

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МУРМАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматики
и вычислительной техники

ИНФОРМАТИКА

Методические указания

к практическим работам для студентов 1 курса специальности

16.03.03 «Холодильная, криогенная техника и системы жизнеобеспечения»

Мурманск

2020

Составитель – Майорова Ольга Викторовна, старший преподаватель кафедры автоматике и вычислительной техники Мурманского государственного технического университета.

ОГЛАВЛЕНИЕ

1. Введение.....	4
2. Практическая работа №1	5
3. Практическая работа №2	7
4. Практическая работа №3	14
5. Практическая работа №4.....	18
6. Практическая работа №5.....	20
7. Практическая работа №6.....	27
8. Практическая работа №7.....	34
9. Практическая работа №8.....	36
10. Практическая работа №9.....	53
11. Практическая работа №10.....	56
12. Практическая работа № 11.....	62
13. Практическая работа № 12.....	69
13. Литература.....	75

Введение

Методические указания для проведения практических работ по дисциплине «Информатика» составлены на основе ФГОС ВО по направлению подготовки (специальности) *16.03.03 Холодильная, криогенная техника и системы жизнеобеспечения*, утвержденного 12 ноября 2015 г. приказом Министерства образования и науки РФ № 1332.

Цель дисциплины – формирование компетенций в соответствии с ФГОС по направлению подготовки бакалавра и учебным планом направления подготовки (специальности) *16.03.03 «Холодильная, криогенная техника и системы жизнеобеспечения»*, профиля (специализации): *«Холодильная техника и технология»*.

Задачи дисциплины: дать необходимые знания по основам информатики, позволяющие успешно применять полученные знания и навыки в практической деятельности.

Практическая работа №1. Измерение информации. Объем данных. Скорость передачи информации.

Цель: Ознакомиться с основными способами измерения информации.

Методические указания

Единицы измерения информации:

1 бит – минимальная единица измерения информации

1 байт = 8 бит

1 Килобайт (Кбайт) = 1024 байт = 2^{10} байт

1 Мегабайт (Мбайт) = 1024 Килобайт = 2^{20} байт

1 Гигабайт (Гбайт) = 1024 Мегабайт = 2^{30} байт

1 Терабайт (Тбайт) = 1024 Гигабайт = 2^{40} байт

Основные формулы:

- 1) $N = 2^i$, где N – число всех возможных равновероятных событий (исходов опыта), i – количество информации в сообщении (бит)
- 2) $i = \log_2(1/p)$, где i – количество информации в сообщении об одном из неравновероятных событий (бит), p – вероятность наступления данного события (исхода опыта)
- 3) $V = k \cdot i$, где V – объем данных в сообщении (бит), k – количество символов в сообщении, i – количество информации (бит), приходящееся на один символ сообщения
- 4) $V = x \cdot y \cdot i$, где V – объем графического изображения (бит), x – количество пикселей по горизонтали, y – количество пикселей по вертикали, i – количество информации (бит) на один пиксел.
- 5) $V = i \cdot t \cdot f \cdot s$, V – объем звукового файла (бит), i – глубина кодирования звука (бит), t – длительность записи (с), f – частота дискретизации (Гц), s – количество каналов звучания ($s=1$ – моно, $s=2$ – стерео, $s=4$ – квадро)
- 6) $U = V / t$, где U – скорость передачи информации (бит/с), V – объем данных (бит), t – время передачи данных (с)

Пример 1. В корзине лежат 8 шаров. Все шары разного цвета. Сколько информации несет сообщение о том, что из корзины достали красный шар?

Решение: Количество информации в сообщении вычислим по формуле $N = 2^i$, где $N = 8$. Тогда $i = 3$.

Ответ: 3 бита.

Пример 2. В корзине лежат 8 черных шаров и 24 белых. Сколько информации несет сообщение о том, что достали черный шар?

Решение: Вычислим вероятность появления шара черного цвета: $p = \frac{8}{24} = \frac{1}{4}$.

Тогда количество информации $i = \log_2 \frac{1}{p} = \log_2 \frac{1}{1/4} = \log_2 4 = 2$ (бита).

Ответ: 2 бита.

Пример 3. Определить количество информации (байт), содержащее в сообщении, состоящем из 256 символов, при условии, что каждый символ кодируется 1 байтом.

Решение: $V = 256 \cdot 8 \text{ бит} = 256 \text{ байт}$

Ответ: 256 байт.

Пример 4. Определить размер файла, содержащего цветное растровое изображение размером 640x480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами.

Решение: $V = 640 \cdot 480 \cdot 3 \text{ байта} = 640 \cdot 480 \cdot 24 \text{ бит} = 7372800 \text{ бит} = 921600 \text{ байт} = 900 \text{ Кбайт}$

Ответ: 900 килобайт.

Пример 5. Определить размер (в мегабайтах) файла, содержащего одноканальную (моно) звукозапись с частотой дискретизации 48 кГц, глубиной кодирования 16 бит, временем записи 2 минуты.

Решение: $V = 48000 \text{ Гц} \cdot 16 \text{ бит} \cdot 120 \text{ с} = 92160000 \text{ бит} = 11520000 \text{ байт} = 112500 \text{ Кбайт} = 10,98 \text{ Мбайт}$

Ответ: 10,98 мегабайт.

Пример 6. Скорость передачи данных равна 256000 бит/с. Передача файла заняла 15 секунд. Определите размер файла в килобайтах.

Решение: $V = v \cdot t = 256000 \text{ бит/с} \cdot 15 \text{ с} = 3840000 \text{ бит} = 480000 \text{ байт} = 468,75 \text{ Кбайт}$ файла

Ответ: 468,75 килобайт.

Пример 7. Документ объемом 10 Мбайт можно передать с одного компьютера на другой двумя способами:

А) Сжать архиватором, передать архив по каналу связи, распаковать

Б) Передать по каналу связи без использования архиватора.

Какой способ быстрее и насколько, если

– средняя скорость передачи данных по каналу связи составляет 2^{18} бит в секунду,

– объем сжатого архиватором документа равен 30% от исходного,

– время, требуемое на сжатие документа – 7 секунд, на распаковку – 1 секунда?

В ответе напишите букву А, если способ А быстрее или Б, если быстрее способ Б. Сразу после буквы напишите количество секунд, насколько один способ быстрее другого. Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Решение: Определим время передачи файла для каждого способа (А и Б).

$$\text{А) } t = t_{\text{сжатия}} + \frac{V_{\text{сж}}}{v} + t_{\text{распаковки}} = 7 \text{ с} + \frac{(10 \cdot 1024 \cdot 1024 \cdot 8 \text{ бит}) \cdot 30\%}{2^{18} \text{ бит/с}} + 1 \text{ с} = 104 \text{ с}$$

$$\text{Б) } t = \frac{V}{v} = \frac{10 \cdot 1024 \cdot 1024 \text{ байт}}{2^{18} \text{ бит/с}} = \frac{10 \cdot 1024 \cdot 1024 \cdot 8 \text{ бит}}{262144 \text{ бит/с}} = 320 \text{ с}$$

Ответ: А216

Задания для выполнения

1. Рассчитать объем видеопамати, необходимый для хранения графического изображения с разрешением 640x480 и количеством отображаемых цветов, равным 65536.

2. В урне находятся 8 белых и 24 чёрных шара. Какое количество информации несёт сообщение о том, что из урны достали белый шар? А чёрный шар?
3. Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 640×480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами.
4. Сообщение содержит 4096 символов. Объём сообщения при использовании равномерного кода составил 1/512 Мбайта. Чему равна мощность алфавита, с помощью которого записано данное сообщение?
5. Производилась двухканальная (стерео) звукозапись с частотой дискретизации 64 кГц и 16-битным разрешением. В результате был получен файл размером 48 Мбайт, сжатие данных не производилось. Определите приблизительно, сколько времени (в минутах) проводилась запись

Практическая работа №2. Кодирование числовой информации. Системы счисления. Представление чисел в памяти компьютера.

Цель: Ознакомиться с понятием кодирования информации. Закрепить навыки работы в различных системах счисления. Изучить представление целых чисел (прямой, обратный и дополнительный коды) и вещественных чисел в памяти компьютера.

Методические указания

Код – это правило однозначного соответствия символов (их комбинаций) первичного¹ алфавита символам (их комбинациям) вторичного алфавита.

Основные *характеристики* кода:

- основание (**q**, количество различных символов, используемых для записи кодовых слов);
- значность (**k**, длина кода) – количество символов в кодовом слове;
- количество комбинаций ($N = q^k$).

Кодирование – процесс перевода информации, представленной символами первичного алфавита, в последовательность кодов.

Декодирование – перевод последовательности кодов в соответствующий набор символов первичного алфавита.

Операции кодирования и декодирования называются *обратимыми*, если их последовательное применение не приводит к потере информации.

Система счисления – совокупность приёмов и правил наименования и обозначения чисел, позволяющих установить взаимно однозначное соответствие между любым числом и его представлением в виде конечного числа символов.

Алфавит системы счисления – символы (цифры, буквы) для записи чисел.

Основание системы счисления – количество символов (цифр, букв) для записи чисел.

¹ Алфавит источника сообщений

Системы счисления можно разделить на две группы: позиционные и непозиционные.

В позиционных системах счисления вклад цифры в число зависит от ее позиции в записи числа. В непозиционных системах счисления значение цифры (символа) остается постоянным.

Примеры: 1) позиционные системы счисления: 10-чная, 2-чная, 8-чная, 16-чная и т.д.;

2) непозиционные системы счисления: римская система счисления.

При записи чисел в различных системах счисления принято основание системы счисления указывать справа внизу возле числа. Если основание не указано, то, как правило, подразумевается 10-чная система счисления.

Развернутая запись числа в любой позиционной системе счисления с основанием q :

$$A_q = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + \dots + a_{-m}q^{-m}$$

где a_i – цифры системы счисления; n, m – число целых и дробных разрядов соответственно.

Пример:

$$1011,1_2 = 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1}$$
$$276,52_{10} = 2*8^2 + 7*8^1 + 6*8^0 + 5*8^{-1} + 2*8^{-2}$$

Правила перевода чисел между системами счисления:

1. Алгоритм перевода целого 10-чного числа в любую другую позиционную систему счисления:
 - делим нацело с остатком число на нужное основание системы счисления;
 - продолжаем процесс деления нацело с остатком до тех пор, пока частное не получится равным нулю;
 - выписываем остатки от целочисленных делений в порядке, обратном их получению.
2. Алгоритм перевода дробного 10-чного числа в любую другую позиционную систему счисления:
 - отдельно переведем целую часть 10-чного числа по алгоритму перевода целых чисел;
 - дробную часть числа умножаем на основание той системы счисления, в которую переводим число;
 - умножение проводим до тех пор, пока в дробной части не получим ноль либо до требуемой в задаче степени точности представления числа;
 - выписываем получившиеся в целой части значения в порядке их получения.
3. Алгоритм перевода числа из какой-либо позиционной системы счисления в 10-чную:
 - пронумеруем разряды числа, начиная с 0 в целой части (от запятой в числе справа налево) и с -1 в дробной части (от запятой в числе слева направо);

- умножаем каждую цифру числа на основание системы счисления (в которой записано число) возведенное в степень номера соответствующего разряда;
 - выполним сложение полученных чисел.
4. Алгоритм перевода 2-чного числа в 8-ричную систему счисления:
- начиная от запятой (справа налево в целой части числа и слева направо в дробной) разбиваем двоичную запись числа на группы по три цифры (триады);
 - заменяем каждую триаду соответствующей 8-ричной цифрой
5. Алгоритм перевода 2-чного числа в 16-ричную систему счисления:
- начиная от запятой (справа налево в целой части числа и слева направо в дробной) разбиваем двоичную запись числа на группы по четыре цифры (тетрады);
 - заменяем каждую тетраду соответствующим 16-ричным символом (цифрой или буквой).
6. Алгоритм перевода 8-чного числа в 2-чную систему счисления:
- каждую цифру 8-ричного числа заменить соответствующей 2-чной триадой (тройкой двоичных цифр);
 - если двоичное представление 8-ричной цифры меньше трех разрядов, то дописать нужное количество нулей слева от двоичного числа;
 - выписать двоичные триады в том порядке, в каком записаны цифры в 8-ричном числе.
7. Алгоритм перевода 16-чного числа в 2-чную систему счисления:
- каждую цифру (букву) 16-ричного числа заменить соответствующей 2-чной тетрадой (четверкой двоичных цифр);
 - если двоичное представление 16-ричной цифры (буквы) меньше четырех разрядов, то дописать нужное количество нулей слева от двоичного числа;
 - выписать двоичные тетрады в том порядке, в каком записаны цифры в 16-ричном числе.

Таблица соответствия чисел в основных позиционных системах счисления:

Основание системы счисления			
10	2	8	16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

16	10000	20	10
17	10001	21	11

Пример 1. Перевести число 75_{10} в двоичную, восьмеричную и шестнадцатеричную системы счисления.

Решение: $75 : 2 = 37$ (остаток 1)
 $37 : 2 = 18$ (остаток 1)
 $18 : 2 = 9$ (остаток 0)
 $9 : 2 = 4$ (остаток 1)
 $4 : 2 = 2$ (остаток 0)
 $2 : 2 = 1$ (остаток 0)
 $1 : 2 = 0$ (остаток 1)

Записываем остатки от целочисленных делений в порядке, обратном их получению: 1001011.

В итоге: $75_{10} = 1001011_2$

Аналогично переведем число 75_{10} в 8-ричную и 16-ричную системы счисления:

$75 : 8 = 9$ (остаток 3)	$75 : 16 = 4$ (остаток 11 или буква B_{16})
$9 : 8 = 1$ (остаток 1)	$4 : 16 = 0$ (остаток 4)
$1 : 8 = 0$ (остаток 1)	$75_{10} = 4B_{16}$
$75_{10} = 113_8$	

Ответ: $75_{10} = 1001011_2 = 113_8 = 4B_{16}$

Пример 2. Перевести число $0,25_{10}$ в двоичную, восьмеричную и шестнадцатеричную системы счисления.

Решение:

0	25
0	*2
0	5
1	*2
0	0

0	25
2	*8
0	0

0	25
4	*16
0	0

Записываем получившиеся в целой части результаты умножений в порядке их получения.

