение Program, закодировав кодом ACSII и используя для проверки код четности.

Решение.

1) Закодируем сообщение кодом ACSII, воспользовавшись таблицей (приложение).

$P=120_8= 001\ 010\ 000_2$

$r=162_8= 001\ 110\ 010_2$

$o=157_8= 001\ 101\ 111_2$

$g=147_8= 001\ 100\ 111_2$

$r=162_8= 001\ 110\ 010_2$

$a=141_8= 001\ 100\ 001_2$

$m=155_8= 001\ 101\ 101_2$

Сообщение будет иметь вид

00101000000111001000110111100110011100111001000110000100110110 1

2)Разобьем его на блоки по 7 символов

0010100 0000111 0010001 1011110 0110011 1001110 0100011 0000100 1101101

Добавим в каждый блок по 1 символу – проверка на четность, получим закодированное сообщение для передачи.

00101000 00001111 00100010 10111101 01100110 10011100 01000111 00001001 11011011

Ответ: сообщение Program, закодированное для передачи, имеет вид

00101000 00001111 00100010 10111101 01100110 10011100 01000111 00001001 11011011

Пример 4. Сообщение Fall 1980 нужно закодировать кодом ASCII дляисправления пакетов.

Решение.

1) Закодируем сообщение кодом ACSII, воспользовавшись таблицей (приложение).Здесь проверки на четность не используются.

F = 106 = 01	000	110		1 = 061 = 00	110	001
a = 141 = 01	100	001		9 = 071 = 00	111	001
l = 154 = 01	101	100		8 = 070 = 00	111	000
l = 154 = 01	101	100		0 = 060 = 00	110	000
sp = 040 = 00	100	000				

2) Контрольная сумма сообщения получается поразрядным сложением битов всех символов, входящих в данный разряд, по модулю 2, и равна

$$00\ 000\ 111_2 = 007_8 = \text{BEL}.$$

Таким образом, закодированное сообщение имеет вид Fall 1980 BEL, где BEL — символ кода ASCII для $00\ 000\ 111_2$.

*Ответ:*закодированное сообщение имеет вид Fall 1980 BEL

Задания для самостоятельной работы

1Сформировать отправляемое сообщение кодом четности.

а) 10111001
б) 01101011
в) 01101001
г) 10011101

2 Проверить принятое сообщение на наличие ошибки.

а) 10101110110001
б) 00101100100100
в) 01101001010001
г) 10101100011101

3 Необходимо отправить сообщение, закодировав кодом ACSII и используя для проверки код четности.

а) Access
б) Backup
в) Cancel
г) Device

4 Сообщения закодировать кодом ASCII дляисправления пакетов.

а) Centralized database
б) Computer test
в) Digital signal
г) Functionblock

Тема 2.2. Перевод числовой информации из одной позиционной системы в другую

Практическое занятие: «Осуществление перевода чисел из одной системы счисления в другую»

Теоретическая часть

Система счисления — совокупность приемов и правил для записи чисел цифровыми знаками или символами.

Все системы представления чисел делят на позиционные и непозиционные. Самый простой способ записи чисел может быть описан выражением

$$A_D = D_1 + D_2 + \cdots + D_K = \sum_{t=1}^{t=k} D_1$$

где A — запись числа A в системе счисления D; D_1 — символы системы, образующие базу $D = \{D_1, D_2,...,D_k\}$.

По этому принципу построены непозиционные системы счисления.

Непозиционная система счисления — система, для которой значение символа не зависит от его положения в числе.

В общем случае системы счисления можно построить по следующему принципу:

$$A_B = a_1 B_1 + a_2 B_2 + \cdots + a_n B_n \qquad (1)$$

где A_B —запись числа A в системе счисления с основанием B_i; a_i — цифра (символ) системы счисления с основанием B_i; B_i — база, или основание системы.

Если предположить, что $B_i = q^i$, где q – основание системы счисления, то с учетом (1)

$$B_i = q \ \ B_{i-1} \qquad (2)$$

Позиционная система счисления — система, удовлетворяющая равенству (2). Естественная позиционная система счисления имеет место, если q - целое положительное число.

Для позиционной системы счисления справедливо равенство

$$A_q = \sum_{i=-m}^{i=n} a_i q^i \qquad (3)$$

Или $A_q = a_n q^n + \cdots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \cdots + a_{-m} q^{-m}$, где

A_q —произвольное число, записанное в системе счисления с основанием q;

n + 1, m — количество целых и дробных разрядов.

В процессе преобразования информации в цифровом автомате возникает необходимость перевода чисел из одной позиционной системы счисления в другую. Это обусловлено тем, что в качестве внутреннего алфавита наиболее целесообразно использовать двоичный алфавит с символами 0 и 1.

Дляреализации машинных алгоритмов перевода применяются следующие методы.

Перевод целых чисел делением на основание q_2 новой системы счисления. Целое число A_{q2} в системе с основанием q_2 записывается в виде

$$A_{q2} = b_k q_2^k + b_{k-1} q_2^{k-1} + \cdots b_1 q_2^1 + b_0 q_2^0$$

Переписав это выражение по схеме Горнера, получим

$$A_{q2} = \left(\ldots\left((b_k q_2 + b_{k-1})q_2 + \cdots\right)q_2 + b_1\right)q_2 + b_0 \qquad (4)$$

Правую часть выражения (4) разделим на величину основания q_2. В результате определим первый остаток b_0 и целую часть $\left(\ldots\left((b_k q_2 + b_{k-1})q_2 + \cdots\right)q_2 + b_1\right)$. Разделив целую часть на q_2, найдем второй остаток b_1. Повторяя процесс деления к +1 раз, получим последнее целое частное b_k, которое, по условию, меньше основания системы q_2 является старшей цифрой числа, представленного в системе с основанием q_2.

Рассмотрим методы перевода дробной части числа

$$A_{q1} = a_{-1} q_1^{-1} + \cdots + a_{-m} q_1^{-m}$$

В новой системе с основанием q_2это число будет изображено как 0, $b_{-1}, \ldots, b_i$, или

$$A_{q2} = b_{-1} q_2^{-1} + \cdots + b_{-x} q_2^{-x}$$

Переписав это выражение по схеме Горнера, получим

$$A_{q2} = q_2^{-1}(b_{-1} + q_2^{-1}(b_{-2} + \cdots q_2^{-1} b_{-i})) \quad (5)$$

Если правую часть выражения (5) умножить на q_2, то получится новая неправильная дробь, в целой части которой будет число b_{-1}. Умножив затем оставшуюся дробную часть на величину основания q_2, получим дробь, в целой части которой будет b_{-2}, и т. д. Повторяя процесс умножения s раз, найдем все s цифр числа в новой системе счисления. При этом все действия должны выполняться по правилам q_1 -арифметики, и следовательно, в целой части получающихся дробей будут проявляться эквиваленты цифр новой системы счисления, записанные в исходной системе счисления.

При переводе правильных дробей из одной системы счисления в другую можно получить дробь в виде бесконечного или расходящегося ряда. Процесс перевода можно закончить, если появится дробная часть, имеющая во всех разрядах нули, или будет достигнута заданная точность перевода (получено требуемое количество разрядов результата). Последнее означает, что при переводе дроби необходимо указать количество разрядов числа в новой системе счисления. Естественно, что при этом возникает погрешность перевода чисел, которую надо оценивать.

Для перевода неправильных дробей из одной системы счисления в другую необходим раздельный перевод целой и дробной частей по правилам, описанным выше. Полученные результаты записывают в виде новой дроби в системе с основанием q_2.

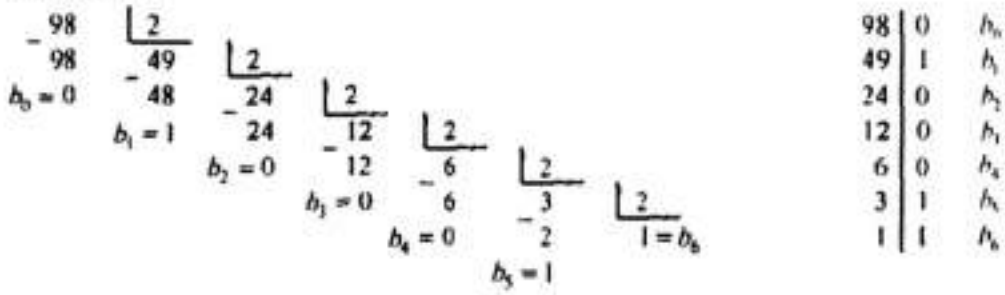
Использование промежуточной системы счисления. Этот метод применяют при переводе из десятичной системы в двоичную и наоборот. В качестве промежуточных систем счисления целесообразно использовать системы с основанием $q = 2^k$. При этом существенно упрощается преобразование информации из системы счисления с основанием $q = 2^k$ в двоичную систему и наоборот. Преобразование фактически сводится к тому,

что символы первоначальной информации, заданной в системе с основанием $q = 2^k$, заменяются соответствующими двоичными эквивалентами. Представление десятичных чисел в таком виде называется десятично-двоичным. Обратное преобразование из двоичной системы в систему с основанием $q = 2^k$ сводится к тому, что двоичный код разбивается на группы по k двоичных разрядов в каждой (начиная от младших разрядов для целых чисел или с первого разряда после запятой для правильных дробей); эти группы (диады, триады, тетрадыи т. д.) заменяются соответствующими символами исходной системы счисления.