В итоге: $0,25_{10} = 0,01_2 = 0,2_8 = 0,4_{16}$

Пример 3. Записать число 11001111010101_2 в 8-ричной и 16-ричной системах счисления.

Решение: Разобьем двоичную запись число справа налево (т.к. число целое) на группы по три разряда с добавлением незначащих нулей (при необходимости): $11001111010101_2 = 011\ 001\ 111\ 010\ 101$. Заменим полученные двоичные триады соответствующими 8-ричными числами: $011\ 001\ 111\ 010\ 101 = 3\ 1\ 7\ 2\ 5$. Получаем ответ: $11001111010101_2 = 31725_8$

Для перевода исходного числа в 16-ричную систему счисления разобьем двоичную запись число справа налево (т.к. число целое) на группы по четыре

разряда с добавлением незначащих нулей (при необходимости):
 $11001111010101_2 = 0011\ 0011\ 1101\ 0101$. Заменяем полученные двоичные тетрады соответствующими 16-ричными числами: $0011\ 0011\ 1101\ 0101 = 3\ 3\ D\ 5$. Получаем ответ: $11001111010101_2 = 33D5_{16}$

Ответ: $11001111010101_2 = 31725_8 = 33D5_{16}$

Пример 4. Записать числа 523_8 и $6FB_{16}$ в двоичной и десятичной системах счисления.

Решение: $523_8 = 101\ 010\ 011_2$; $523_8 = 5 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0 = 5 \cdot 64 + 16 + 3 = 339_{10}$
 $6FB_{16} = 0110\ 1111\ 1011_2$; $6FB_{16} = 6 \cdot 16^2 + 15 \cdot 16^1 + 11 \cdot 16^0 = 6 \cdot 256 + 240 + 11 = 1787_{10}$

Ответ: $523_8 = 101\ 010\ 011_2 = 339_{10}$; $6FB_{16} = 0110\ 1111\ 1011_2 = 1787_{10}$

Пример 5. Укажите наименьшее четырёхзначное восьмеричное число, двоичная запись которого содержит 6 единиц.

Решение: Каждый разряд 8-ричного числа соответствует трем двоичным разрядам, следовательно, в двоичной записи четырёхзначного восьмеричного числа 12 разрядов. Самый старший разряд равен 1, теперь осталось распределить 5 единиц в записи числа так, чтобы оно оставалось наименьшим: $1\ _ _ _ _ _ _ \ 1\ 1\ 1\ 1\ 1_2$. В остальных разрядах запишем 0. В результате получим следующую последовательность: 100000011111_2 . Переведем двоичное число в 8-ричную систему счисления: $100\ 000\ 011\ 111_2 = 4037_8$

Ответ: 4037_8

Пример 6. В какой системе счисления справедливо равенство $21 + 24 = 100$?

Решение: Пусть x – искомое основание системы счисления. Запишем каждое число в развернутом виде: $21_x = 2 \cdot x^1 + 1 \cdot x^0$, $24_x = 2 \cdot x^1 + 4 \cdot x^0$, $100_x = 1 \cdot x^2 + 0 \cdot x^1 + 0 \cdot x^0$

Получим уравнение: $2 \cdot x + 1 + 2 \cdot x + 4 = x^2$

Решениями данного уравнения являются два корня: $x_1 = -1$ и $x_2 = 5$

Основание системы счисления не может быть отрицательным, следовательно, первый корень не подходит, остается второй $x_2 = 5$.

Ответ: числа записаны в 5-ричной системе счисления.

Пример 7. Укажите через запятую в порядке возрастания все основания систем счисления большие четырех, в которых запись десятичного числа 76 оканчивается на 4.

Решение: $76 - 4 = 72$. Найдем все делители 72, большие четырех: 6, 8, 9, 12, 18, 36. Это и есть искомые основания систем счисления.

Ответ: 6, 8, 9, 12, 18, 36.

Пример 8. Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 25, запись которых в системе счисления с основанием 6 начинается на 4.

Решение: $4x_6 \leq 25_{10}$. В задаче нужно подобрать такие значения x , чтобы полученное 6-ричное число при переводе в 10-чную систему счисления было меньше или равно 25.

$$\text{При } x=0 \text{ получим } 40_6 = 4 \cdot 6^1 + 0 \cdot 6^0 = 24_{10}$$

$$\text{При } x=1 \text{ получим } 41_6 = 4 \cdot 6^1 + 1 \cdot 6^0 = 25_{10}$$

Ответ: 24, 25.

Целые числа могут представляться в компьютере без знака или со знаком.

Целые числа без знака обычно занимают в памяти компьютера один или два байта.

Число 72_{10} в однобайтовом формате: 01001000_2

Число 72_{10} в двухбайтовом формате: 0000000001001000_2 .

Целые числа со знаком. Обычно занимают в памяти компьютера один, два или четыре байта.

В вычислительной технике применяются три формы записи (кодирования) целых чисел со знаком: прямой, обратный и дополнительный коды.

Прямой код двоичного числа включает в себя код знака (знак " + " соответствует 0, знак " - " - 1) и абсолютное значение этого числа.

Обратный код положительных чисел такой же, как и прямой код, в знаковом разряде - 0. Для получения обратного кода отрицательного числа его разряды инвертируются (0 заменяется на 1 и, наоборот), в знаковом разряде - цифра 1.

Дополнительный код положительных чисел такой же, как и прямой код. У отрицательных чисел для получения дополнительного кода нужно прибавить единицу к младшему разряду обратного кода.

В современных ЭВМ, как правило, отрицательные числа представляют в виде дополнительного или обратного кода, что при суммировании двух чисел с разными знаками позволяет заменить вычитание на обычное сложение и упростить тем самым конструкцию арифметико-логического устройства.

Пример 9. Записать прямой код числа 24_{10} в однобайтовой ячейке памяти.

Решение:

- 1) Переведем число 24 в двоичную систему счисления: 11000_2
- 2) Запишем прямой код числа 24 в однобайтовом формате (8 разрядов) с учетом знака числа (в старшем разряде запишем 0, т.к. число положительное): 00011000

Ответ: 00011000

Пример 10. Записать прямой, обратный и дополнительный коды числа -24_{10} в однобайтовой ячейке памяти.

Решение:

- 1) Переведем модуль числа -24 в двоичную систему счисления: 11000_2
- 2) Запишем прямой код числа -24 в однобайтовом формате (8 разрядов) с учетом знака числа (в старшем разряде запишем 1, т.к. число отрицательное): 10011000
- 3) Запишем обратный код числа -24 : 11100111
- 4) Запишем дополнительный код числа -24 : 11101000

Ответ: 11101000

Пример 11. Записать представление числа $37,25_{10}$ в форме с плавающей точкой в 4-х байтовой ячейке памяти.

Решение:

- 1) В 4-х байтовой ячейке памяти 1 разряд отводится под знак числа (0 для положительных, 1 для отрицательных чисел), 7 разрядов – для записи смещенного порядка, 24 разряда – для абсолютной величины мантиисы числа.
- 2) Переведем число $37,25_{10}$ в двоичную систему счисления: $100101,01_2$
- 3) Нормализуем число: $0,10010101 \cdot 10^{110}$ (все числа записаны в двоичной системе счисления)
- 4) Запишем мантиису числа нормализованное число с учетом 24-разрядного представления: $0,100101010000000000000000$
- 5) Вычислим смещенный порядок (прибавив к математическому порядку величину смещения для данного формата, равную 64): $6+64=70$
- 6) Представим смещенный порядок в двоичной системе счисления с учетом 7-разрядного представления: 1000110
- 7) Запишем представление числа $37,25_{10}$ в форме с плавающей точкой в 4-х байтовой ячейке памяти: $01000110100101010000000000000000$, где старший 0 – это знак мантиисы, следующие 7 разрядов (выделены темно-серым цветом) – смещенный порядок, остальные (выделены серым цветом) – абсолютная величина мантиисы.

Ответ: 01000110100101010000000000000000

Задания для выполнения

1. Найти основание системы счисления, в которой десятичное число 59 эквивалентно числу 214 в искомой системе счисления.

2. Расположите числа, записанные в различных системах счисления, в порядке возрастания: 35_{10} , 36_8 , $3A_{16}$, 100101_2 , 130_4 .
3. Вычислить: $1010_{10} + (106_{16} - 11011101_2) * 12_8$
4. Даны 4 целых числа, записанные в двоичной системе: 10001011 , 10111000 , 10011011 , 10110100 . Сколько среди них чисел, больших, чем $A4_{16} + 20_8$?
5. Определите количество натуральных чисел, удовлетворяющих неравенству: $11000011_2 < x < CA_{16}$.
6. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 39 оканчивается на 3.
7. Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 20, запись которых в системе счисления с основанием 3 начинается на 2?
8. Записать прямой, обратный и дополнительный коды числа -35_{10} в 2-х байтовой ячейке памяти.
9. По шестнадцатеричной форме внутреннего представления целого числа в 2-х байтовой ячейке $F67D$ восстановить само число.
10. Записать представление числа $125,62_{10}$ в форме с плавающей точкой в 4-х байтовой ячейке памяти.
11. По шестнадцатеричной форме внутреннего представления вещественного числа 4-х байтовой ячейке памяти $C5DB0000$ восстановить само число

Практическая работа №3. Логические основы работы компьютера.

Цель: Научиться составлять таблицы истинности логических выражений, выполнять преобразования логических выражений с помощью законов и правил алгебры логики, решать логические задачи.

Методические указания

Логическое высказывание – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Истинному значению высказывания ставят в соответствие 1, а ложному – 0.

Для обращения к логическим высказываниям им назначают имена (А, В, и т.п.).

Высказывания могут быть простыми (элементарными) и сложными (составными). Простые высказывания соответствуют алгебраическим переменным, а сложные являются аналогом алгебраических функций. Функции могут получаться путем объединения переменных с помощью логических действий (И, ИЛИ). Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности простых высказываний.

Основные логические операции

1. Операция **НЕ** (отрицание, инверсия). Результат отрицания всегда противоположен значению аргумента.

Таблица истинности:

X	$\neg X$
0	1
1	0

Таблица истинности – это таблица значений переменных для логических операций, в которой указываются все возможные комбинации логических переменных, а также соответствующие им результаты операций.

Количество строк в таблице истинности равно количеству возможных комбинаций значений логических переменных, входящих в логическое выражение (*кол-во строк* = 2^n , где n – количество логических переменных).

Количество столбцов в таблице истинности равно *количеству логических переменных + количеству логических операций*.

2. Логическое **И** (конъюнкция, логическое умножение, &)

Таблица истинности:

X	Y	$X \wedge Y$
0	0	0
0	1	0
1	0	0
1	1	1

Операция конъюнкции имеет результат «истина» только в том случае, если оба ее операнда истинны.

3. Операция **ИЛИ** (дизъюнкция, логическое сложение)

Таблица истинности:

X	Y	$X \vee Y$
0	0	0
0	1	1
1	0	1
1	1	1

Высказывание $X \vee Y$ ложно тогда и только тогда, когда оба высказывания X и Y ложны.

4. Операция **исключающее ИЛИ** (XOR)

Таблица истинности:

X	Y	$X \text{ XOR } Y$
0	0	0
0	1	1
1	0	1
1	1	0

Операция XOR фактически сравнивает на совпадение два двоичных разряда.

5. Операция импликации.

Таблица истинности:

X	Y	$X \rightarrow Y$
0	0	1
0	1	1
1	0	0
1	1	1

Высказывание $X \rightarrow Y$ ложно тогда и только тогда, когда X истинно, а Y ложно.

Импликацию можно выразить через дизъюнкцию и отрицание: $X \rightarrow Y = \neg A \vee B$.

6. Операция эквиваленции (двойная импликация)

Таблица истинности:

X	Y	$X \leftrightarrow Y$
0	0	1
0	1	0
1	0	0
1	1	1

Составное высказывание $X \leftrightarrow Y$ истинно тогда и только тогда, когда значения X и Y совпадают (оба высказывания одновременно либо ложны, либо истинны).

Эквиваленцию можно выразить через отрицание, дизъюнкцию и конъюнкцию: $X \leftrightarrow Y = (\neg A \vee B) \wedge (\neg B \vee A)$

Приоритет операций при вычислении значения логического выражения в порядке понижения):

- 1) отрицание (NOT, НЕ);
- 2) конъюнкция (AND, И);
- 3) дизъюнкция и исключающее ИЛИ (OR, ИЛИ; XOR, ИСКЛЮЧАЮЩЕЕ ИЛИ);
- 4) операции отношения ($=, \neq, >, <, \geq, \leq$).

Если существует необходимость изменения порядка вычисления значения выражения, надо использовать круглые скобки.

Логические законы и правила преобразования логических выражений

Закон тождества: $A = A$

Закон непротиворечия: $A \wedge \neg A = 0$

Закон исключенного третьего: $A \vee \neg A = 1$

Закон двойного отрицания: $\neg(\neg A) = A$

Законы де Моргана: $\neg(A \vee B) = \neg A \wedge \neg B$; $\neg(A \wedge B) = \neg A \vee \neg B$

Закон коммутативности: $A \wedge B = B \wedge A$; $A \vee B = B \vee A$

Закон ассоциативности: $(A \wedge B) \wedge C = A \wedge (B \wedge C)$; $(A \vee B) \vee C = A \vee (B \vee C)$

Закон дистрибутивности: $(A \wedge B) \vee (A \wedge C) = A \wedge (B \vee C)$; $(A \vee B) \wedge (A \vee C) = A \vee (B \wedge C)$

Пример 1. Упростить логическое выражение: $(A \wedge B) \vee (A \wedge \neg B)$.

Решение: Воспользуемся законом дистрибутивности и вынесем за скобки А:
 $(A \wedge B) \vee (A \wedge \neg B) = A \wedge (B \vee \neg B)$.

По закону исключенного третьего $B \vee \neg B = 1$, следовательно: $A \wedge (B \vee \neg B) = A \wedge 1 = A$.

Ответ: А.

Пример 2. Дано логическое выражение, зависящее от 6 логических переменных: $X_1 \wedge \neg X_2 \wedge X_3 \wedge \neg X_4 \wedge X_5 \wedge X_6$. Определить количество различных наборов значений переменных, при которых выражение истинно.