Практическая часть

Пример 1. Перевести десятичное число$A = 98$ в двоичную систему счисления ($q_2 = 2$).

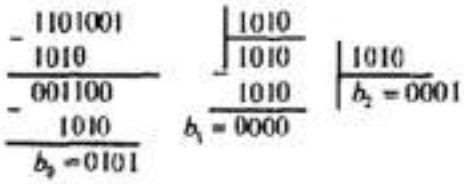
Решение.



Ответ: $A_2 = 1100010_2$

Пример 2.Перевести двоичное число $A_2 = 1101001_2$в десятичную систему счисления ($q_2 = 10$). Основание q_2изображается в двоичной системе эквивалентом $q_2 = 1010_2$.

Решение.



$b_0 = 0101_2 = 5_{10}$; $b_1 = 0000_2 = 0_{10}$; $b_2 = 0001_2 = 1_{10}$; $A_{10} = 105$
Ответ: $A_{10} = 105_{10}$

Пример 3.Перевести десятичную дробь$A = 0,625$ в двоичную систему счисления ($q_2 = 2$).

Решение.

0,	×	625 2
$b_{-1} = 1$,	×	250 2
$b_{-2} = 0$,	×	500 2
$b_{-3} = 1$,	×	000 2
$b_{-4} = 0$,		000

Ответ: $A_2 = 0{,}1010_2$

Пример 4.Перевести двоичную дробь $A_2 = 0{,}1101_2$в десятичную систему счисления ($q_2 = 1010_2$).

Решение.

	0,	×	1101 1010
$b_{-1} = 8$	1000,	×	0010 1010
$b_{-2} = 1$	0001,	×	0100 1010
$b_{-3} = 2$	0010,	×	1000 1010
$b_{-4} = 5$	0101.		0000

Ответ: $A_{10} = 0{,}8125_{10}$

Пример 5.Перевести десятичную дробь$A = 98{,}625$ в двоичную систему счисления ($q_2 = 2$).

Решение.

Результаты перевода целой и дробной части возьмем из примеров 1 и 3.

Ответ: $A_2 = 1100010{,}1010_2$

Пример 6. Перевести десятичное число$A = 121$ в двоичную систему счисления, используя в качестве промежуточной восьмеричную систему счисления ($q_2 = 8$).

Решение.

$q_2 = 8$

121	1
15	7
1	1

3 шага

$q_2 = 2$

121	1
60	0
30	0
15	1
7	1
3	1
1	1

7 шагов

Ответ: $A = 121 = 171_8 = 1111001_2$

Задания для самостоятельной работы

1. Перевести десятичное число A = 153 в двоичную систему счисления.
2. Перевести двоичное число A = 100010101011,01 в десятичную систему счисления.
3. Перевести десятичное число A = 135,656 в двоичную систему счисления с точностью до пяти знаков после запятой.
4. Перевести двоичное число A_2=10111011 в десятичную систему счисления методом деления на основание,
5. Перевести восьмеричное число A_8 =345,766 в двоичную систему счисления.
6. Записать десятичное число A = 79,346 в двоично-десятичной форме.
7. Перевести восьмеричное число A_8 = 326 в троичную систему счисления.
8. Перевести восьмеричное число A_8 =15,647 в двоичную систему счисления.
9. Перевести троичное число A_3 =1211 в пятеричную систему счисления.
10. Какое из чисел больше 5_{10} или 10_5; 1000_2 или 10_8.
11. Существуют ли системы счисления с основаниями Ри Q, в которых $12_P > 21_Q$?

Тема 2.4 Системы счисления с отрицательным основанием
Практическое занятие: «Представление чисел в системе счисления с отрицательным основанием»
Теоретическая часть

Двоичную систему счисления с основанием $q = -2$, в которой используются символы *0, 1,* назовем минус-двоичной системой счисления. В этой системе можно представлять как положительные, так и отрицательные числа (таблица 2).

Таблица 2

Десятичное число	Двоичный эквивалент для q = -2	Десятичное число	Двоичный эквивалент для q = -2
0	000000	-10	001010
1	000001	+15	010011
-1	000011	-15	110001
+5	000101	21	010101
-5	001111	-21	111111
+10	011110		

Из таблицы видно, что методы перевода целых десятичных чисел в систему счисления с основанием $q= -2$ аналогичны методам перевода десятичных чисел в систему счисления с положительным основанием. Но при переводе десятичных чисел необходимо учитывать следующее: при использовании метода последовательного деления на основание новой системы все остатки от деления на каждом шаге должны быть положительными числами, которые не превышают абсолютного значения нового основания q. Это правило распространяется и на случай перевода правильных дробей методом последовательного умножения на основу q, где появляющиеся целые части дробей также должны быть положительными числами, значение которых меньше величины q.

В случае перевода правильной дроби (или дробной части смешанной дроби) необходимо, чтобы дробь на каждом шаге удовлетворяла требованию.

$$\frac{|q|}{|q|+1} \le A \le 1/(|q|+1), \text{или} -2/3 \le A \le 1/3, \qquad (6)$$

гдеA– дробь, переводимая в систему счисления с основанием $q = -2$. Если ограничение *(1)* не выполняется, то дробь А представляют и виде

$$A = 1 - \Delta,$$

где Δ должно лежать в указанных пределах.

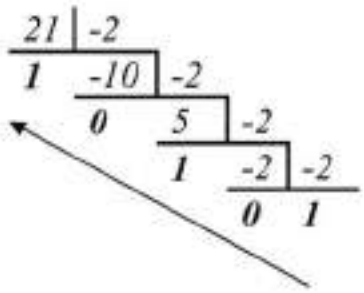
Практическая часть

Пример 1. Перевести десятичное число$A = 21$ в минус-двоичную систему счисления методом деления на основание системы ($q= -2$).

Решение

21	1
-10	0
5	1
-2	0
1	1

или



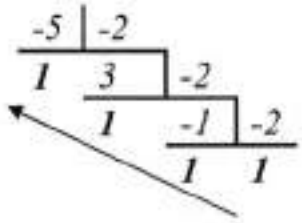
Ответ: $A = 21_{10} = 10101_{-2}$

Пример 2. Перевести десятичное число $A = -5$ в минус-двоичную систему счисления методом деления на основание системы ($q = -2$).

Решение

-5	1
3	1
-1	1
1	1

или



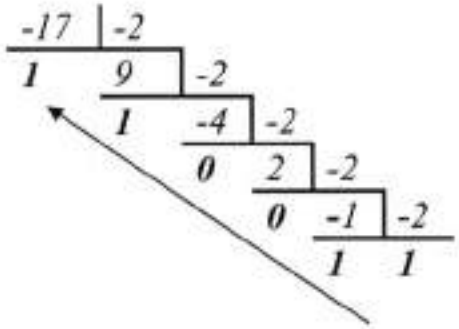
Ответ: $A = -5_{10} = 1111_{-2}$

Пример 3. Перевести десятичное число $A = -17$ в минус-двоичную систему счисления методом деления на основание системы ($q = -2$).

Решение

-17	1
9	1
-4	0
2	0
-1	1
1	1

или



Ответ: $A = -17_{10} = 110011_{-2}$

Пример 4. Перевести десятичную дробь $A = 0,625$ в минус-двоичную систему счисления с точностью до трех знаков после запятой.

Решение. Исходная дробь не удовлетворяет неравенству (6). Преобразуем ее к виду $0,625 = 1 - 0,375$

(1) $-0,375$

x

-2

$0,750$ неравенство (6) не удовлетворяется

(1) $-0,250$

x

-2

$0,500$ неравенство (6) не удовлетворяется

(1) *– 0,500*

x

-2

1,000 неравенство (6) удовлетворяется

Ответ: A = 0,625 = 1,111

Пример 5 . Перевести десятичную дробь*A = -0,125* в минус-двоичнуюсистему счисления с точностью до трех знаков после запятой.

Решение. Исходная дробьудовлетворяет неравенству (6).

(0) *– 0,125*

x

-2

(0) *0,250* неравенство (6) удовлетворяется

x

-2

– 0,500 неравенство (6) не удовлетворяется

(1) *– 0,500*

x

-2

1,000 неравенство (6) удовлетворяется

Ответ: A = 0,125 = 0,011

Пример 6 .

Решение. Исходная дробьудовлетворяет неравенству (6).

(0) *0,125*

x

-2

(0) *- 0,250* неравенство (6) удовлетворяется

x

-2

(0) *- 0,500* неравенство (6) не удовлетворяется

x

-2

1,000 неравенство (6) удовлетворяется

Ответ: A = - 0,125 = 0,001

Задания для самостоятельной работы

1 Представить расчет перевода десятичных чисел таблицы, расположенной в теоретической части практического занятия, в минус-двоичную систему счисления методом деления на основание системы (q= -2).

2. Перевести десятичное число в минус-двоичную систему счисления методом деления на основание системы ($q = -2$).

а) 31
б) -26
в) 25
г) -35

3 Перевести десятичную дробь в минус-двоичную систему счисления с точностью до трех знаков после запятой.

а) 0,145
б) -0,415
в) 0,642
г) 0,255

Тема 2.6Представление отрицательных чисел
Практическое занятие: «Выполнение операции вычитания»
Теоретическая часть

Один из способов выполнения операции вычитания с помощью сумматора — замена знака вычитаемого на противоположный и прибавление его к уменьшаемому:

$$A\text{-}B = A + (\text{-}B). \quad (7)$$

Этим операцию арифметического вычитания заменяют операцией алгебраического сложения, которая и становится основной операцией сумматора. Возникает вопрос: как представлять отрицательные числа в цифровом автомате?