Решение: Высказывание представляет собой конъюнкцию шести логических переменных. Высказывание истинно (по условию задачи). Все составляющие высказывание операнды должны быть истинными (т.к. результат конъюнкции принимает значение «истина» только, если все ее составляющие истинны). Приходим к выводу, что $X_1=1$, $X_2=0$ (т.к. $\neg X_2=1$), $X_3=1$, $X_4=0$ (т.к. $\neg X_4=1$), $X_5 = 1$, $X_6=1$. Только одна последовательность (набор) значений логических переменных приводит исходное уравнение к значению «истина».

Ответ: 1

Пример 3. Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F. Определить подходящее выражение для F.

X	Y	Z	F
1	1	1	1
1	1	0	1
1	0	1	1

1) $X \vee \neg Y \vee Z$ 2) $X \wedge Y \wedge Z$ 3) $X \wedge Y \wedge \neg Z$ 4) $\neg X \vee Y \vee \neg Z$

Решение: Подставим значения логических переменных каждой строки таблицы в логическое выражение и сравним результаты вычислений со значением последнего столбца таблицы. Если вычисленное по формуле значение логической функции совпадет с указанным в таблице, то ответ найден.

Проверим 1) вариант ответа:

$X=1, Y=1, Z=1 \rightarrow F = X \vee \neg Y \vee Z = 1$ совпадает со значением F в таблице истинности

$X=1, Y=1, Z=0 \rightarrow F = X \vee \neg Y \vee Z = 1$ совпадает со значением F в таблице истинности

$X=1, Y=0, Z=1 \rightarrow F = X \vee \neg Y \vee Z = 1$ совпадает со значением F в таблице истинности

Видим, что полученные в вычислениях результаты совпали с данными таблицы истинности, следовательно, это и есть подходящая функция и правильный ответ.

Ответ: 1.

Пример 4. Определить значение числа X , для которого истинно высказывание

$$((X < 4) \rightarrow (X < 3)) \wedge ((X < 3) \rightarrow (X < 1))$$

1) 1 2) 2 3) 3 4) 4

Решение: $((X < 4) \rightarrow (X < 3)) \wedge ((X < 3) \rightarrow (X < 1)) = 1$. Получается: $((X < 4) \rightarrow (X < 3)) = 1$ и $((X < 3) \rightarrow (X < 1)) = 1$. Подставляя поочередно предложенные варианты значений переменной X в выражения, находим, что только при $X = 4$ (4-й вариант ответа) составляющие конъюнкцию выражения истинны и итоговое значение логического выражения истинно.

Ответ: 4.

Задания для выполнения

- 1) Для заданного логического выражения:
 - a) Построить таблицу истинности;
 - b) Упростить высказывания, используя законы алгебры логики;
 - c) Полученный результат проверить, построив для него таблицу истинности.

1	$(A \leftrightarrow B) \vee A\bar{B} \vee C$
2	$(AC \rightarrow B) \vee A\bar{C}$
4	$(\bar{A} \leftrightarrow B)(A \rightarrow BC)$
5	$(B \rightarrow C) \vee (B \rightarrow AC)$

- 2) Указать значения переменных K, L, M, N , при которых логическое выражение $(\neg K \vee M) \rightarrow (\neg L \vee M \vee N)$ ложно.
- 3) Дан фрагмент таблицы истинности выражения F .

x_1	x_2	x_3	x_4	x_5	x_6	F
1	1	0	0	0	0	0
1	0	1	0	0	1	0
1	0	0	1	0	0	0

Определить выражение, соответствующее F .

- 1) $(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6)$
 - 2) $(x_1 \wedge x_3) \vee (x_3 \wedge x_5) \vee (x_5 \wedge x_1)$
 - 3) $(x_2 \wedge x_4) \vee (x_4 \wedge x_6) \vee (x_6 \wedge x_2)$
 - 4) $(x_1 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$
- 4) A, B и C – целые числа, для которых истинно высказывание:

$$(C < A \vee C < B) \wedge \neg(C + 1 < A) \wedge \neg(C + 1 < B)$$

$A=45, B=18$. Определить значение переменной C .

Практическая работа №4. Текстовый процессор (редактор). Создание, редактирование, форматирование текстов, документов.

Цель: Сформировать навыки работы с текстовым процессором (редактором) MS Word.

Методические указания

Текстовый редактор – это программа, обеспечивающая пользователя ПК средствами создания, обработки, печати и хранения документов различной природы и степени сложности, включающих текстовые данные, таблицы, графики, изображения, формулы и т.д.

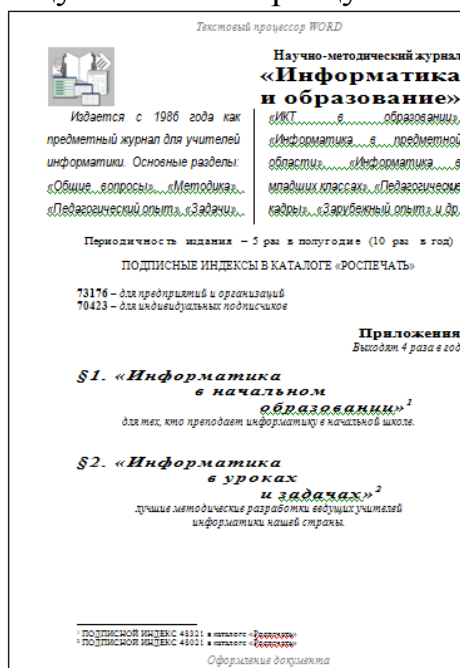
Microsoft Word представляет собой самый распространённый и один из наиболее мощных текстовых редакторов.

Возможности программы:

- набор (ввод) текста;
- редактирование содержимого документа;
- форматирование содержимого документа;
- одновременная работа с несколькими документами;
- проверка орфографии и грамматики;
- автоматическое форматирование документа;
- включение в документы таблиц, рисунков и математических формул;
- коллективная работа над большими документами и т.д.

Задания для выполнения

1. Оформить страницу согласно образцу:



- 1) Наберите весь текст, не применяя операций форматирования и выравнивания используя клавишу ENTER только для отделения абзацев друг от друга.

- 2) Для фрагмента текста «Научно-методический журнал» установите параметры: размер 16, полужирный, выравнивание по правому краю.
 - 3) Для строки «Информатика и образование» установите параметры: размер шрифта 24, полужирный, масштабирование символов 150%, выравнивание по правому краю.
 - 4) Вставьте рисунок, задайте вариант обтекания картинки текстом (обтекание вокруг рамки) и ее местоположение (по левому краю).
 - 5) Для фрагмента текста «Издается ... и др.» установите параметры: тип шрифта Arial Narrow, размер – 16 пт, курсив, выравнивание по ширине, абзацный отступ 1 см, межстрочный интервал Полуторный; оформите этот фрагмент текста в виде двух колонок (Разметка – Колонки – 2 колонки).
 - 6) Для строки «Периодичность...» установите параметры: тип шрифта Times New Roman, размер шрифта 12 пт, масштабирование символов 150%, выравнивание по центру.
 - 7) Самостоятельно, используя образец, подберите параметры форматирования оставшейся части документа.
 - 8) Вставьте символы и номера параграфов перед названиями приложений к изданию (Вставка – Символ – Другие символы – Специальные знаки).
 - 9) Оформите в виде сносок информацию о подписных индексах приложений к журналу (Ссылки – Вставить сноску).
 - 10) Задайте окантовку страницы (Разметка страницы – Границы страниц).
 - 11) Добавьте в документ колонтитулы (Вставка – Верхний колонтитул, Вставка – Нижний колонтитул). Текст верхнего колонтитула: Текстовый процессор WORD. Текст нижнего колонтитула: Оформление документа.
 - 12) Просмотрите полученный документ в режиме предварительного просмотра (Файл – Печать – Предварительный просмотр).
2. Создать текстовый документ, содержащий следующие элементы: форматированный текст, абзац текста в 2 колонки, рисунок, схема, таблица, колонтитулы, сноски, список, границы страницы, специальные символы и т.п.
 3. Создать документ на основе готового шаблона (деловое письмо, резюме) и разработать свой собственный шаблон.

Практическая работа №5. Принципы построения и работы с электронными таблицами. Применение электронных таблиц для решения задач.

Цель: Сформировать навыки работы с табличным процессором MS Excel.

Методические указания

Программы, предназначенные для создания и обработки табличных данных в электронном виде, называются *табличными процессорами*.

Рабочий лист электронной таблицы состоит из *строк* и *столбцов*. Столбцы озаглавлены прописными латинскими буквами (A, B, AA). Строки обозначаются арабскими цифрами. На пересечении строк и столбцов находятся *ячейки* таблицы. Обозначение ячейки составляется из номера столбца и строки, на пересечении которых она находится, например, F4. Одна из ячеек всегда является *активной* и выделяется *рамкой*. Существует другой способ адресации ячейки. Столбцы задаются цифрами, перед номером строки ставится буква R, перед номером столбца – буква C, например, R4C6.

Группа соседних ячеек, образующих в таблице область прямоугольной формы, называется *диапазоном*. Диапазон ячеек обозначается адресами ячеек, расположенных в противоположных углах (верхний левый и правый нижний) прямоугольника и разделенных двоеточием, например A1:A5.

Вводить данные можно только в активную (выделенную) ячейку. В электронных таблицах используются данные следующих типов: текст, число, дата и время, формула, функция. Тип данных определяется автоматически при вводе. Тип данных можно переопределять.

Для ввода данных необходимо выбрать ячейку и начать ввод. В *строке формул* также отобразится содержимое ячейки. Ввод данных можно осуществлять непосредственно в строке формул. Для завершения ввода необходимо нажать клавиши Enter или Tab.

Вычисления в таблицах осуществляются при помощи формул. *Формула* может содержать числовые константы и ссылки на ячейки, соединенные знаками математических операций ("+", "-", "*", "/"). Для определения порядка выполнения действий используются круглые скобки. Перед формулой надо ввести знак равенства (=). Если ячейка содержит формулу, то на рабочем листе в ней отражен результат вычисления. Если сделать ячейку активной, то в строке формул отображается сама формула.

При копировании формулы из одной ячейки в другую автоматически изменяются адреса ячеек, входящих в состав формул. Такая адресация ячеек называется *относительной* и обозначается, например, A1. В тех случаях, когда необходимо не изменять адрес некоторой ячейки при копировании формулы, используется абсолютный адрес ячейки и обозначается, например, \$A\$1. Относительные ссылки автоматически корректируются при их копировании, а абсолютные ссылки – нет. Существует возможность задания смешанного типа адресации ячейки (когда либо строка, либо столбец указаны абсолютным адресом, например, \$A1 или A\$1).

В табличных процессорах имеется большое количество встроенных функций. *Функции* – заранее определенные формулы, которые выполняют вычисления по заданным величинам и в указанном порядке. Запись каждой

функции состоит из трех элементов: знака равенства "="; названия функции; аргумента.

Диаграмма – графическое представление числовых данных. Диаграммы создаются на основе данных из существующих листов. Для построения диаграммы используется *Мастер диаграмм*.

Табличные процессоры предоставляют широкий спектр возможностей для обработки данных: фильтрация и поиск данных, подбор параметра и поиск решения, аппроксимация данных и т.д.

Задания для выполнения

1. Создать таблицу для начисления заработной платы работникам предприятия.

Расчет заработной платы

№	ФИО	Доходы			Налоги	Всего начислено	Всего удержано	К выдаче
		Оклад	Коэффициент	Полярная надбавка	Подходный налог			
1	Петров А.В.	6100	2440	4880	1744,6	13420	5099,6	8320,4
2	Сидоров Н.Г.	6100	2440	4880	1744,6	13420	5099,6	8320,4
3	Артемьева О.Н.	10000	4000	8000	2860	22000	8360	13640
4	Иванова С.Ю.	5000	2000	4000	1430	11000	4180	6820
5	Киров М.Э.	23500	9400	18800	6721	51700	19646	32054
6	Григорьев А.А.	2496	998,4	1996,8	713,856	5491,2	2086,656	3404,544
7	Матвеев И.И.	7850	3140	6280	2245,1	17270	6562,6	10707,4
8	Николаев М.В.	12500	5000	10000	3575	27500	10450	17050
9	Ильин К.Д.	30000	12000	24000	8580	66000	25080	40920
10	Дмитриев В.И.	50000	20000	40000	14300	110000	41800	68200

Указания к выполнению:

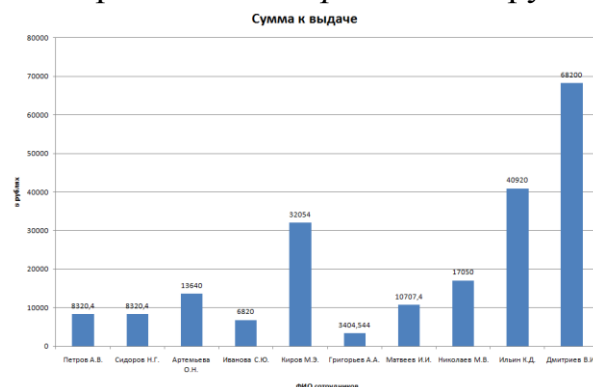
- 1) В ячейку A2 введите заголовок: Расчет заработной платы
- 2) Шапку (названия столбцов) таблицы расположите в диапазоне A5:I6
- 3) Расположите текст в ячейках E6:H6 в две строки:
 - выделите указанные ячейки
 - выполните команду на вкладке Главная – группа Выравнивание – кнопка Перенос текста
- 4) Задайте вертикальное расположение текста в шапке таблицы:
 - выделите ячейки D6:H6
 - выполните команду на вкладке Главная – группа Выравнивание – кнопка Ориентация – Повернуть текст вверх

- 5) Выровняйте заголовок по центру относительно таблицы
- 6) Заполните столбец №, используя автозаполнение (достаточно 10 записей)
- 7) Заполните столбцы ФИО, Оклад согласно образцу, значения остальных ячеек рассчитайте по формулам:
 - коэффициент составляет 40% **от** оклада: установите курсор в ячейку D7, введите формулу =C7*40% или =C7*0,4
 - скопируйте данную формулу в остальные ячейки столбца Коэффициент
 - аналогично рассчитайте Полярную надбавку, которая составляет 80% от оклада
 - в столбце Всего начислено подсчитайте доходы сотрудника (сумма всех доходов)
 - подоходный налог рассчитывается по формуле: = Всего начислено*13%
 - в столбце Всего удержано подсчитываются расходы сотрудника (сумма всех налогов)
 - в столбце К выдаче рассчитывается денежная сумма, выдаваемая сотруднику (доходы – расходы)
- 8) Отформатируйте таблицу: задайте параметры шрифта, выравнивание текста, границы и заливку ячеек таблицы
- 9) Переименуйте Лист1 в Расчет, выбрав команду Переименовать контекстного меню
- 10) Сохраните документ

2. Построить диаграмму, отображающую заработную плату всех сотрудников

Указания:

- 1) Выполните команду Вставка – Гистограмма. На экране появится список, в котором необходимо выбрать *тип диаграммы*. В группе Тип



выберите *Гистограмма с группировкой*

- 2) В группе Данные щелкните по кнопке Выбрать данные для указания *источника данных диаграммы*. В строке Диапазон данных для диаграммы указываются ячейки с данными, на основе которых будет

построена диаграмма. Если выделяемую область в таблице закрывает диалоговое окно, то его можно передвинуть за заголовок или щелкнуть по кнопке в правой части строки Диапазон данных для диаграммы:

- выделите два диапазона ячеек – B7:B16, в котором находятся фамилии сотрудников и I7:I16, в котором отображается сумма к выдаче. Так как, диапазоны несмежные, то их выделяют при нажатой клавише CTRL
- в результате в строке Диапазон данных для диаграммы появится ссылка на лист Расчет и выбранные блоки ячеек: =Расчет!\$B\$7:\$B\$16;Расчет!\$I\$7:\$I\$16
- Если вы ошиблись при указании диапазона, очистите строку Диапазон данных для диаграммы и выделите его заново. Затем нажмите кнопку ОК.

3) Задайте *параметры диаграммы*:

- на вкладке Работа с диаграммами – Макет добавьте Название диаграммы: Сумма к выдаче
- на вкладке Работа с диаграммами – Макет добавьте названия осей: "ФИО сотрудников" (для горизонтальной оси) и "в рублях" (для вертикальной оси)
- на вкладке Работа с диаграммами – Макет уберите Легенду
- на вкладке Работа с диаграммами – Макет задайте отображение подписей данных в группе Подписи данных.

4) Задайте место для размещения диаграммы:

- на вкладке Работа с диаграммами – Конструктор щелкните по кнопке Переместить диаграмму. В появившемся окне Размещение диаграммы установите переключатель в строке На отдельном листе
- в результате в рабочую книгу добавится один лист под названием Диаграмма1 с построенной на нем диаграммой. Готовую диаграмму можно отредактировать.

5) Сохраните файл.

3. Создать таблицу учета выручки от продажи мороженого, если известно, что фирма торгует по пяти округам города только в летние месяцы

*Продажа мороженого
в летний период*

район/месяц	июнь	июль	август	всего по округу	% выручки по округу
Центральный	10 000,00р.	25 002,33р.	22 463,45р.	57 465,78р.	19,06%
Западный	15 045,36р.	24 621,00р.	18 700,50р.	58 366,86р.	19,36%
Южный	25 000,00р.	30 150,00р.	27 320,00р.	82 470,00р.	27,36%

Восточный	20 065,44р.	27 580,26р.	22 650,00р.	70 295,70р.	23,32%
Северный	5 500,00р.	14 790,00р.	12 560,00р.	32 850,00р.	10,90%

75 610,80р.	122 143,59р.	103 693,95р.	301 448,34р.
Всего за месяц			Общая выручка

Указания:

- 1) Создайте новый документ
- 2) В ячейку A3 введите текст "Продажа мороженого", в ячейку A4 – "в летний период"
- 3) Расположите таблицу в диапазоне A6:F11
- 4) Заполните исходными данными столбцы: район/месяц, июнь, июль, август
- 5) В ячейку E7 введите формулу для расчета выручки за три месяца по центральному округу. Скопируйте данную формулу в остальные ячейки столбца Всего по округу
- 6) В ячейку B13 введите формулу, подсчитывающую выручку всех округов за июль. Скопируйте данную формулу в ячейки C13:E13
- 7) Для диапазонов ячеек B7:E11 и B13:E13 задайте денежный формат:
 - выделив указанные диапазоны, выполните действия: вкладка Главная – пункт Число – числовой формат Денежный. В поле Число десятичных знаков введите 2
- 8) В ячейку B14 введите текст "Всего за месяц"; в ячейку E14 – "Общая выручка"
- 9) Заполните столбец "% выручки по округу", используя абсолютную ссылку на ячейку, содержащую общую выручку:
 - В ячейку F7 введите формулу =E7/\$E\$13 (Знак \$ указывает на абсолютную адресацию)
 - Используя автозаполнение, скопируйте формулу в ячейки F8:F11
- 10) Задайте процентный формат:
 - выделив диапазон, выполните действия: вкладка Главная – пункт Число – числовой формат Процентный. В поле Число десятичных знаков введите 2
- 11) Отформатируйте таблицу (задайте параметры шрифта, выравнивание текста, рамки и заливку)
- 12) Заголовки расположите по центру относительно таблицы
- 13) Вставьте рисунок: Вставка – Иллюстрации – Фигуры
- 14) Сохраните файл.

4. Построить отображающую каждого округа выручку.

Указания:



диаграмму, вклад в общую

- 1) Выполните команду Вставка – Диаграммы – Объемная разрезанная круговая
- 2) В качестве источника данных выберите два диапазона: содержащий названия районов и процентный вклад каждого округа в общую выручку
- 3) Для размещения диаграммы выберите имеющийся лист. В итоге на экране появится круговая диаграмма с легендой
- 4) Переместите диаграмму ниже таблицы и измените ее размер с помощью маркеров
- 5) Отредактируйте готовую диаграмму следующим образом:
 - вставьте заголовок диаграммы "Доля выручки каждого района"
 - разместите Легенду внизу диаграммы
 - в пункте Подписи данных выберите Дополнительные параметры подписей данных – Параметры подписи – Значения
- 6) Отформатируйте элементы диаграммы:
 - задайте параметры шрифта для заголовка диаграммы, подписей данных, легенды
 - измените заливку и границу секторов диаграммы
- 7) Сохраните документ.

5. Подготовить таблицу и диаграмму к печати.

Указания:

- 1) Выполните команду Файл – Печать – Предварительный просмотр, щелкните по кнопке Параметры страницы
- 2) В появившемся диалоговом окне во вкладке Поля задайте верхнее и нижнее поля по 2 см, левое и правое – по 1 см, в группе Центрировать на странице активизируйте переключатель горизонтально
- 3) На вкладке Колонтитулы щелкните по кнопке Создать верхний колонтитул, в появившемся окне напишите "Отчет1", нажмите ОК. Для нижнего колонтитула вставьте номер страницы.

Если необходимо подготовить к печати только диаграмму, то для ее просмотра диаграмму нужно выделить щелчком мыши.

6. Разработать таблицу начисления стипендии, содержащую следующие поля: №, ФИО, Названия дисциплин, Средний балл, Размер стипендии. При вычислении среднего балла и размера стипендии использовать соответствующие формулы/функции (СРЗНАЧ, ЕСЛИ и т.д.). Задать форматирование таблицы по своему усмотрению. Построить диаграммы, отображающие содержащуюся в таблице информацию.

Практическая работа №6. Базы данных и системы управления базами данных.

Цель: Научиться работать с базами данных в СУБД Access.

Методические указания

База данных (БД) – это совокупность определенным образом организованной информации, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

Системы управления базами данных (СУБД) – специальные программы, предназначенные для создания баз данных, а также выполнения операции поиска и сортировки данных.

Одной из таких программ является СУБД Microsoft Access, входящая в состав пакета Microsoft Office. Задания и пример их выполнения разработаны для Microsoft Access.

Задания для выполнения

1. Разработать однотабличную базу данных "Адресная книга".

Указания:

- 1) Создайте файл базы данных с именем *Адреса*
- 2) Перейдите в режим конструктора таблицы и создайте следующие поля:
 - **Код адреса** (тип данных – счетчик)
 - **Имя** (тип данных – текстовый)
 - **Фамилия** (тип данных – текстовый)
 - **Адрес** (тип данных – текстовый)
 - **Телефон домашний** (тип данных – текстовый)
- 3) Перейдите в режим таблицы и заполните только поля **Имя** и **Фамилия**:

Код адреса	Имя	Фамилия	Адрес	Телефон домашний
1	Ольга	Сергеева		
2	Николай	Иванов		
3	Алексей	Федоров		
4	Анна	Волкова		
5	Нина	Никольская		

- 4) Перейдите в режим конструктора таблицы, выберите поле **Телефон домашний**, в разделе свойств щелкните в строке *Маска ввода* и задайте шаблон 00-00-00.
- 5) Перейдите в режим таблицы и заполните поле номерами телефона.
- 6) Вернитесь в режим конструктора:
 - Переименуйте поле **Код адреса** в **Номер** и задайте для поля **Номер** новую подпись: №
 - Измените имя поля **Телефон домашний** на **ДТ** и задайте подпись для этого поля: *Домашний телефон*

- Добавьте новое поле **Отч** (тип данных – текстовый, подпись – Отчество) после поля **Имя**
- Добавьте новое поле **ДР** (тип данных – дата/время, форма поля – Длинный формат даты, подпись – День рождения)
- Добавьте новое поле **Хобби** (тип данных – текстовый)

7) Перейдите в режим таблицы и заполните поля новыми данными:

№	Имя	Отчество	Фамилия	Адрес	Домашний телефон	День рождения	Хобби
1	Ольга	Петровна	Сергеева	пр.Связи,100-10	11-22-33	12 февраля 1981 г.	лыжи
2	Николай	Андреевич	Иванов	ул.Пушкина,20-1	22-33-44	14 апреля 1970 г.	теннис
3	Алексей	Павлович	Федоров	ул.Кирова, 1-200	55-33-11	21 мая 1974 г	футбол
4	Анна	Николаевна	Волкова	пр.Ленина, 23-24	11-13-14	1 мая 2000 г.	вязание
5	Нина	Сергеевна	Никольская	ул.Северная, 47/1-75		12 июня 2002 г.	лыжи
6	Юрий	Викторович	Иванов	ул.Пушкина, 12-6		30 сентября 2015 г.	
7	Сергей	Андреевич	Сахнов	пр.Космонавтов, 10/2-4	33-22-11	13 декабря 1978 г.	Теннис
8	Инга	Юрьевна	Горина	пр.Связи,1-45	11-56-78	5 июля 1979 г..	
9	Наталья	Владимировна	Евсеева	ул.Попова,34-6		6 июня 2000 г.	Вязание
10	Михаил	Викторович	Зубарев	пр.Ленина,3-32	12-23-44	10 октября 1981 г.	хоккей

- 8) Выполните сортировку в таблице по фамилиям в алфавитном порядке.
- 9) Выполните поиск записи в таблице (например, по фамилии Иванов).
- 10) С помощью фильтра выясните, кто увлекается вязанием.
- 11) Создайте форму *Анкета* для отображения данных в таблице:
- Щелкните по кнопке *Другие формы* на вкладке *Создание – Формы*
 - В появившемся списке выберите *Мастер форм*
 - В появившемся окне *Создание форм* перенесите все доступные поля таблицы в выбранные поля, нажмите кнопку *Далее*
 - Выберите внешний вид формы *В один столбец*
 - Выберите стиль формы по своему усмотрению
 - Задайте имя формы *Анкета* и нажмите кнопку *Готово*
- 12) Добавьте новые данные, щелкнув по кнопке *Новая запись*.
- 13) В таблице *Адреса* создайте новое поле **Пол** (тип данных – текстовый). В столбце *Описание* конструктора таблицы введите поясняющий текст: 1 – мужской, 2 – женский. Закройте таблицу, сохранив изменения.

- 14) Откройте форму *Анкета* в режиме конструктора.
- Щелкните по кнопке *Группа переключателей* и растяните рамку элемента в любом месте формы
 - В диалоговом окне *Создание группы переключателей* введите две подписи: мужской, женский (в двух строках). Нажмите кнопку *Далее*
 - В следующем окне откажитесь от задания переключателя по умолчанию. Нажмите кнопку *Далее*
 - В следующем окне согласитесь с предложенными именами и значениями. Нажмите кнопку *Далее*
 - В следующем окне активизируйте переключатель *Сохранить значение в поле* и укажите поле *Пол*. Нажмите кнопку *Далее*
 - В следующих окнах задайте оформление и подпись
 - В режиме формы установите для каждого адресата значение *Пол* и закройте форму.
- 15) В таблице *Адреса* создайте новое поле **Знак** (тип данных – текстовый, подпись – Знак Зодиака, размер поля - 10)
- 16) Откройте форму *Анкета* в режиме конструктора
- Добавьте элемент *Поле со списком* и активизируйте переключатель *будет введен фиксированный набор значений*
 - В появившемся диалоговом окне введите названия знаков (овен, телец, близнецы, рак, лев, дева, весы, скорпион, стрелец, козерог, водолей, рыбы)
 - В новом окне активизируйте переключатель *Сохранить в поле*, в раскрывающемся списке выберите **Знак**
 - В последнем окне задайте подпись *Знак Зодиака*
 - Перейдите в режим формы и укажите для каждого адресата знак зодиака.
- 17) С помощью *Мастера запросов* создайте простой запрос, выбрав доступными поля **Имя, Фамилия, Знак** из таблицы *Адреса*.
- 18) В режиме конструктора запроса задайте условия для поиска по знаку зодиака (например, родившихся под знаком весы) и запустите запрос на выполнение (кнопка *Выполнить*)
- 19) Создайте новый запрос с помощью *Конструктора запросов*
- Щелкните по кнопке *Конструктор запросов* на вкладке *Создание*
 - Добавьте таблицу *Адреса*
 - Выберите из списка полей таблицы **Имя, Фамилия, Адрес, ДР, Пол, Хобби**
 - Задайте условия отбора для ответа на вопросы: Кто из мужчин увлекается теннисом; кто живет на пр.Связи; кто имеет имя, начинающееся с буквы Н и т.д.

20) С помощью *Мастера отчетов* создайте отчет о всех записях таблицы Адреса.

2. Разработать многотабличную базу данных "Библиотека".