Для машинного представления отрицательных чисел используют прямой, дополнительный и обратный коды.

Прямой код числи A = - 0, $a_1,a_2 \ldots a_n$ — машинное изображение этого числа в виде $[A]_{пр} = 1,a_1,a_2 \ldots a_n$.

Из определения следует, что в прямом коде все цифровые разряды отрицательного числа остаются неизменными, а в знаковой части записывается единица.

Правила преобразования чисел в прямой код можно сформулировать так:

$$[A]_{пр} = \begin{cases} A, \text{если } A \geq 0, \\ 1 + |A|, \text{если } A \leq 0. \end{cases} \quad (8)$$

Дополнительный код числа A = $-0,a_1a_2\ldots a_n$ — такое машинное изображение этого числа $[A]_д = 1, \bar{a}_1\bar{a}_2\ldots \bar{a}_n$для которого $\bar{a}_1 = 0$ при $a_1 = 1$, и $\bar{a}_1 = 1$ при $\bar{a}_k = 1$, за исключениемпоследнего значащего разряда, для которого $\bar{a}_k = 1$ при $a_k = 1$.

Так как положительные числа не меняют своего изображения в дополнительном коде, то правила преобразования в дополнительный код можно написать следующим образом:

$$[A]_д = \begin{cases} A, \text{если } A \geq 0, \\ q - |A|, \text{если } A < 0. \end{cases} \quad (9)$$

Обратный код числа A = -0, $a_1,a_2\ldots a_n$— такое машинное изображение этого числа $[A]_{об} = 1, \overline{a_1}\overline{a_2} \ldots \overline{a_n}$для которого$\overline{a_1} = 0$, если, $a_1=1$, и $\overline{a_1} = 1$, если $a_1 = 0$.

Из определения следует, что обратный код двоичного числа является инверсным изображением самого числа, в котором все разряды исходного числа принимают инверсное (обратное) значение, т.е. все нули заменяются на единицы, а все единицы — на нули.

Правила преобразования чисел в обратный код можно сформулировать следующим образом;

$$[A]_{об} = \begin{cases} A, \text{если } A \geq 0 \\ q - q^{-n} + A, \text{если } A \leq 0 \end{cases} \quad (10)$$

Для получения дополнительного кода отрицательных чисел можно воспользоваться следующим образом: сначала инвертируется цифровая часть исходного числа, в результате получается его обратный код; затем добавляется единица в младший разряд цифровой части числа и тем самым получается дополнительный код этого изображения.

Практическая часть

Пример 1 **.** Представить число А = -0,101110 в прямом коде

Решение.

Используя определение прямого кода, если А = -0,101110, то $[A]_{пр} = 1,101110$.

Ответ: $[A]_{пр} = 1,101110$

Пример 2 **.** Представить число А = -0,101110 в дополнительном коде

Решение.

Используя определение дополнительного кода число А = -0,101110 запишется в дополнительном коде так:

$[A]_{д} = 1,010010$.

Ответ: $[A]_{д} = 1,010010$

Пример 3 **.** Представить число А = -0,101110 в обратном коде

Решение.

Используя определение обратного кода для числа А = - 0,101110, получим$[A]_{об} = 1,010001$.

Ответ: $[A]_{об} = 1,010001$

Пример 4 Найти обратный и дополнительный коды числа $A = -0,111000_2$

Решение.

Используя определение обратного кода, получим $[A]_{об} = 1,000111$.

Для нахождения дополнительного числа добавим единицу вмладшийразряд его изображения

$$\begin{array}{r} 1,000111 \\ + \\ \underline{\qquad\qquad 1} \\ [A]_{д} = 1,001000 \end{array}$$

Ответ: $[A]_{об} = 1,000111$; $[A]_{д} = 1,001000$

Пример 5Выполнить операциювычитания в двоичной системе счисления*0,101001 – 0, 001111*

Решение.

$$\begin{array}{r} \dot{0},101\dot{0}01 \\ - \\ \underline{0,001111} \\ 0,011010 \end{array}$$

Ответ:0,011010

Задания для самостоятельной работы

1 Представить числа в прямом, обратном и дополнительном кодах

а) A = 0,1101101

б) A = - 0,001100

в) A = 0,0011010010

г) A = - 0,1100100100

д) A = -0,01010100101

е) A = -0,111001100

2 Выполнить операцию вычмтания в двоичной системе счисления

а) 0,101001 – 0, 001111

б) 0,0101110 – 0, 0101001

в) 0,11001101 – 0,10001101

г) 0,00010011 – 0,00001101

д) 0,1011000 – 0,0110101

е) 0,0001010 – 0,0000111

Тема 3.2Сложение чисел, представленных в форме с фиксированной запятой, на двоичных сумматорах

Практическое занятие: «Выполнение операции сложения»

Теоретическая часть

Двоичная система счисления определяется простотой выполнения арифметических действий:

сложение	вычитание	умножение
0+ 0 = 0	0 – 0= 0	0 × 0 = 0
0 + 1 = 1	1 - 0= 1	0 × 1 = 0
1 + 0 =1	1–1 =0	1 × 0 = 0
1 + 1 =(1)0	0 - 1=(1) 1	1 × 1 = 1
перенос в старший разряд	заем в старшем разряде	

В основу арифметико-логического устройства любого компьютера может быть положен либо сумматор, либо вычитатель.

Рассмотрим формальные правила выполнения арифметических операций сложения и вычитания на уровне разрядов операндов.

На основе правил двоичной арифметики можно записать правила сложения двоичных цифр так, как показано втаблице 3, где a_i, b_i — разряды операндов А и В соответственно; c_i— разряд суммы; n_i — перенос из данного разряда в соседний старший.

Двоичный полусумматор — устройство, выполняющее арифметические действия по правилам, указанным в таблице 3.

Таблица 3

a_i	b_i	c_i	n_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Таблица 4

a_i	b_i	n_{i-1}	c_i	n_i
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Появление единицы переноса при сложении двух разрядов несколько изменяет правила сложения двоичных цифр (таблица 4). Обобщая вышеизложенное, можно сформулировать правила поразрядных действий при сложении операндов А и В:

$$a_i + b_i + n_{i-1} = c_i + n_i \qquad (3.3)$$

где n_{i-1} — перенос из $(i - 1)$-го разряда; n_i — перенос в $(i+ 1)$-й разряд (переносы принимают значения 0 или 1).

Двоичный сумматор — устройство, выполняющее арифметические действия по правилам, указанным в таблице 3.2. Условные обозначения двоичных полусумматоров и сумматоров показаны на рисунке1, а и б соответственно.

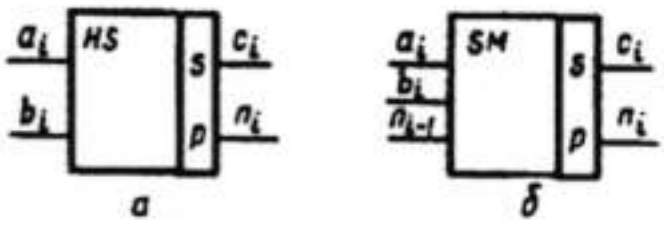


Рисунок 1 – Условное обозначение полусумматора и сумматора

Двоичный сумматор прямого кода (ДСПК) — сумматор, в котором отсутствует цепь, поразрядного переноса между старшим цифровым и знаковым разрядами (рис. 2, а). На ДСПК можно складывать только числа, имеющие одинаковые знаки, т. е. такой сумматор не может выполнять операцию алгебраического сложения. Пусть заданы операнды

$$[A]_{\text{пр}} = Sg_A a_1 a_2 \ldots a_n \quad [B]_{\text{пр}} = Sg_B b_1 b_2 \ldots b_n,$$

где Sg_A, Sg_Bсоответственно содержимое знаковых разрядов изображений дляA и B (символ происходит от английского слова sign — знак); a_i, b_i— цифровые разряды изображений.

Если $Sg_A = Sg_B$, то сумма чисел будет иметь знак любого из слагаемых, а цифровая часть результата получится после сложения цифровых частей операндов.

При сложении чисел на ДСПК возможен случай, когда абсолютное значение суммы операндов превышает единицу. Тогда имеет место переполнение разрядной сетки автомата. Признак переполнения — наличие единицы переноса из старшего разряда цифровой части сумматора. В этом случае должен вырабатываться сигнал переполнения $\varphi = 1$, по которому происходит автоматический останов машины и корректировка масштабных коэффициентов с таким расчетом, чтобы избежать появления переполнения.

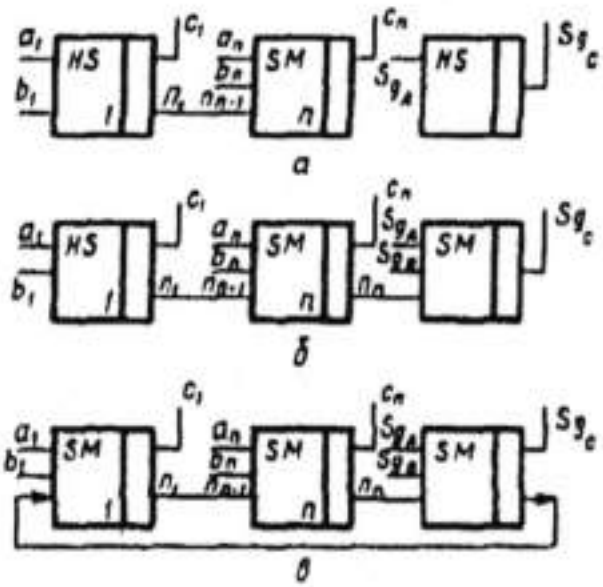


Рисунок 2 – Структурные схемы двоичных сумматоров на n разрядов

Двоичный сумматор дополнительного кода (ДСДК) — сумматор, оперирующий изображениями чисел в дополнительном коде. Характерная особенность ДСДК — наличие цепи поразрядного переноса из старшего разряда цифровой части в знаковый

разряд (рисунок 2, б).