База данных содержит 4 таблицы. В первой таблице *Разделы* указываются сведения о литературных разделах. Во второй таблице *Книги* хранятся сведения о книгах, имеющихся в домашней библиотеке. В третьей таблице *Читатели* регистрируются сведения об абонентах библиотеки. В четвертой таблице *Учет* осуществляется учет выдаваемых книг из домашней библиотеки.

Указания:

- 1) Создайте файл базы данных с именем Домашняя библиотека.
- 2) В режиме конструктора создайте таблицу *Разделы*, содержащую 2 поля:
 - **Код раздела** (тип данных – текстовый, размер поля – 2, Обязательное поле – да, Ключевое поле)
 - **Название раздела** (тип данных – текстовый, размер поля – 30, Обязательное поле – да).
- 3) В режиме таблицы заполните ее данными:

Код раздела	Название раздела
Д	Детективы
И	Историческая литература
П	Приключения
ПЛ	Политическая литература
ПО	Поэзия
Т	Техническая литература
Ф	Фантастика
Ю	Юмор

Сохраните и закройте таблицу *Разделы*.

- 4) Создайте таблицу *Книги*, содержащую 5 полей:
 - **Код книги** (тип данных – текстовый, размер поля – 5, Обязательное поле – да, Ключевое поле)
 - **Код раздела** (тип данных – текстовый, размер поля – 2, Обязательное поле – да)
 - **Автор** (тип данных – текстовый, размер поля – 40)
 - **Название** (тип данных – текстовый)
 - **Год издания** (тип данных – числовой, размер поля – целое)

5) Перейдите в режим таблицы и заполните таблицу данными:

Код книги	Код раздела	Автор	Название	Год издания
Д1	Д	Г.К. Честерстон	Мудрость отца Брауна	1995
Д2	Д	А.Конан Дойль	Записки о Шерлоке Холмсе	1997
П1	П	Ч. Диккенс	Приключения Оливера Твиста	1990
ПЛ1	ПЛ	Н.Н. Яковлев	Силуэты Вашингтона	1987

- 6) Вернитесь в режим конструктора
- 7) Для поля **Код раздела** создайте столбец подстановки из названий разделов:

- В поле **Код раздела** щелкните в столбце Тип данных (выпадает список типов данных)
- Выберите *Мастер подстановок*². Появится диалоговое окно Создание подстановки. Оставьте активным переключатель «Столбец подстановки использует значения из таблицы или запроса», нажмите кнопку Далее. В следующем окне выберите таблицу *Разделы* и нажмите кнопку Далее. Перенесите поля **Код раздела**, **Название раздела** из области Доступные поля в область Выбранные поля и нажмите кнопку Далее. В появившемся окне увеличьте ширину столбца **Название раздела** так, чтобы все надписи в нем были видны полностью. Оставьте активным флажок «Скрыть ключевой столбец», нажмите Далее. В следующем окне задайте подпись **Раздел**, нажмите Готово. Подтвердите сохранение изменений.

8) Перейдите в режим таблицы. Дополните таблицу новыми данными³:

Код книги	Код раздела	Автор	Название	Год издания
ПЛ2	Политическая литература	А.Лауринчюкас	Врата страха	2000
Ф1	Фантастика	Г.Уэлс	Машина времени	1987
Ф2	Фантастика	А.Н.Толстой	Аэлита	1980
Ф3	Фантастика	И.Ефремов	Туманность Андромеды	1985
Ю1	Юмор	И.Ильф, Е.Петров	Двенадцать стульев	1994
Ю2	Юмор	И.Ильф, Е.Петров	Золотой теленок	1994

Сохраните и закройте таблицу *Книги*.

9) Создайте таблицу *Читатели*, содержащую 4 поля:

- **Код читателя** (тип данных – текстовый, размер поля – 5, Обязательное поле – да, Ключевое поле)
- **ФИО** (тип данных – текстовый, Обязательное поле – да)
- **Адрес** (тип данных – текстовый)
- **Телефон** (тип данных – текстовый, размер поля – 10, маска ввода – 00-00-00)

10) Для поля **Код читателя** заполните следующие свойства:

- **Условие на значение**. Это свойство определяет требования к данным, вводимым в поле. Введите выражение: **“Ж*” or “М*”**
- **Сообщение об ошибке**. Это свойство позволяет создать текст сообщения, выводимого на экран, если введенные данные не соответствуют критерию, заданном в свойстве **Условие на значение**. Введите текст: **“Код читателя должен начинаться с буквы Ж или М”**

11) Перейдите в режим таблицы и заполните ее:

² Мастер подстановки позволяет создать столбец подстановки, в котором отображается список значений. Поместить эти значения в столбец подстановки можно одним из следующих способов: 1) использовать значения из существующей таблицы или запроса; 2) ввести фиксированный набор значений.

³ После использования Мастера подстановки в столбце **Раздел** вместо кодов разделов появились их названия. При вводе новых записей названия разделов можно выбирать щелчком мыши из списка.

Код читателя	ФИО	Адрес	Телефон
Ж1	Иванова Мария Викторовна	ул. Ленина, 100-12	11-22-33
Ж2	Самойлова Валентина Геннадьевна	пр. Гагарина, 1-100	12-34-56
Ж3	Иванова Анна Николаевна	пр. Гагарина, 3/1-17	
М1	Зотов Андрей Николаевич	ул. Сомова, 12а-4	23-45-67
М2	Григорьев Сергей Сергеевич	пер. Туманный, 23	45-67-89
М3	Титов Алексей Романович	ул. Строителей, 3/4-6	
М4	Антонов Павел Сергеевич	ул. Ленина, 12-4	19-86-37

Сохраните и закройте таблицу *Читатели*.

- 12) Создайте таблицу *Учет*, содержащую 4 поля:
- **Код книги** (тип данных – текстовый, размер поля – 5, Обязательное поле – да)
 - **Код читателя** (тип данных – текстовый, размер поля – 5, Обязательное поле – да)
 - **Дата выдачи** (тип данных – дата/время, формат поля – краткий формат даты, Обязательное поле – Да)
 - **Дата возврата** (тип данных – дата/время, формат поля – краткий формат даты)
- 13) Задайте ключ таблицы. Для данной таблицы он является составным и будет включать два поля: **Код книги** и **Дата выдачи**. Выделите эти строки при нажатой клавише CTRL и нажмите кнопку **Ключевое поле**.
- 14) Перейдите в режим таблицы и заполните ее:

Код книги	Код читателя	Дата выдачи	Дата возврата
Д1	М1	11.12.2001	20.01.2002
Д1	М3	17.03.2002	24.04.2002
ПЛ1	Ж1	17.03.2002	01.06.2002
ПЛ2	М2	19.04.2002	20.08.2002
Ф1	Ж1	11.01.2003	
Ф2	Ж1	22.01.2003	03.02.2003
Ф3	Ж2	02.02.2003	I
Ю1	Ж3	04.02.2003	
Ю2	М1	04.02.2003	
ПЛ2	М3	04.02.2003	

Сохраните и закройте таблицу *Учет*.

- 15) Создайте схему данных:
- Щелкните по кнопке *Схема данных* (вкладка *Работа с базами данных*)
 - На экране появятся две таблицы: *Разделы* и *Книги*.
 - Добавьте две остальные таблицы в схему данных, щелкнув по кнопке *Отобразить таблицу* и указав таблицы *Читатели* и *Учет*

- Установите связи между таблицами:
 - a) Перетащите поле **Код книги** из таблицы *Книги* на поле **Код книги** в таблице *Учет*. На экране появится диалоговое окно *Связи*
 - b) Установите флажок *Обеспечение целостности данных*. При этом станут доступными два нижних флажка *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*
 - c) Щелкните по кнопке *Создать*. В результате между таблицами будет установлена связь с отношением «один-к-многим». Это значит, что каждой записи из таблицы *Книги* может соответствовать несколько записей в таблице *Учет*
 - d) Аналогично установите связь между таблицами *Учет* и *Читатели* по полю **Код читателя**
 - e) Выполните двойной щелчок на линии, связывающей таблицы *Разделы* и *Книги*. В окне *Связи* включите флажок *Обеспечение целостности данных*, нажмите *Создать*
 - Закройте окно *Схема данных*, сохранив сделанные изменения.
- 16) Создайте запрос, который выводит информацию о количестве книг в каждом разделе:
- На вкладке *Создание* нажмите кнопку *Конструктор запросов*
 - Добавьте две таблицы: *Книги* и *Разделы*
 - В первый столбец из таблицы *Разделы* перетащите поле **Название раздела**
 - Во второй столбец перетащите поле **Название** из таблицы *Книги*
 - Щелкните по кнопке *Итоги* (в макете запроса появится новая строка *Групповая операция*)
 - Щелкните в ячейке *Группировка* во втором столбце (выпадет список статистических функций)
 - Выберите функцию *Count*⁴ и запустите запрос на выполнение
 - Появится таблица с двумя столбцами: *Название раздела* и *Count_Название*
 - Вернитесь в режим конструктора
 - Измените подпись для поля **Count_Название**
 - a) Щелкните по кнопке *Свойства*. Появится диалоговое окно *Свойства поля*
 - b) На вкладке *Общие* в строке *Подпись* введите текст: *Количество книг в разделе*
 - Запустите запрос на выполнение. Обратите внимание на изменившийся заголовок второго столбца.

⁴ Функция *Count* возвращает количество всех непустых значений поля.

- Сохраните Запрос под именем *Количество книг в разделе*.
- 17) Создайте и сохраните запрос, который выдает информацию о книгах, вышедших после 1990 года. Какие из этих книг относятся к разделу Детективы.
 - 18) Создайте и сохраните запрос, выводящий информацию о читателях, не сдавших книги. При этом вывести ФИО читателя, его телефон, адрес и название книги.
 - 19) Создайте и сохраните запрос о всех выданных книгах в 2001 или 2002 году. Подготовьте на основе этого запроса отчет.
 - 20) Создайте форму для ввода информации о читателях. Добавьте в базу данных информацию о трех новых читателях.
 - 21) Создайте форму для ввода информации о книгах. Добавьте сведения о нескольких книгах по разделам Поэзия и Историческая литература.
 - 22) Дополните таблицу *Учет* информацией о взятых книгах новыми читателями.
 - 23) Создайте отчет о всех книгах содержащихся в домашней библиотеке.

Практическая работа №7. Создание мультимедиа презентаций.

Цель: Сформировать навыки создания мультимедиа презентаций с помощью программы Microsoft PowerPoint.

Методические указания

PowerPoint – это программа подготовки и демонстрации связной последовательности слайдов, выполненных в едином стиле и хранящихся в одном файле. Демонстрируемая последовательность слайдов называется презентацией. Программа PowerPoint входит в состав семейства программных продуктов Microsoft Office.

Слайд – это одна страница визуального материала.

Задания для выполнения

1. Создание презентации по теме *«Структура курса лекций по изучению пакета Microsoft Office»*

Указания:

- 1) Запустите Microsoft PowerPoint
- 2) На экране появится первый слайд с разметкой для ввода текста (метками – заполнителями)
- 3) Задайте оформление слайдов (вкладка Дизайн – Темы)
- 4) Щелкнув по метке-заполнителю, введите текст заголовка слайда «Microsoft Office».
- 5) Аналогично введите текст подзаголовка «содержание курса 36 часов».

6) Создайте 2-й слайд:

- Для создания нового слайда выполните команду **Создать слайд** на вкладке Главная. В диалоговом окне **Создать слайд** выберите разметку слайда **Два объекта**
- В метке-заполнителе заголовка введите текст «Microsoft Word»
- Заполните левую колонку текстом:
 - Форматирование абзацев
 - Текстовые эффекты
 - Форматирование списков
 - Использование графики
- Заполните вторую колонку текстом:
 - Создание и форматирование таблиц
 - Вставка объектов
 - Создание документов на основе шаблонов

7) Создайте 3-й слайд:

- Добавьте новый слайд и выберите ту же разметку слайда
- Введите заголовок «Microsoft Excel». Выровняйте его по центру
- Введите текст в левую колонку:
 - Основные понятия
 - Заполнение ячеек данными
 - Форматирование таблиц
 - Ссылки на ячейки
- Введите текст в правую колонку:
 - Сортировка данных
 - Фильтрация данных
 - Заполнение таблицы в режиме формы
 - Мастер функций
 - Организация работы с листами рабочей книги

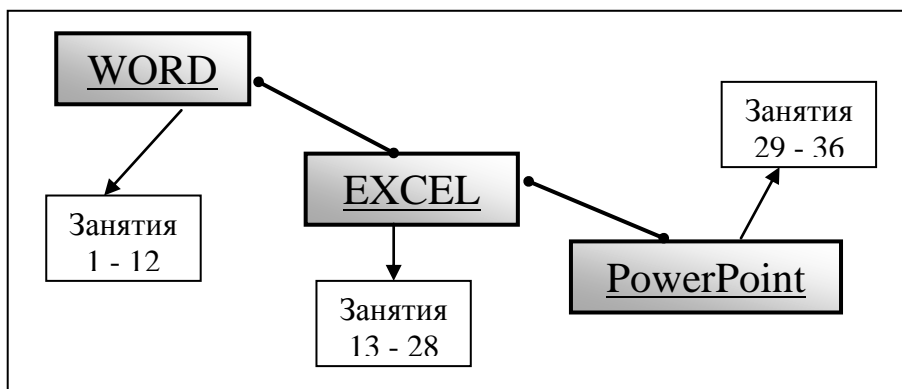
8) Создайте 4-й слайд:

- Введите заголовок «Microsoft PowerPoint» и расположите его по центру
- Введите текст:
 - Создание презентации
 - Применение шаблона дизайна
 - Форматирование шрифта
 - Рисование и вставка графики
 - Настройка анимации
 - Демонстрация презентации
 - Печать слайдов

9) Создайте 5-й слайд:

- Вставьте новый слайд и выберите разметку **Только заголовок**
- Введите текст заголовка «Структура курса» и выровняйте его по правому краю

- Оформите схему с помощью элементов на вкладке *Средства рисования*:



- 10) Осуществите демонстрацию презентации (вкладка Показ слайдов).
- 11) На 3-ем слайде измените вид маркера (символ, цвет, размер) в обеих колонках.
- 12) Поставьте слайд со структурной схемой после титульного:
 - Перейдите в режим сортировщика слайдов (вкладка Вид – Сортировщик слайдов)
 - Поместите 5-й слайд между 1-м и 2-м (перетащите слайд)
- 13) Задайте эффект появления на экране для каждого слайда (вкладка Анимация).
- 14) Задайте эффекты анимации для элементов слайдов.
- 15) Сохраните презентацию.

Практическая работа №8. Основы компьютерной графики.

Цель: Познакомиться с основами компьютерной графики. Освоить основные приемы создания чертежей с помощью программы КОМПАС-3D.

Методические указания

Для знакомства с компьютерной графикой и получения навыков построения чертежей в данной практической работе используется система автоматизированного проектирования КОМПАС-3D LT. Система предназначена для выполнения учебных проектно-конструкторских работ в различных отраслях деятельности.

Задания для выполнения

1. Создать чертеж, начертить геометрическую фигуру (рис. 1), нанести размеры и сохранить чертеж.

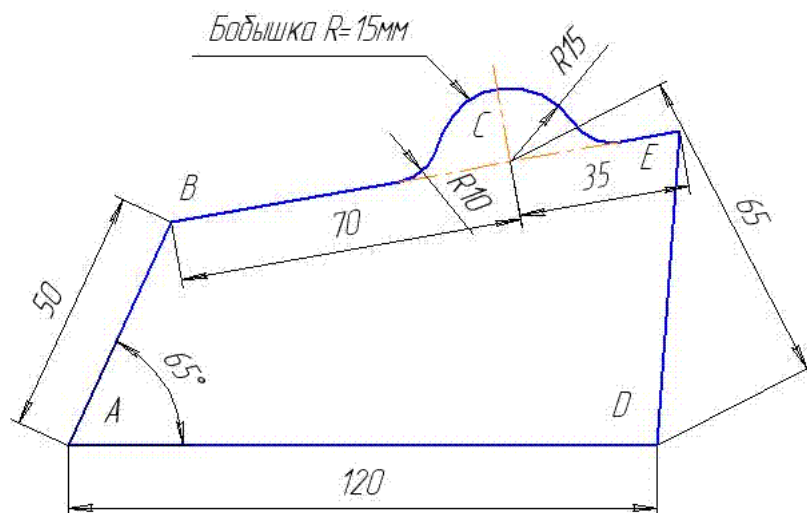


Рис. 1. Геометрическая фигура.

Порядок выполнения работы:

1. Создайте чертеж в программе КОМПАС-3D LT (рис. 2).

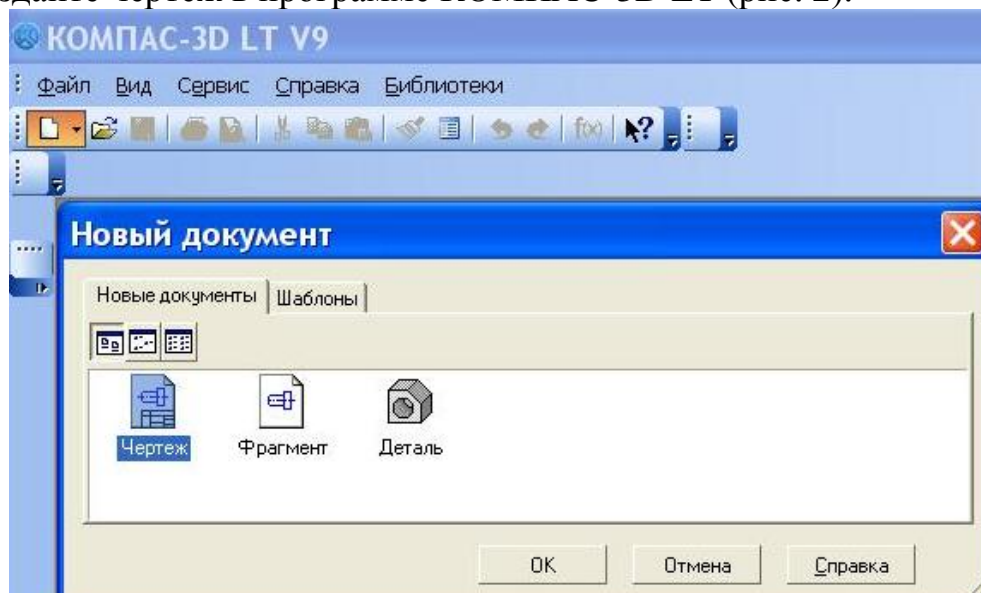


Рис. 2. Создание нового чертежа.

2. После создания чертежа проверьте подключение панелей. Обратите внимание на подключение на **Панели инструментов Глобальных привязок** (рис. 3).

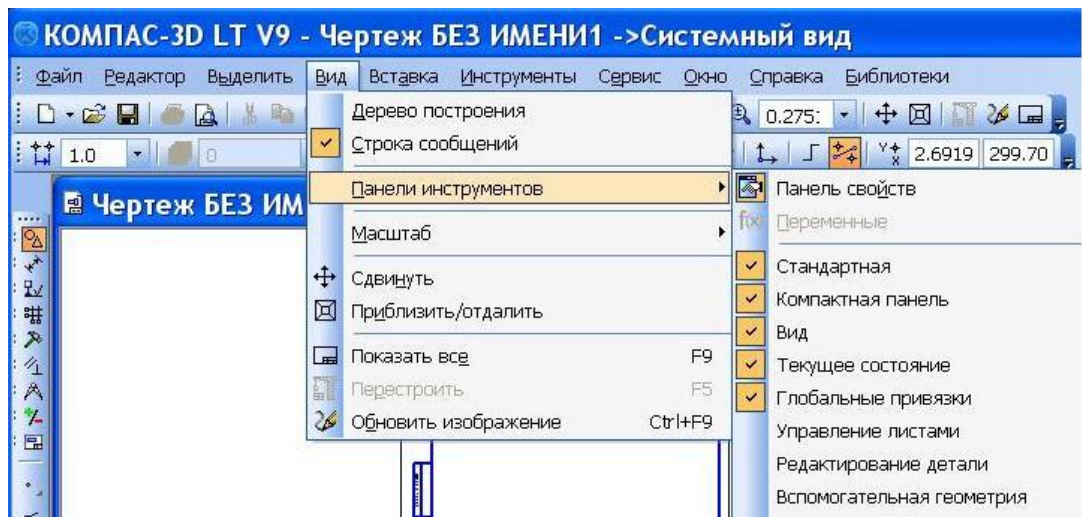
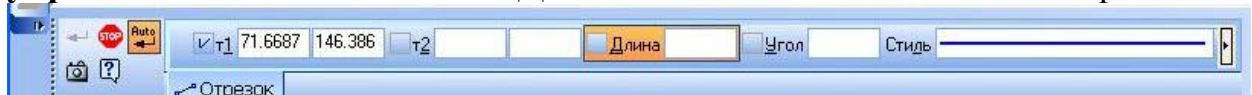


Рис. 3. Проверка подключения панелей инструментов.

3. В Меню выберите **Инструменты** в подменю - **Отрезки** и в следующем подменю выберите **Отрезок**.

Внизу чертежа появились **Панель свойств** и **Панель специального управления** с активным окном **Длина** для ввода значения длины отрезка.



4. Обратите внимание на кнопку команды **Автосоздание объекта** на **Панели специального управления**. В режиме автоматического создания после ввода необходимого набора параметров объект создается щелчком ЛКМ (левой кнопки мыши) без дополнительного нажатия на кнопку **Создать объект**.

Если кнопка **Автосоздание объекта** отжата – щелчком ЛКМ (левой клавиши мыши) нажмите ее. Колесиком мыши увеличьте масштаб поля чертежа. Увеличение масштаба поля чертежа производится при вращении колесика мыши, сдвиг - при нажатии на колесико мыши.

5. Введите в активное окно **Длина** 120 и нажмите клавишу **Tab**, активным стало окно **Угол** для ввода значения угла.

6. Введите в активное окно **Угол** 0 и нажмите **Enter** для фиксирования ввода. На экране появился фантом отрезка с введенными значениями длины и угла (фантом изображен тонкой линией).

7. Щелчком левой клавиши мыши (ЛКМ) зафиксируйте расположение фантома отрезка на чертеже. Стиль линии зафиксированного отрезка изменился с **Тонкой** на **Основную**. Вновь стало активным поле **Длина**.

8. С клавиатуры в поле **Длина** введите 50. Нажмите **Tab**.

9. В поле **Угол** введите 65. Нажмите **Enter**. На экране появится фантом второго отрезка.

10. Подведите фантом отрезка к левой точке первого отрезка, появится надпись "Ближайшая точка". Это означает, что можно фиксировать ввод, так

как ближайшая точка является той, из которой требуется провести отрезок (рис. 4).

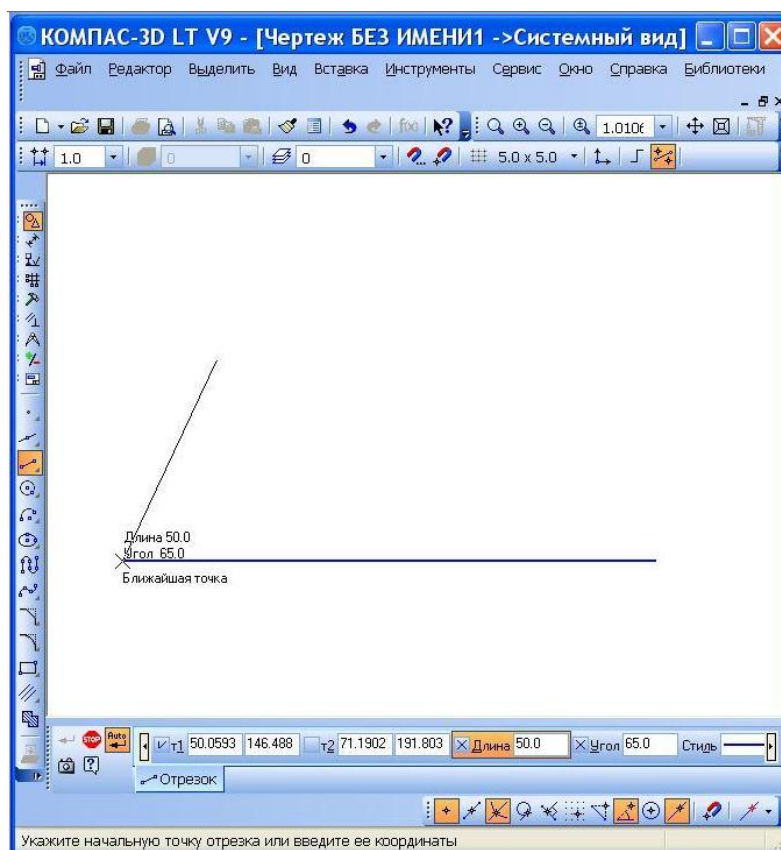


Рис. 4. Фантом второго отрезка подведен к месту ввода.

11. Щелчком ЛКМ зафиксируйте второй отрезок.

12. Прервите команду. Нажмите или кнопку **Прервать команду** или клавишу **Esc** на клавиатуре.

Для нахождения точки **С** проведите две вспомогательные окружности радиусом 70 и 65 мм. Для этого:

13. Из **Меню Инструменты** выберите **Окружность, Окружность с центром**.

После выбора типа окружности активизировалась **Панель свойств**.

14. Введите значение 70 в поле **Радиус**. Выберите **Без осей**.

15. Подведите фантом окружности к нужной точке (рис. 5).

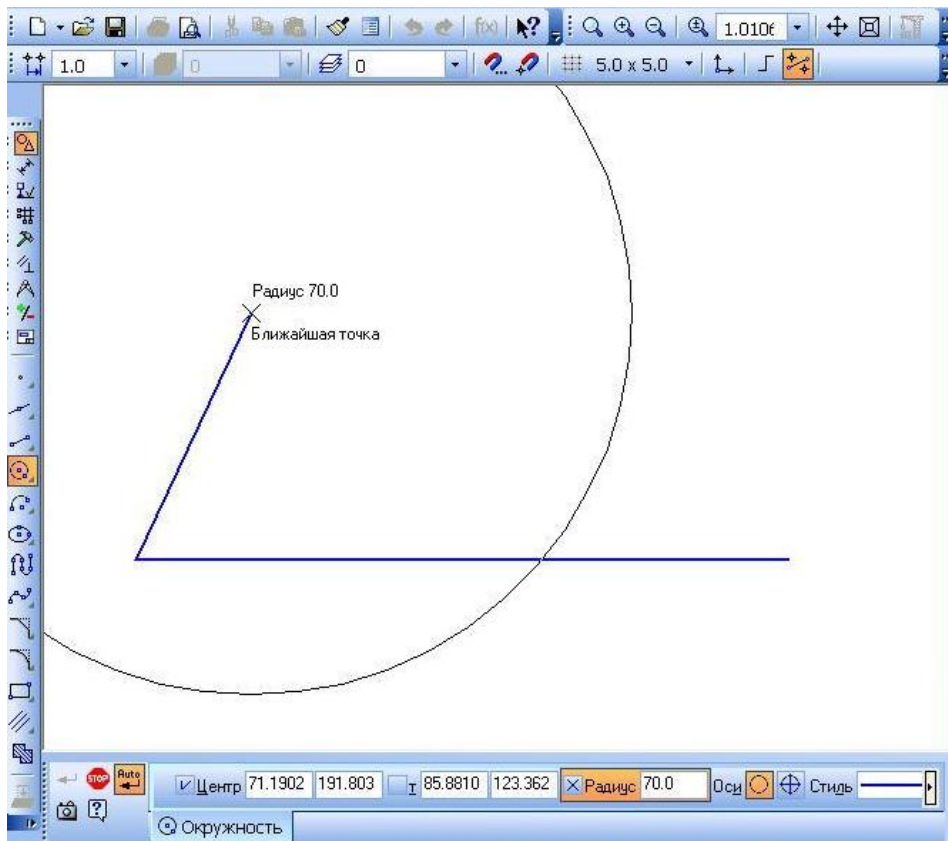


Рис. 5. Фантом первой окружности.

16. ЛКМ зафиксируйте первую окружность.

17. Для второй окружности введите значение 65 в активное поле **Радиус** (рис. 6).