Правила сложения чисел на ДСДК определяются теоремой.

Теорема. *Сумма дополнительных кодов чисел есть дополнительный код результата.*

Теорема справедлива для всех случаев, в которых не возникает переполнение разрядной сетки, что позволяет складывать автоматные изображения чисел по правилам двоичной арифметики (таблица4), не разделяя знаковую и цифровую части изображений.

Двоичный сумматор обратного кода (ДСОК) — сумматор, оперирующий изображениями чисел в обратном коде. Характерная особенность ДСОК — наличие цепи кругового, или циклического, переноса нз знакового разряда в младший разряд цифровой части (рисунок2, в). Правила сложения чисел на ДСОК определяются теоремой.

Теорема.*Сумма обратных кодов чисел есть обратный код результата.*

На ДСОК машинные изображения чисел также складываются по правилам, приведенным в таблице 4.

Практическая часть

Пример 1. Сложить числаA = 0,1001, B = 0,0101 на сумматоре прямого кода.

Решение.

$$\begin{array}{ll} [A]_{пр} = 0{,}1001 & Sg_A = 0 \\ + & \\ \underline{[B]_{пр} = 0{,}0101} & \underline{Sg_B = 0} \\ [C]_{пр} = 0{,}1110 & Sg_C = 0 \end{array}$$

Ответ:$[C]_{пр} = 0{,}1110$

Пример 2. Сложить числаA = – 0,0011, B = – 0,1001 на сумматоре прямого кода.

Решение.

$$\begin{array}{lll} [A]_{пр} = 1{,}0011 & Sg_A = 1 & ,0011 \\ & & + \\ [B]_{пр} = 1{,}1001 & \underline{Sg_B = 1} & \underline{,1001} \\ & Sg_C = 1 & ,1100 \end{array}$$

Ответ:$[C]_{пр} = 1{,}1100$

Пример 3.3. Найти сумму чиселA = 0,1001, B = 0,0100, используя сумматор дополнительного кода.

Решение.

$$[A]_д = 0{,}1001$$
$$+$$
$$\underline{[B]_д = 0{,}0100}$$
$$[C]_д = 0{,}1101$$

Ответ: $C = 0{,}1101$

Пример 4. Найти сумму чисел $A = -0{,}0011$, $B = 0{,}0110$ на сумматоре дополнительного кода.

Решение.

$$[A]_д = 1{,}1101$$
$$+$$
$$\underline{[B]_д = 0{,}0110}$$
$$[C]_д = 0{,}0011$$

Ответ: $C = 0{,}0011$

Пример 5. Найти сумму чисел $A = 0{,}1011$, $B = -0{,}0101$ на сумматоре дополнительного кода.

Решение.

$$[A]_д = 0{,}1011$$
$$+$$
$$\underline{[B]_д = 1{,}1011}$$
$$[C]_д = 0{,}0110$$

Ответ: $C = 0{,}0110$

Пример 6. Найти сумму чисел $A = 0{,}0101$, $B = 0{,}1001$, используя сумматоробратного кода.

Решение.

$$[A]_{об} = 0{,}0101$$
$$+$$
$$\underline{[B]_{об} = 0{,}1001}$$
$$[C]_{об} = 0{,}1110$$

Ответ: $C = 0{,}1110$

Пример 7. Найти сумму чисел $A = -0{,}0111$, $B = 0{,}1101$, используяДСОК.

Решение.

$$[A]_{об} = 1{,}1000$$
$$+$$
$$\underline{[B]_{об} = 0{,}1101}$$
$$0{,}0101$$
$$+\underline{1}$$
$$[C]_{об} = 0{,}0110$$

Ответ: $C = 0{,}0110$

Пример 8. Найти сумму чиселA = 0,0101, B = – 0,0110, используяДСОК.

Решение.

$$[A]_{об} = 0{,}0101$$
$$+$$
$$\underline{[B]_{об} = 1{,}1001}$$
$$[C]_{об} = 1{,}1110$$

Ответ: C = – 0,0001

Пример 9. Найти сумму чиселA = – 0,1001, B = – 0,0101, используяДСОК.

Решение.

$$[A]_{об} = 1{,}0110$$
$$+$$
$$\underline{[B]_{об} = 1{,}1010}$$
$$1{,}0000$$
$$+\underline{\ldots..1}$$
$$[C]_{об} = 1{,}0001$$

Ответ: C = – 0,1110

Задания для самостоятельной работы

1.Сложить числаA иB,используя двоичный сумматор прямого кода:

а) A = 0,10110, B = 0,01100

б)A = - 0,001011, B = - 0,100101

в)A = 0,01101, B = - 0,10101

г)A = 0,0100110, B = 0,0010101

д)A = - 0,0101001, B = - 0,0010011

2.Сложить числаA иB, используя двоичный сумматордополнительного кода:

а) A = 0,1101101, B = - 0,1010011

б)A = - 0,0101100, B = - 0,1001100

в)A = - 0,1110110, B = 0,1001101

г)A = - 0,1011011, B = - 0,1101110

д)A = - 0,0101100, B = - 0,1000101

3.Сложить числаA иB, используя двоичный сумматоробратного кода:

а) A = 0,1101101, B = - 0,1001001

б)A = - 0,1101101, B = - 0,1010101

в)A = - 0,0101100, B = 0,1101011

г)A = - 0,1001101, B = - 0,0010111

д)A = 0,1011010, B = - 0,1000110

Тема 3.4. Особенности сложения чисел, представленных в форме с плавающей запятой

Практическое занятие: «Выполнений операции сложения и нормализации чисел, представленных в форме с плавающей запятой»

Теоретическая часть

Числа, представленные в форме с плавающей запятой, изображаются двумя частями — мантиссой и порядком. При операции алгебраического сложения действия, выполняемые над мантиссами и порядками, различны. Следовательно, в цифровом автомате должны быть два раздельных устройства для обработки мантисс и для обработки порядков.

Так как для чисел с плавающей запятой справедливо условие

$$q^{-1} \le |m_a| < 1, \qquad (1)$$

где –qоснование системы счисления, m_a – мантисса числа, то всякий результат, не удовлетворяющий этому условию, должен быть приведен в соответствие с формулой (1).

Такую операцию называют *нормализацией числа.* Операция нормализации числа состоит из проверки выполнимости условия (1) и сдвига изображения мантиссы в ту или иную сторону. Сдвиги могут осуществляться на один разряд и более в левую или правую сторону в пределах разрядной сетки машины.

Простой сдвиг — операция, выполняемая по следующим правилам:

Исходная комбинация	Сдвинутая влево на один разряд	Сдвинутая вправо на один разряд
$0,a_1a_2 \dots a_n$	$a_1, a_2 \dots a_n 0$	$0,0a_1 \dots a_{n-1}$
$1,a_1a_2 \dots a_n$	$a_1, a_2 \dots a_n \alpha^1$	$0,1a_1 \dots a_{n-1}$

Модифицированный сдвиг — операция над модифицированными изображениями, выполняемая следующим образом:

Исходная комбинация	Сдвинутая влево на один разряд	Сдвинутая вправо на один разряд
$00,a_1a_2 \dots a_n$	$0a_1, a_2 \dots a_n 0$	$00,0a_1 \dots a_{n-1}$
$01,a_1a_2 \dots a_n$	$1a_1, a_2 \dots a_n 0$	$00,1a_1 \dots a_{n-1}$
$10,a_1a_2 \dots a_n$	$0a_1, a_2 \dots a_n \alpha^*$	$1,0a_1 \dots a_{n-1}$
$11,a_1a_2 \dots a_n$	$1a_1, a_2 \dots a_n \alpha^*$	$1,1a_1 \dots a_{n-1}$

Нарушение нормализации числа— невыполнение условия (1). Так как условие(1) содержит два неравенства, то может быть нарушение справа и слева. Признак нарушения нормализации числа справа γ (когда величина результата равна или превышает единицу) — наличие разноименных комбинаций в знаковых разрядах сумматора, т. е.

$$\gamma=1, \text{ если } Sg_1 \wedge \overline{Sg_2} = 1 : \overline{Sg_1} \wedge Sg_2 = 1 \qquad (2)$$

---

*Величина α зависит от кода: для дополнительного $\alpha = 0$, для обратного $\alpha = 1$

(в остальных случаях γ =0), где γ — признак нарушения нормализации числа справа, указывающий на необходимость сдвига числа вправо на один разряд.

Признак нарушения нормализации числа слева σ(когда результат по собственной величине оказывается меньше*1/q)* — наличие одинаковых комбинаций в разряде переполнения и старшем разряде цифровой части сумматора (p_1):

$$\delta=1, \text{ если } Sg_2 \wedge p_1=1;\ \overline{Sg_2} \wedge \overline{p_1}=1 \qquad (3)$$

(в остальных случаях $\delta = 0$), где δ — признак нарушения нормализации, указывающий на необходимость сдвига числа влево на один разряд.