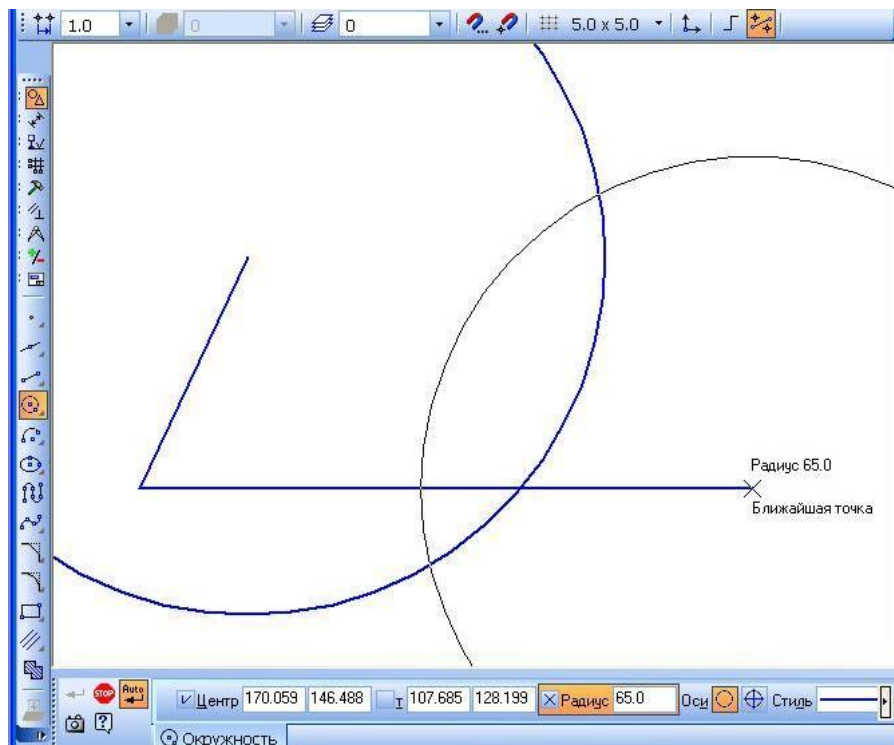


Рис. 6. Фантом второй окружности перед фиксированием ввода.

18. ЛКМ зафиксируйте вторую окружность.

Измените стиль линий окружностей с **Основная** на **Тонкую**. Для этого:

19. ЛКМ выделите вначале одну окружность

20. При нажатой клавише **Shift** выделите вторую окружность. Правой кнопкой мыши вызовите контекстное меню. Измените стиль линии с **Основная** на **Тонкую**.

21. Через **Меню** выберите команду **Отрезок** (рис. 7).

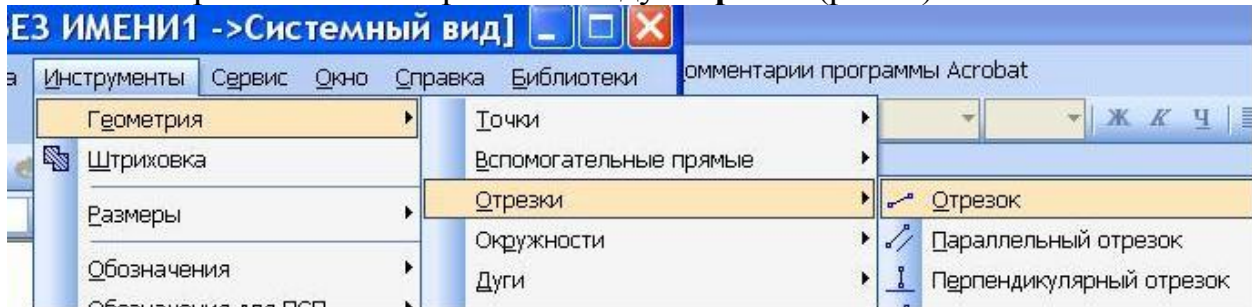


Рис. 7. Выбор отрезка через Меню.

22. На **Панели свойств** в поле **Длина** введите длину отрезка АЕ ($70 + 0,5 \cdot 70 = 105$ мм).

23. ЛКМ зафиксируйте точку начала отрезка.

24. Подведите фантом отрезка к месту пересечения окружностей (рис. 8).

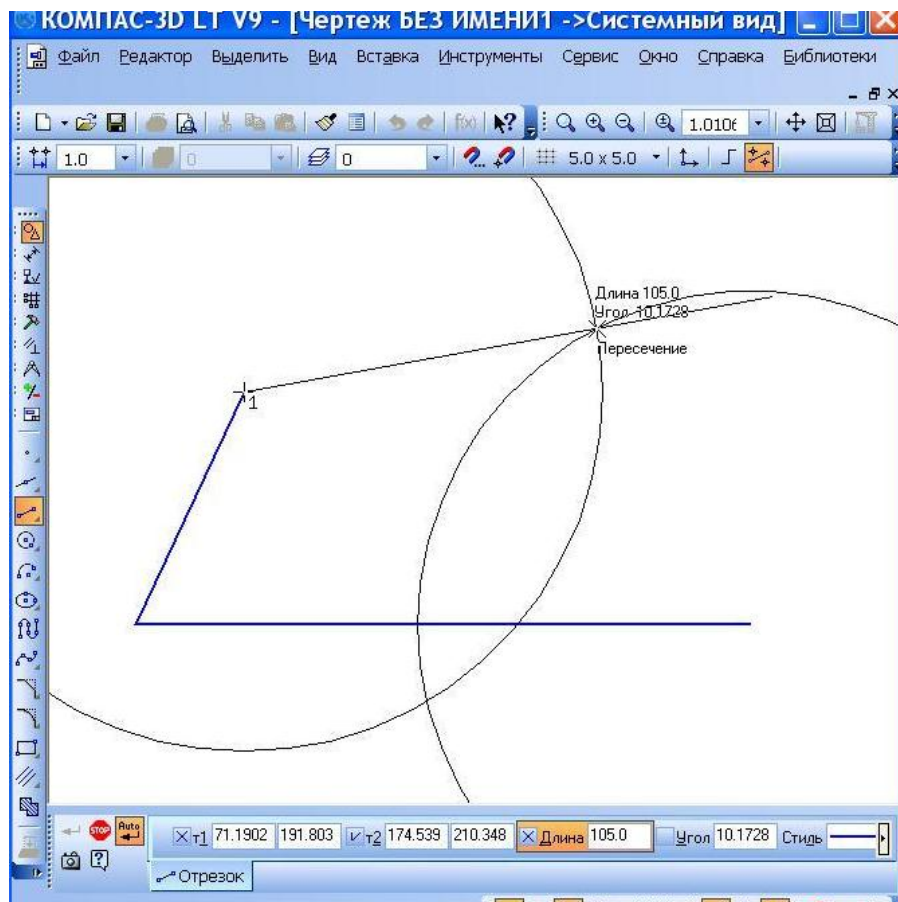


Рис. 8. Фантом отрезка АЕ готов к фиксации.

25. ЛКМ зафиксируйте ввод отрезка АЕ.
26. ЛКМ зафиксируйте точку D (правый конец горизонтального отрезка).
27. Направьте фантом отрезка к точке E (правый конец только что созданного отрезка). (Рис. 9).

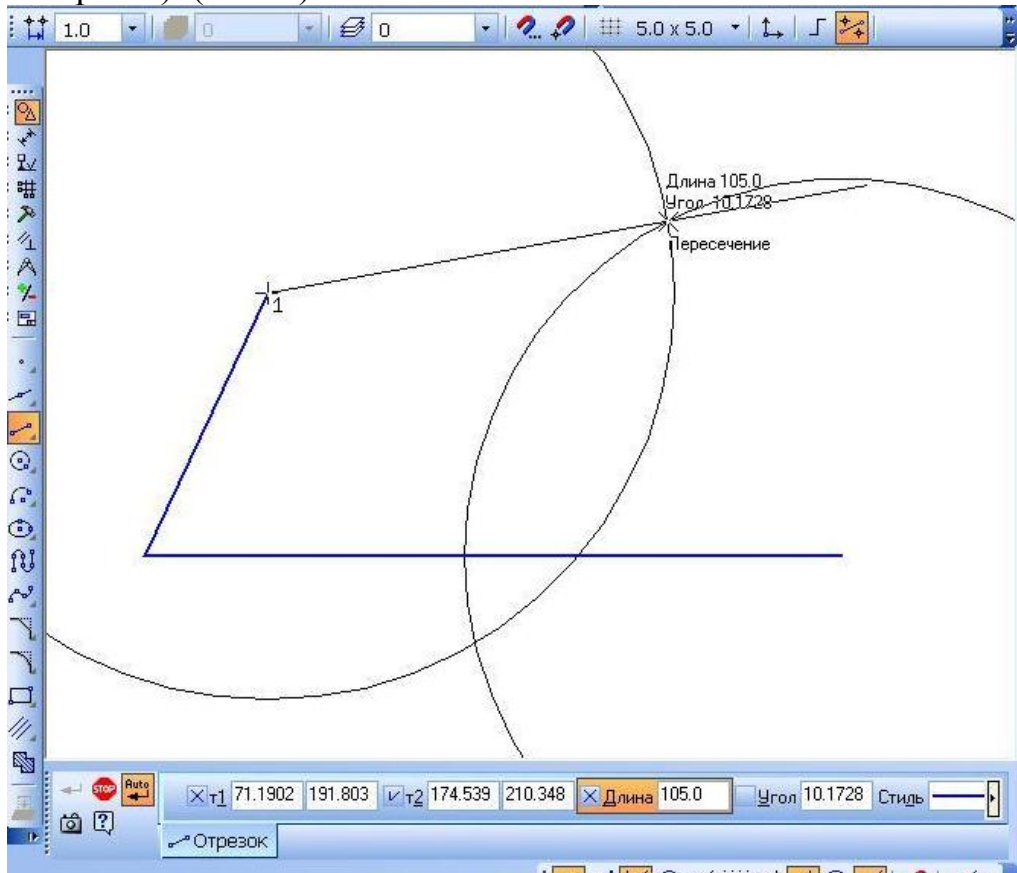


Рис. 9. Фантом отрезка DE готов к фиксации.

28. ЛКМ зафиксируйте отрезок. Прервите команду нажатием **Esc**.
29. Через **Инструменты - Геометрия** вызовите команду **Окружность**.
30. С клавиатуры введите в поле **Радиус** - 15. Выберите **Без осей**.
31. Подведите фантом окружности к месту пересечения окружностей (рис. 10).

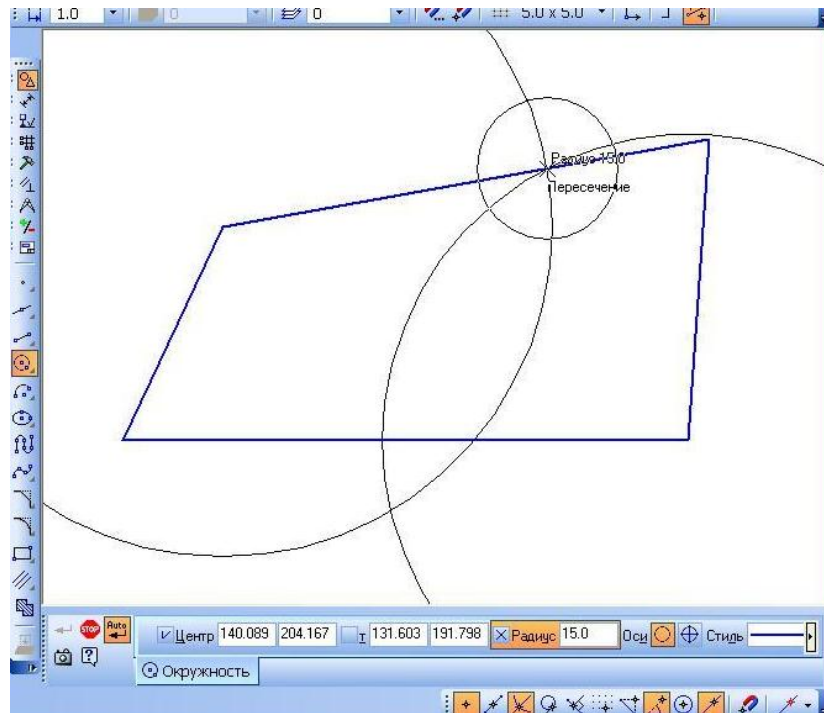



Рис. 10. Фантом окружности готов к фиксации.

32. ЛКМ зафиксируйте ввод окружности. Прервите команду (нажмите **Esc**).

Выделите и удалите обе вспомогательные окружности. Для удаления части окружности и линии:

33. Через **Редактор - Удалить** вызовите команду **Часть кривой**. Указатель мышки приобрел вид "ловушки"  .

"Ловушкой" наведите на нижнюю часть окружности, окружность изменила цвет (рис. 11).

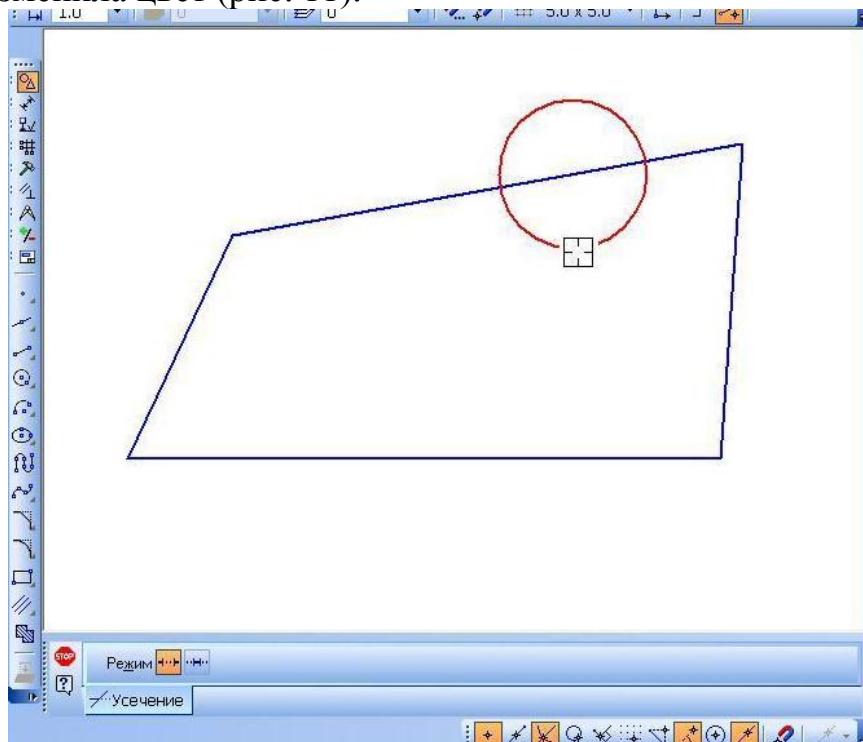


Рис. 11. Вид окружности перед удалением нижней части.