Таким образом, операция нормализации числа состоит из совокупности сдвигов и проверки наличия признаков нарушения γ и δ.

В случае, если числа$A=m_A p_A$ и $B=m_B p_B$имеют одинаковый порядок$p_A = p_B$ и обе мантиссы удовлетворяют условию нормализации, то сложение мантисс осуществляетсянасоответствующем сумматоре по правилам, изложенным ранее для чисел, представленных в форме с фиксированной запятой. Если после сложения мантисса результата удовлетворяет условию нормализации (т.е. $\delta = 0$, γ =0), то к этому результату приписывается порядок любого из операндов. В противном случае происходит нормализация числа.

В случае сложения чисел, представленных в форме сплавающей занятой, когда их порядки не равны друг другу, т.е.$p_A \neq p_B$, необходимым условием является соответствие разрядов операндов друг другу. Значит, прежде всею нужно уравнять порядки, что, естественно, повлечет за собой временное нарушение нормализации одного из слагаемых. Выравнивание порядков означает, что порядок меньшего числа надо увеличить на величину $\Delta p = |p_A - p_B|$- что означает **сдвиг** мантиссы **меньшего** числа **вправона** количество разрядов, равное Δp.

Следовательно, цифровой автомат должен самостоятельно определять, какой из двух операндов меньший. На это укажет знак разности: $p_A - p_B$положительный знак будет при$p_A \geq p_B$, а отрицательный — при$p_A < p_B$.

Операции сложения и вычитания чисел в форме с плавающей запятой осуществляются во всех современных машинах по изложенным выше правилам.

Практическая часть

Пример 1.Найти сумму чисел A=$-0{,}1100 * 2^4$и $B = -0{,}1000 * 2^4$, если числа складываются на сумматоре обратного кода (шесть разрядов для мантиссы и четыре разряда для порядка)

*Решение.*Машинные изображения операндов записываются н следующем виде

$$[m_A]^{\text{м}}_{\text{об}} = 11{,}0011;\ [p_A]_{\text{об}} = [p_B]_{\text{об}} = 0{,}100; [m_B]^{\text{м}}_{\text{об}} = 11{,}0111.$$

Затем складываются мантиссы:

$$\begin{array}{r} 11{,}0011 \\ + \\ 11{,}0111 \\ \hline [m_C]^{м}_{об}=10{,}0111\ (\delta=0,\ \gamma=1). \end{array}$$

Здесь произошло нарушение нормализации справа и требуется модифицированный сдвиг мантиссы результата вправо на один разряд:

$$[m'_C]^{м}_{об}=11{,}0101\ (\delta=0,\ \gamma=0).$$

Одновременно со сдвигом проводится коррекция порядка результата на величину +0,001, или $[p^{i}_C]_{об} = 0{,}100 + 0{,}001 = 0{,}101$. В результате получается окончательный результат:

Ответ: C = -0,1010*2^5

Пример 2 Сложить числа *A*и *B*, заданные в форме с фиксированной точкой: m_A=100110;X_A=101; $X_B = -111001; X_B = 011$. Длявыполнения операции сложения использовать сумматор дополнительного кода, имеющий семь битов для мантиссы со знаком, четыре бита для характеристики со знаком.

Решение. Запишем машинные изображения мантисс:

$[m_A]^{м}_{д} = 00{,}100110; [m_B]^{м}_{д} = -11{,}000111;$

Исходные числа памяти машины можно хранить либо в прямом, либо в обратном (дополнительном) кодах.Если числа хранятся в памяти машины в прямом коде, то при выполнении операции сложения (вычитания) на сумматорах обратного (дополнительного) кода необходимо провести преобразование из прямого кола в обратный (дополнительный) код. По окончании операции должно проводиться преобразование результата из обратного (дополнительного) кода в прямой.

При выполнении данного примера предполагается, что числа в памяти машины хранятся в дополнительном коде.

Прежде всего необходимо сравнить характеристики:

$$\Delta X = [X_A]_д - [X_B]_д = 0{,}101 + 1{,}101 = 0{,}010.$$

Разность характеристик— положительная: второй порядок меньше первого на 2. Следовательно, мантисса второго числа сдвигается на два разряда (сдвиг модифицированный) и после этого мантиссы складываются:

$$\begin{array}{r} [\overline{m}_B]_д = 11{,}110001 \\ + \\ [m_B]_д=00{,}100110 \\ \hline [m_C]_д= 00{,}010111(\delta=1,\ \gamma=0). \end{array}$$

Так как δ=1, то проводится сдвиг влево на один разряд с коррекцией характеристик

$[m'_C]_д$ =00,101110 (δ=0, γ =0),$[X_C]_д$=0,101+1,111=0,100.

Таким образом. Окончательный результат получен в нормализованном виде.

*Ответ:*C= +101110,X_C=100.

Задания для самостоятельной работы

1. Найти сумму чисел Аи B, если числа складываются на сумматоре обратного кода (шесть разрядов для мантиссы и четыре разряда для порядка):

а) A=0,0101 $*$ 2^3и $B = -0{,}1101 * 2^3$

б)A=0,1011 $*$ 2^{-3}и $B = 0{,}1101 * 2^{-3}$

2. Найти сумму чисел Аи B, если числа складываются на сумматоре обратного кода (семь разрядов для мантиссы и четыре разряда для порядка):

а)A=0,10110 $*$ 2^{-3}и $B = -0{,}11011 * 2^{-4}$

б)A=0,11001 $*$ 2^3и $B = 0{,}10111 * 2^4$

3. Найти сумму чисел Аи B, если числа складываются на сумматоре обратного кода (восемь разрядов для мантиссы и четыре разряда для порядка):

а)A=0,101011 $*$ 2^{-2}и $B = -0{,}001010 * 2^{-4}$

4. Найти сумму чисел Аи B, если числа складываются на сумматоре дополнительного кода (шесть разрядов для мантиссы и четыре разряда для порядка):

а)A=0,1101 $*$ 2^{-4}и $B = -0{,}1001 * 2^{-4}$

5. Найти сумму чисел Аи B, если числа складываются на сумматоре дополнительного кода (восемь разрядов для мантиссы и четыре разряда для порядка):

а)A=0,011011 $*$ 2^{-3}и $B = -0{,}100110 * 2^{-2}$

6. Найти сумму чисел Аи B, если числа складываются на сумматоре дополнительного кода (семь разрядов для мантиссы и четыре разряда для порядка):

а)A=0,11101 $*$ 2^5и $B = -0{,}10100 * 2^3$

Тема 4.3. Особенности умножения чисел, представленных в форме с плавающей запятой

Практическое занятие: «Выполнение операции умножения чисел»

Теоретическая часть

Применительно к двоичной системе счисления наиболее известны следующие основные способы выполнения операции умножения:

1) умножение начиная с младших разрядов множителя:

```
    1101   — множимое,
×   1101   — множитель,
    1101
   0000
  1101     — частные произведения,
 1101
10101001   — произведение;
```

2) умножение начиная со старших разрядов множителя:

```
    1101     — множимое,
×   1101     — множитель,
    1101
+   1101     — частные произведения,
     0000
      1101
10101001   — произведение.
```

В обоих случаях операция умножения состоит из ряда последовательных операций сложения частных произведений. Операциями сложения управляют разряды множителя: если в каком-то разряде множителя находится единица, то к сумме частных произведений добавляется множимое с соответствующим сдвигом; если в разряде множителя — нуль, то множимое не прибавляется.

Таким образом, кроме операции сложения чисел для получения произведения необходима операция сдвига чисел. При этом появляется возможность сдвигать множимое или сумму частных произведений, что даетоснование для разных методов реализации операции умножения.

Метод 1. Пусть A — множимое ($A>0$), B — множитель ($B>0$), C — произведение.

В данном случае сдвигается вправо множимое, и умножение начинается со старших разрядов. Структурная схема рассмотренного множительного устройства представлена на рисунке 3, *а*.

Метод 2. Пусть $A = 0, a_1 a_2 \ldots a_n$ — множимое и $B = 0, b_1 b_2 \ldots b_n$ — множитель.

Здесь умножение начинается с младших разрядов, и сдвигается вправо сумма частных произведений. Структурная схема множительного устройства, реализующего этот метод, представлена на рисунке 3, *б*.

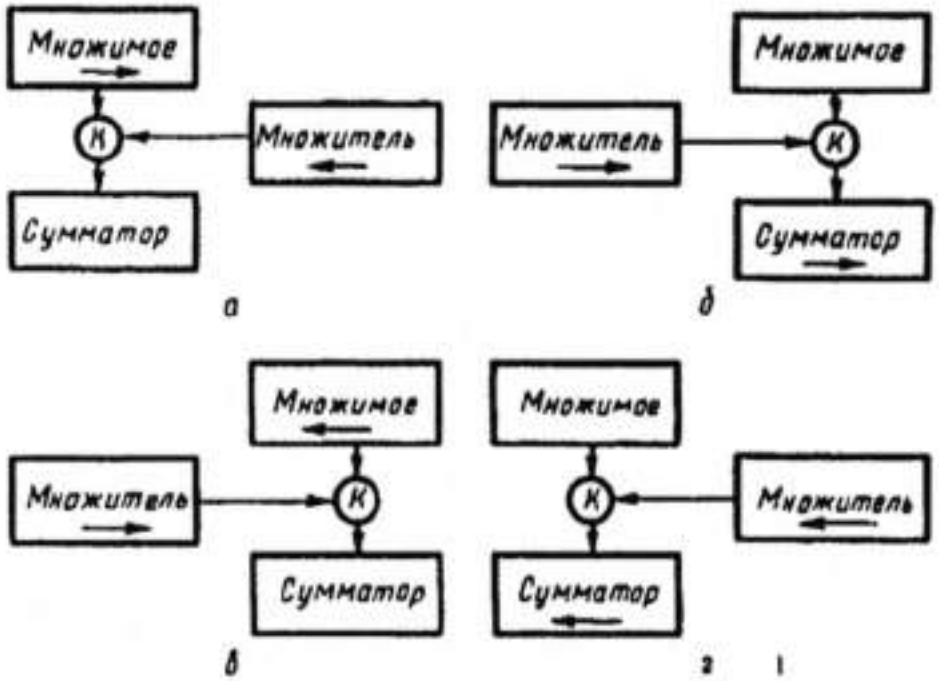


Рисунок 3 – Структурная схема множительного устройства

Метод 3. Умножение начинается с младших разрядов, и множимое сдвигается влево на одни разряд в каждом такте. Схема множительного устройства представлена на рисунке 3, *в*.

Метод 4. Умножение начинается со старшего разряда и в каждом такте сдвигается влево сумма частных произведений. Схема множительного устройства представлена на рисунке 3,*г*.

Таким образом, для реализации операции умножения необходимо иметь сумматор, регистры для хранения множимого и множителя и схему анализа разрядов множителя. Сумматор и регистры должны иметь цепи сдвига содержимого в ту или иную сторону в соответствии с принятым методом умножения.

Умножение чисел, представленных в формес фиксированной запятой, на двоичном сумматоре прямого кода.Пусть заданы машинные изображения двух чисел:

$$[A]_{\text{пр}} = Sg_A, a_1 a_2 \dots a_n, [B]_{\text{пр}} = Sg_B, b_1 b_2 \dots b_n.$$

Тогда их произведение

$$C_{\text{пр}} = Sg_C, c_1 c_2 \dots c_n,$$

где — $Sg_C = Sg_A \oplus Sg_B$; $\oplus$ — знак сложения по модулю 2.

При выполнении этой операции должны быть заданы структурная схема устройства, на котором проводится операция, и метод умножения.

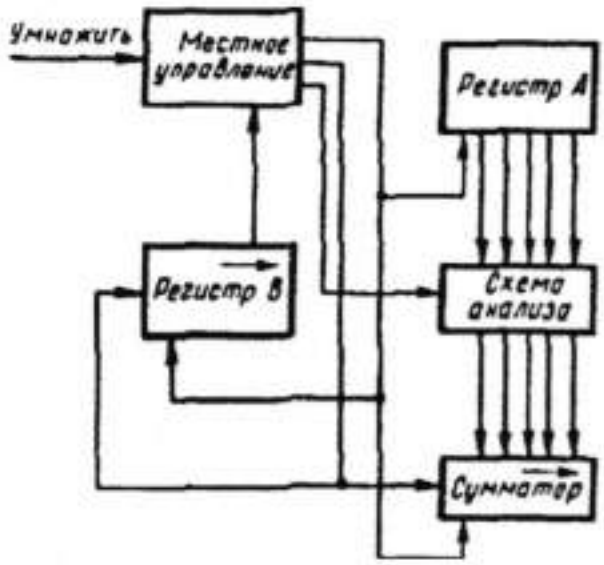


Рисунок 4 - Структурная схема множительного устройства

При умножении будут использованы метод 2 и устройство, показанное на рисунке 4.

Запись всех действий, выполняемых устройством, осуществляется с помощью условных обозначений, т. е. = — оператор присваивания означает, что блоку, который указан слева от оператора, присваивается значение, указанное справа от оператора; $\overrightarrow{[Pr A]}$— сдвиг содержимого регистра Pr вправо на один разряд; [СМ] — содержимое сумматора СМ; ИП — исходное положение.

Для чисел, представленных в форме с плавающей запятой, обязательным является представление в виде мантиссы и порядка (характеристики). При операции умножения действия, выполняемые над мантиссами и порядками, различны: мантиссы перемножаются, порядки складываются. Очевидно, что результат умножения может получиться ненормализованным, когда потребуется нормализация с соответствующей коррекцией порядка результата. Структурная схема множительного устройства изображена на рисунке 5.

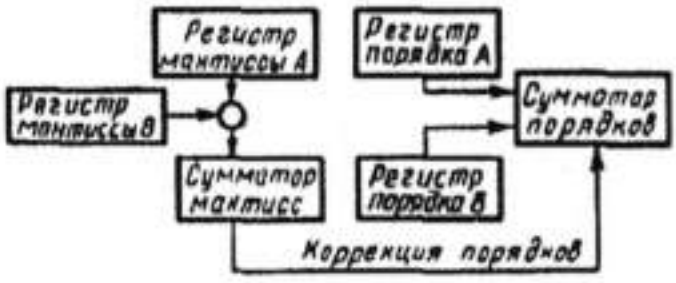


Рисунок 5 – Структурная схема множительного устройства с плавающей запятой

Практическая часть

Пример 1Выполнить умножение чисел с фиксированной запятой на сумматоре прямого кода

$$[A]_{\text{пр}} = 1{,}11010 \text{ и } [B]_{\text{пр}} = 0{,}11001$$

Решение. Знак произведения определением отдельно от цифровой части в соответствии с уравнением

$$Sg_C = Sg_A \oplus Sg_B = 1 \oplus 0 = 1$$

Получение цифровой части можно показать в виде следующей записи. Пусть сумматор имеет 10 разрядов без учета знака, а регистры — 5 разрядов без знака. Введем обозначения соответственно изображения цифровой части множимого и цифровой части множителя.

Последовательность действий в процесс выполнения операции умножения представлена в виде таблицы 5.

Таблица 5

Сумматор	Регистр B	Примечание
0000000000	11001	И.П. $[CM] := 0$; $[PrA] := [A^*]$; $[PrB] := [B]$.
+ 11010		$b_5 = 1$: $[CM] := [CM] + [PrA]$;
11101000000		$[\overline{PrB}]$; $[\overline{CM}]$;
01110100000	—1100	$b_4 = 0$; $[\overline{PrB}]$; $[\overline{CM}]$;
00111010000	—110	$b_3 = 0$; $[\overline{PrB}]$; $[\overline{CM}]$;
00011101000	—11	$b_2 = 1$; $[CM] := [CM] + [PrA]$
+11010		$[\overline{CM}]$ $[\overline{PrB}]$
11101101000		
01110101000	—1	$b_1 = 1$; $[CM] := [CM] + [PrA]$
+11010		
10100010100*		$[\overline{PrB}]$; $[\overline{CM}]$;
10100001010		Конец

* Если в процессе выполнения умножения возникает единица переноса из старшего разряда, то ее надо сохранять.

Ответ: $[C]_{\text{пр}} = 1{,}1010001010.$

Чтобы процесс умножения проходил правильно, необходимо предусмотреть блокировку выработки сигнала переполнения, так как возможно временное переполнение на каком-то шаге умножения (пример 1). Пример показывает, что в данном случае не обязательно иметь сумматор длиной $2n$ разрядов. Хранение «хвостов» произведения можно осуществлять в освобождающихся разрядах регистра множителя. Для этого достаточно обеспечить цепь передачи информации из младшего разряда сумматора в старший разряд регистра множителя.

Пример 2Перемножить числа $A = -0{,}11001 \cdot 2^{-3}$ и $B = 0{,}10011 \cdot 2^{+1}$.

В качестве множительного устройства используется схема, показанная на рисунке 3, где PrA и PrB—соответственно регистры для порядков p_A и p_B.

Решение. Мантиссы перемножаются по правилам, рассмотренным для чисел, представленных в форме с фиксированной запятой. Для перемножения мантисс используется сумматор прямого кода, а для сложения порядков — сумматор обратного кода.

Сначала записываются машинные изображения чисел:

$$[m_A]_{\text{пр}} = 1{,}11001; \quad [p_A]_{\text{об}} = 1{,}100.$$
$$[m_B]_{\text{пр}} = 0{,}10011; \quad [p_B]_{\text{об}} = 0{,}001.$$

Последовательность действий в процессе выполнения операции умножения мантисс представим в таблице 6.

После выполнения указанных действий находится мантисса произведения

$[m_C]_{\text{пр}} = 1{,}0111011011$

Одновременно с этим над порядками проводится операция сложения

$[p_C]_{\text{об}} = [p_A]_{\text{об}} + [p_B]_{\text{об}} = 1{,}100 + 0{,}001 = 1{,}101.$

Так как мантисса результата не удовлетворяет условию нормализации (нарушена левая граница. $\delta = 1$, $\gamma = 0$), то проводится сдвиг мантиссы влево на один разряд $[m'_C]_{\text{пр}} = 1{,}1110110110$, и коррекция порядка$[p'_C]_{\text{об}} = [p_C]_{\text{об}} + 1{,}110 = 1{,}101 + 1{,}110 = 1{,}110.$

Таблица 6

Знак результата	Сумматор	Регистр B	Примечание
$Sg_C = Sg_A \oplus \oplus Sg_B = 1 \oplus 0 = 1$	00000	10011	И.П. $Prm_B := [m_B]$; $Prm_A := [m_A]$; $Prp_A := [p_A]$; $Prp_B := [p_B]$; $CMm := 0$.
	+ 11001		$b_0 = 1$; $CMm := [CMm] + [Prm_A]$;
	11001	01001	$[\overrightarrow{CMm}]$; $[\overrightarrow{Prm_B}]$;
	011001		
	+ 11001		$b_4 = 1$; $CMm := [CMm] + [Prm_A]$;
	1001011		
	1001011	00100	$[\overrightarrow{CMm}]$; $[\overrightarrow{Prm_B}]$;
	01001011	00010	$b_3 = 0$; $[\overrightarrow{CMm}]$; $[\overrightarrow{Prm_B}]$;
	001001011	00001	$b_2 = 0$; $[\overrightarrow{CMm}]$; $[\overrightarrow{Prm_B}]$;
	+ 11001		$b_1 = 1$; $CMm := [CMm] + [Prm_A]$;
	111011011		
	0111011011011	00000	$[\overrightarrow{CMm}]$; $[\overrightarrow{Prm_B}]$;
			Конец

Если сумматор мантисс содержит только *n*разрядов, то после округления получается исходный результат.

Ответ: $C = -0{,}11110 \cdot 2^{-1}$.

Задания для самостоятельной работы

*I*Выполнить умножение чисел с фиксированной запятой на сумматоре прямого кода

а) $A = 0{,}01101$, $B = -0{,}01011$

б)$A = 0{,}11011$, $B = 0{,}10111$

в)$A = 0{,}11010$, $B = -0{,}01011$

г)$A = 0{,}11101$, $B = -0{,}01001$

д)A = - 0,01101, B = 0,10111

2Выполнить умножение чисел с плавающей запятой.Для перемножения мантисс используется сумматор прямого кода, а для сложения порядков — сумматор обратного кода.

а)A=0,10110 $* 2^{-3}$и $B = -0{,}11011 * 2^{-2}$

б)A=0,11001 $* 2^{3}$и $B = 0{,}10111 * 2^{-4}$

в)A=0,11001 $* 2^{-2}$и $B = 0{,}10111 * 2^{+3}$

Тема 4.4. Умножение чисел, представленных в форме с фиксированной запятой, на двоичном сумматоре дополнительного кода
Практическое занятие: «Выполнение операции умножения чисел в дополнительном коде»

Теоретическая часть

В случаях, когда числа в машине хранятся в дополнительных кодах, целесообразно все операции над числами проводить на сумматоре дополнительного кода. Однако при этом возникает ряд особенностей, которые необходимо учитывать.

Произведение дополнительных кодов сомножителей равно дополнительному коду результата только в случае положительного множителя.

Умножение на сумматоре дополнительного кода заключается в анализе разрядов множителя и при $b_i = 1$ в прибавлении дополнительного кода множимого к содержимому сумматора. При этом должны осуществляться модифицированные сдвиги.

Теперь рассмотрим случай, когда множимое A—любое число, а множитель $B < 0$.

При отрицательном множителе произведение дополнительных кодов операндов не равно дополнительному коду результата. Если ввести замену $-A$ на $\overline{A}$, то можно вывести следующее правило:

Если множитель отрицательный, то произведение чисел на сумматоре дополнительного кода получается прибавлением поправки $\overline{A}$ к произведению дополнительных кодов сомножителей.

На сумматоре дополнительного кода в процессе перемножения машинных изображений операндов получаем одновременно знаковую и цифровую части произведения.

Практическая часть

Пример 1. Умножить на сумматоре дополнительного кода (используется метод 2) числа $A = -0{,}10101$ *и* $B = 0{,}10011$.

Решение.

Сначала записываются машинные изображения чисел: $[A]_{\partial}^{м} = 11{,}01011; [B]_{\partial}^{м} = 00{,}10011$.

Последовательность действий, проводимых над числами, представлена в таблице 7.

Ответ: $C = AB = -0{,}0110001111$.

Таблица7

Сумматор	Регистр B	Примечание
00,00000	10011	И.П. СМ := 0; $PrA := [A]^{м}_{д}$; $PrB := [B]$;
+ 11,01011		$b_4 = 1$; СМ := [СМ] + [PrA];
11,01011		
11,10101	→11001	$[\overrightarrow{PrB}]$; $[\overrightarrow{СМ}]$;
+ 11,01011		$b_4 = 1$; СМ := [СМ] + [PrA];
11,00000		$[\overrightarrow{PrB}]$; $[\overrightarrow{СМ}]$;
11,10000	→01100	$b_3 = 0$; $[\overrightarrow{PrB}]$; $[\overrightarrow{СМ}]$;
11,11000	→00110	$b_2 = 0$; $[\overrightarrow{PrB}]$; $[\overrightarrow{СМ}]$;
11,11100	→00011	$b_1 = 0$; СМ := [СМ] + [PrA]
+ 11,01011	0001	
11,00111	→10001	$[\overrightarrow{PrB}]$; $[\overrightarrow{СМ}]$
11,10011		Конец

Пример 2. Умножить на сумматоре дополнительного кода по методу 2 с использованием структурной схемы (рисунок 3), числа $A = -0{,}10111$ и $B = -0{,}11001$.

Решение. Машинные изображения чисел: $[A]^{м}_{д} = 11{,}01001; [B]^{м}_{д} = 11{,}00111$; $\overline{[A]}^{м}_{д} = 00{,}10111$.

Последовательность действийнад числами, показана в таблице 8.

Таблица 8

Сумматор	Регистр B	Примечания
00,00000	00111	И.П. СМ := 0; $PrB := [B'_{м}]_{д}$; [PrA] := $[A]^{м}_{д}$;
+ 11,01001		$\bar{b}_5 = 1$; СМ := [СМ] + $[A]^{м}_{д}$;
11,01001		
11,10100	→10011	$[\overrightarrow{СМ}]$; $[\overrightarrow{PrB}]$;
+ 11,01001		$\bar{b}_4 = 1$; СМ := [СМ] + $[A]^{м}_{д}$;
10,11101		
11,01110	→11001	$[\overrightarrow{СМ}]$; $[\overrightarrow{PrB}]$;
+ 11,01001		$\bar{b}_3 = 1$; СМ := [СМ] + $[A]^{м}_{д}$;
10,10111		$[\overrightarrow{СМ}]$; $[\overrightarrow{PrB}]$;
1101011	→11100	$\bar{b}_2 = 0$; $[\overrightarrow{СМ}]$; $[\overrightarrow{PrB}]$;
11,10101	→11110	$\bar{b}_1 = 0$; $[\overrightarrow{СМ}]$; $[\overrightarrow{PrB}]$;
11,11010	→11111	
+ 00,10111		СМ := [ССМ+ $[\bar{A}]^{м}_{д}$
00,10001	11111	Конец

Ответ: $C = AB = 0{,}1000111111$.

Задания для самостоятельной работы

1. Умножить на сумматоре дополнительного кода (используется метод 2) числа:

а) $A = -0{,}11010$, $B = 0{,}10101$
б) $A = -0{,}10010$, $B = 0{,}11101$
в) $A = -0{,}01101$, $B = 0{,}11011$
г) $A = -0{,}11010$, $B = -0{,}10101$
д) $A = -0{,}11010$, $B = -0{,}11011$
е) $A = -0{,}110110$, $B = -0{,}101001$
ж) $A = 0{,}01101$, $B = -0{,}10011$
з) $A = 0{,}01101$, $B = -0{,}01011$
и) $A = 0{,}110011$, $B = 0{,}100011$

Тема 5.2.Деление чисел, представленных в форме с фиксированной запятой, на сумматорах обратного и дополнительного кода

Практическое занятие: «Выполнение операции деления чисел»

Теоретическая часть

В универсальных вычислительных машинах, как правило, реализуется разновидность «школьного» алгоритма деления.

В общем случае «школьный» алгоритм деления на примере двоичных чисел выглядит следующим образом:

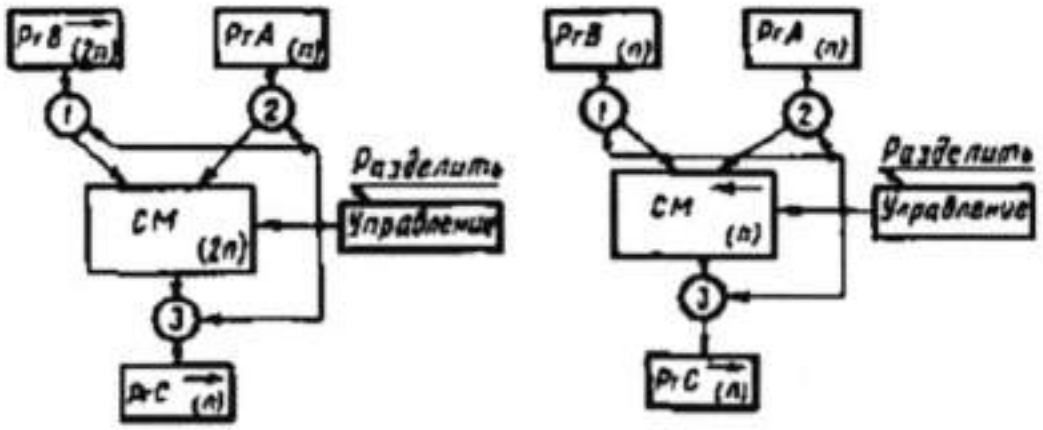
```
делимое –        1100100 | 1010          —делитель
делитель –     –  1010   |-------
                         | 1010          —частное
остаток –      –  0101
                  1010
                -------
восстановление остатка –  – 1011
                  +
                  1010
                -------
                  01010
                – 1010
                -------
                  0000
```

Здесь цифры частного получаются последовательно, начиная со старшего разряда путем вычитания делителя из полученного остатка. Если получен положительный остаток, то цифра частного равна единице; если остаток отрицательный, то цифра частного равна нулю, при этом восстанавливается предыдущий положительный остаток.

В случае положительного остатка для получения следующей цифры частого последний остаток сдвигается влево на один разряд (либо делитель вправо на один разряд) и из него вычитается делитель и т.д.

В случае отрицательного остатка восстанавливается предыдущий положительный остаток прибавлением к отрицательному остатку делителя, и восстановленный остаток сдвигается на один разряд влево (либо сдвигается делитель вправо на один разряд) и из него вычитается делитель. Такой алгоритм деления получил название алгоритма деления с восстановлением остатка.

Реализация рассмотренного алгоритма деления возможна с помощью двоичных сумматоров обратного или дополнительного кода. Для хранения делителя, делимого и накопления очередных разрядов частного выделяются регистры. Минимальный состав блоков, необходимый для выполнения делений, показан на рисунке 6 в двух вариантах. Варианты отличаются способом реализации сдвига. В варианте рисунок 6, а осуществлен сдвиг содержимого регистра делителя, а в варианте рисунок 6, б – сдвиг содержимого сумматора. Стрелки на рисунке указывают направление сдвига. Индекс (n) указывает ориентировочное количество разрядов в цифровой части сумматора и регистра делителя. В дальнейших разделах при выполнении примеров будет использована структура, показанная на рисунке 6,а.



а *б*

Рисунок 6 - Структурные схемы для реализации деления

При делении знаковая и цифровая части частного получаются раздельно. Знак частного образуется по формуле:

$$Sg_C = Sg_A \oplus Sg_B.$$

Для образования цифр частного воспользуемся соответствиемлогических действий над остатками и делителем, приведенным в таблице 9.

Таблица 9

Знак делимого А	+	+	-	-
Знак делителя В	+	-	+	-
Что делать в сумматоре	$A_i + \overline{B} \cdot 2^{-1}$	$A_i + B \cdot 2^{-1}$	$A_i + B \cdot 2^{-1}$	$A_i + \overline{B} \cdot 2^{-1}$

(Символ «—» указывает на изменение знака на противоположный на очередном шаге деления.)

На сумматорах обратного кода возможны алгоритмы получения частного в прямом или обратном коде. Это определяется правилами анализа остатка.

Формальным признаком конца операции деления является количество сдвигов: при достижении числа сдвигов, равного количеству разрядов в частном, вырабатывается сигнал окончания операции деления.

Требуется несколько интерпретировать правила определения цифры частногоМожно сравнивать знаки делимого и остатка на каждом шаге: при совпадении знаков в частном записывается единица, при несовпадении — нуль. Можно сравнивать знаки делителя и остатка, тогда единица в очередной разряд частного записывается при несовпадении знаков, а нуль — при совпадении.

Для получения частного, представленного в обратном коде, все действия должны осуществляться по правилам, приведенным в таблице 10.

Таблица 10

Вариант	1		2		3		4	
Знак делимого	+		+		-		-	
Знак делителя	+		-		+		-	
Что делать в СМ	$A_{i-1} = \overline{B} \cdot 2^{-(i-1)}$		$A_{i-1} = B \cdot 2^{-(i-1)}$		$A_{i-1} = B \cdot 2^{-(i-1)}$		$A_{i-1} = \overline{B} \cdot 2^{-(i-1)}$	
Знак остатка A_i	$A_i \geq 0$	$A_i < 0$	$A_i > 0$	$A_i \leq 0$	$A_i > 0$	$A_i \leq 0$	$A_i \geq 0$	$A_i < 0$
Цифра частного	$\overline{c}_i$=1 $\overline{\overline{c}}_i$=1	$\overline{c}_i$=0 $\overline{\overline{c}}_i$=0	$\overline{c}_i$=0 $\overline{\overline{c}}_i$=1	$\overline{c}$=1 $\overline{\overline{c}}_i$=0	$\overline{c}$=1 $\overline{\overline{c}}_i$=0	$\overline{c}_i$=0 $\overline{\overline{c}}_i$=1	$\overline{c}_i$=0 $\overline{\overline{c}}_i$=0	$\overline{c}_i$=1 $\overline{\overline{c}}_i$=1
		вос.ост.		вос.ост.	вос.ост.		вос.ост.	

Практическая часть

Пример1.Разделить числа А=–0,10011 и В=–0,11001 на двоичном сумматоре обратного кода с получением частного в прямом коде.

Решение: Для реализации примера воспользуемся структурной схемой рис 6,а и методом деления с восстановлением остатка.

Представляем числа в обратных кодах:

$$[A]_{об}^{м} = 11{,}01100; [B]_{об}^{м} = 11{,}00110; \left[\overline{B}\right]_{об}^{м} = 00{,}11001;$$

Определим знак частного:$Sg_c = Sg_A \oplus Sg_B$=$1 \oplus 1 = 0$.

При этом руководствуемся правилами, приведенными в таблице 11.

Таблица11

Вариант	*1*	*2*	*3*	*4*
Знак делимого	+	+	-	-
Знак делителя	+	-	+	-
Что делать в СМ	$A_{i-1} = \overline{B} \cdot 2^{-(i-1)}$	$A_{i-1} = B \cdot 2^{-(i-1)}$	$A_{i-1} = B \cdot 2^{-(i-1)}$	$A_{i-1} = \overline{B} \cdot 2^{-(i-1)}$
Знак остатка A_i	$A_i \geq 0\ A_i < 0$	$A_i \geq 0\ A_i < 0$	$A_i \geq 0\ A_i < 0$	$A_i \geq 0\ A_i < 0$
Цифра частного c_i	$c_i = 1\ c_i = 0$	$c_i = 1\ c_i = 0$	$c_i = 0\ c_i = 1$	$c_i = 0\ c_i = 1$

Последовательность действий для получения цифр частного показана в таблице 12.

Таблица 12

Сумматор ←	Регистр С ←	Примечание
1101100	00000	И.П.СМ:=$[A]^{м}_{об}$:РгВ:=$[B]^{м}_{об}$:РгС:=0;
1011001	0000-	$[\overline{CM}]$; $[\overline{PrC}]$;
+ 0011001		$[CM]:=[CM]+[\overline{B}]^{м}_{об}$;
1110010	00001	$A_1<0; c_1 = 1$;
1100101	0001-	$[\overline{CM}]$; $[\overline{PrC}]$;
+ 0011001		$[CM]:=[CM]+[\overline{B}]^{м}_{об}$;
1111110	00011	$A_2<0; c_2 = 1$;
1111101	0011-	$[\overline{CM}]$; $[\overline{PrC}]$;
+ 0011001		$[CM]:=[CM]+[\overline{B}]^{м}_{об}$;
0010111	00110	$A_3>0; c_3 = 0$;
+ 1100110		*Восстановление остатка*
1111101		$[CM]:=[CM]+[B]^{м}_{об}$;
1111011	0110-	$[\overline{CM}]$; $[\overline{PrC}]$;
+ 0011001		$[CM]:=[CM]+[\overline{B}]^{м}_{об}$;
0010101	11000	$A_4>0; c_4 = 0$;
+ 1100110		*Восстановление остатка*
1111011		$[CM]:=[CM]+[B]^{м}_{об}$;
1110111	1100-	$[\overline{CM}]$; $[\overline{PrC}]$;
+ 0011001		$[CM]:=[CM]+[\overline{B}]^{м}_{об}$;
0010001	11000	$A_5 > 0; c_5 = 0$; $\sum$ сдв=5. *Конец операции*

Ответ: $[C_{пр}] = 0{,}11000$.

Задания для самостоятельной работы

1Разделить числа Аи В= на двоичном сумматоре обратного кода с получением частного в прямом коде:

а) $A = -0{,}10010$, $B = 0{,}11011$

б) $A = -0{,}10101$, $B = 0{,}11101$

в) $A = -0{,}01101$, $B = 0{,}11011$

г) $A = -0{,}10010$, $B = -0{,}11001$

д) $A = -0{,}11010$, $B = -0{,}11011$

Семибитовый код ASII

8-ричный код	Символ	8-ричный код	Символ	8-ричный код	Символ	8-ричный код	Символ
000	NUL	040	SP	100	@	140	
001	SOH	041	!	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	'	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	]	175	}
036	RS	076	>	136	^	176	~
037	US	077	?	137	_	177	DEL

Семибитовый код ASII

8-ричный код	Символ	8-ричный код	Символ	8-ричный код	Символ	8-ричный код	Символ
000	NUL	040	SP	100	@	140	
001	SOH	041	!	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	'	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	]	175	}
036	RS	076	>	136	^	176	~
037	US	077	?	137	_	177	DEL

Семибитовый код ASII

8-ричный код	Символ	8-ричный код	Символ	8-ричный код	Символ	8-ричный код	Символ
000	NUL	040	SP	100	@	140	
001	SOH	041	!	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	'	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	]	175	}
036	RS	076	>	136	^	176	~
037	US	077	?	137	_	177	DEL