34. Щелчком ЛКМ удалите часть окружности.
35. Аналогично удалите часть отрезка (рис. 12).

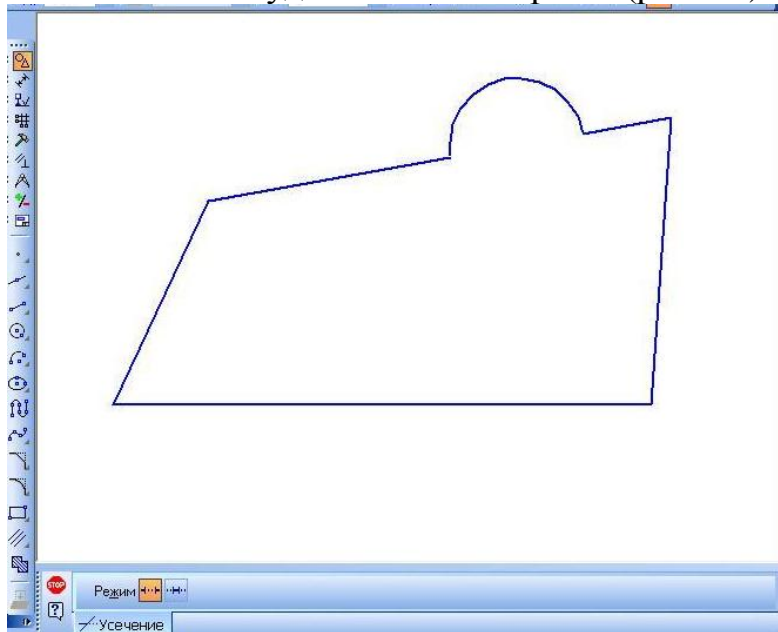


Рис. 12. Вид фигуры после удаления нижней части окружности и части отрезка перед прерыванием команды.

36. Прервите команду **Удалить Часть кривой** (нажмите **Esc**).
37. Вызовите команду **Скругление** (пункт меню **Инструменты-Геометрия-Скругления-Скругление**).
38. Поочередно выделяя "ловушкой" обе линии выполните:
 - первое скругление,
 - второе скругление (рис.13, 14).

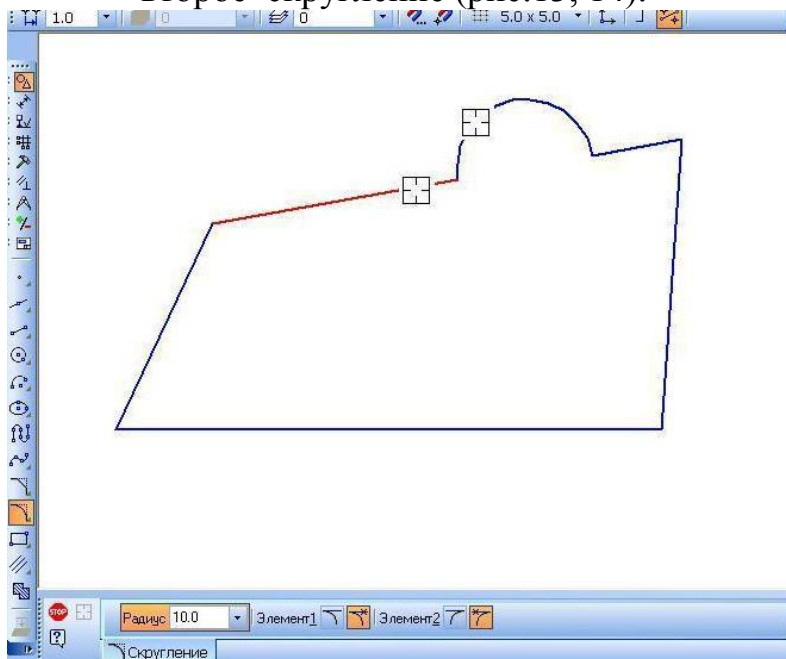


Рис. 13. Выделение линий для первого скругления.

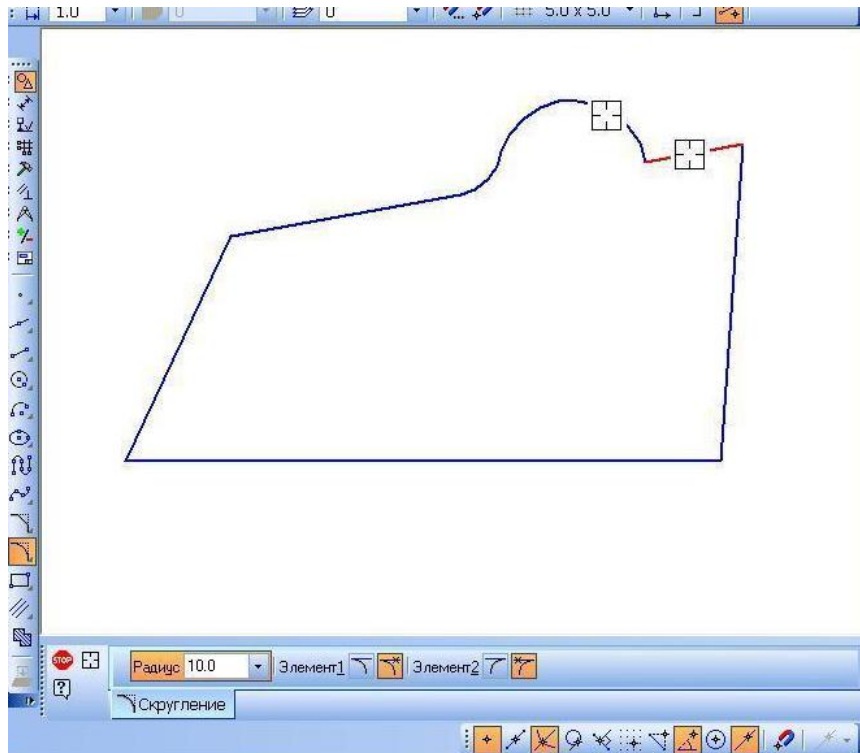


Рис. 14. Выделение линий для второго скругления.

39. Нанесите осевые линии бобышки. Для чего:
 Вызовите команду **Отрезок** (**Инструменты - Геометрия - Отрезки - Отрезок**). Подведите курсор вначале к левой ближайшей точке, потом ко второй точке (рис. 15).

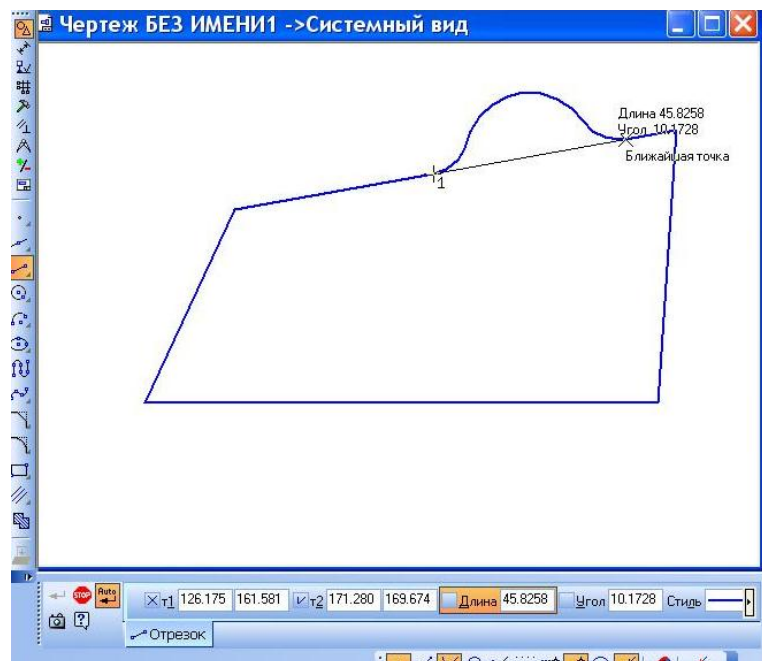


Рис. 15. Перед фиксированием фантома отрезка.

40. ЛКМ зафиксируйте отрезок. Прервите команду **Отрезок** (нажмите **Esc**).

Нанесите вторую осевую линию бобышки. Для чего:

41. Вызовите команду **Перпендикулярный отрезок** (**Инструменты - Геометрия - Отрезки - Перпендикулярный отрезок**).

42. "Ловушкой" выделите и ЛКМ подтвердите выбор исходного отрезка (рис. 16).

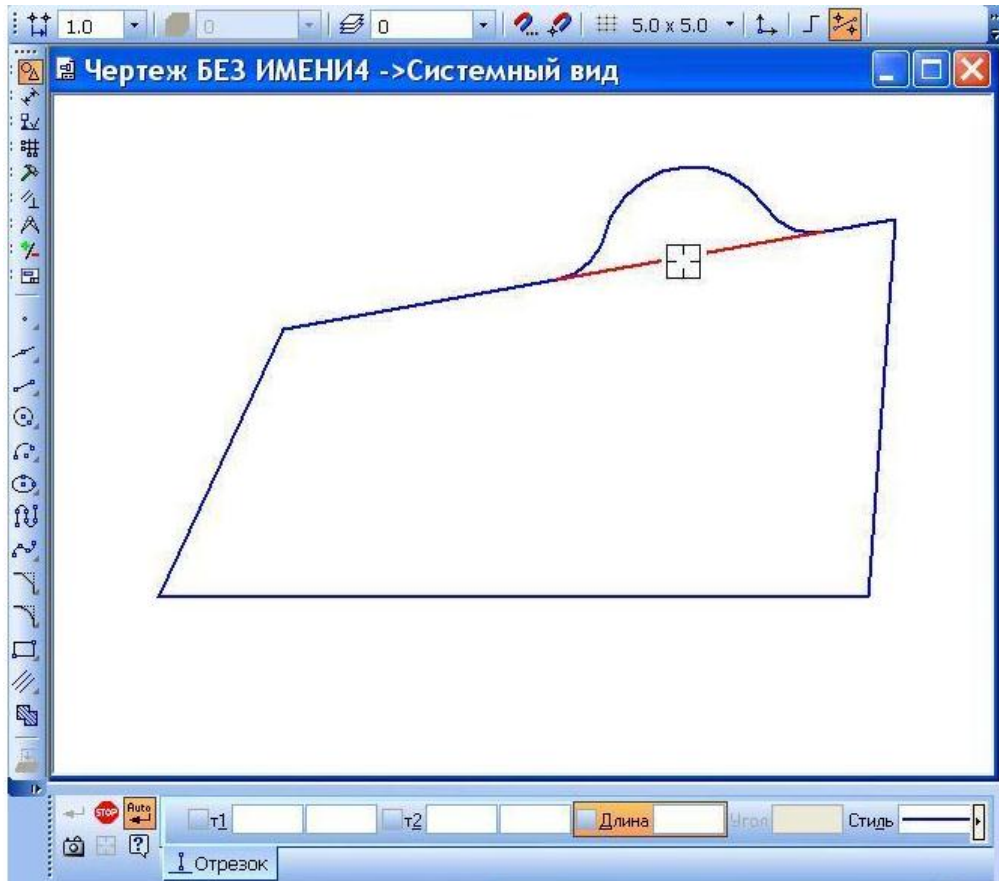


Рис. 16. Выделение отрезка, к которому будет проведен перпендикулярный отрезок перед нажатием ЛКМ.

43. Щелчком ЛКМ подтвердите выбор ближайшей точки (рис. 16).

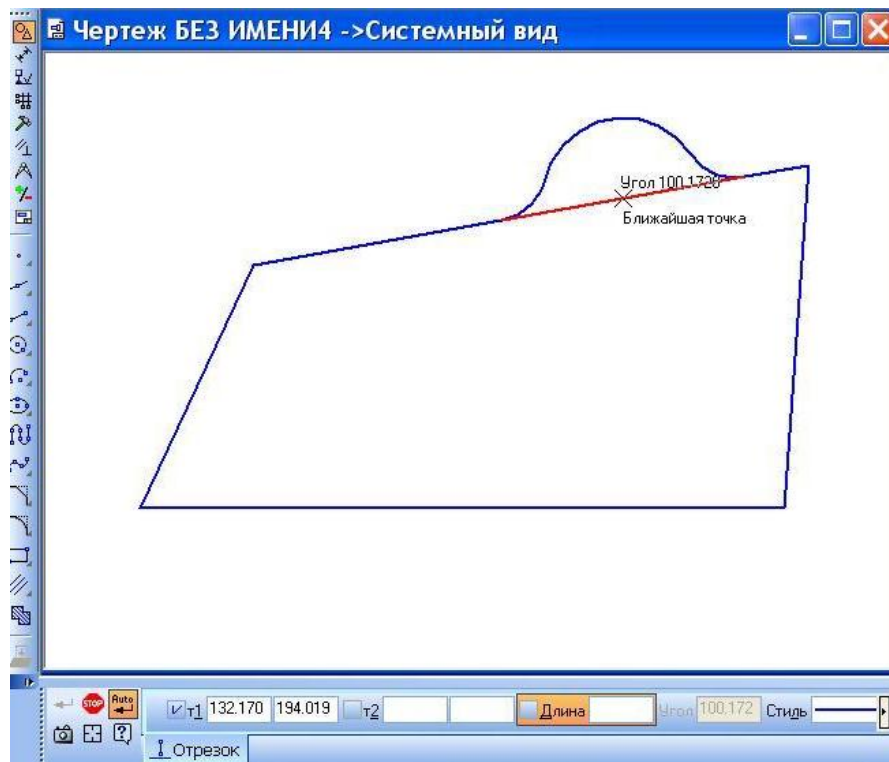


Рис. 16. Выбор ближайшей точки перед после фиксации ЛКМ.

44. Проведите вверх отрезок и щелчком ЛКМ зафиксируйте его ввод. Прервите команду (нажмите **Esc**).

45. Выделите оба отрезка и измените стиль линии с **Основная** на **Осевая** (рис. 17, 18)

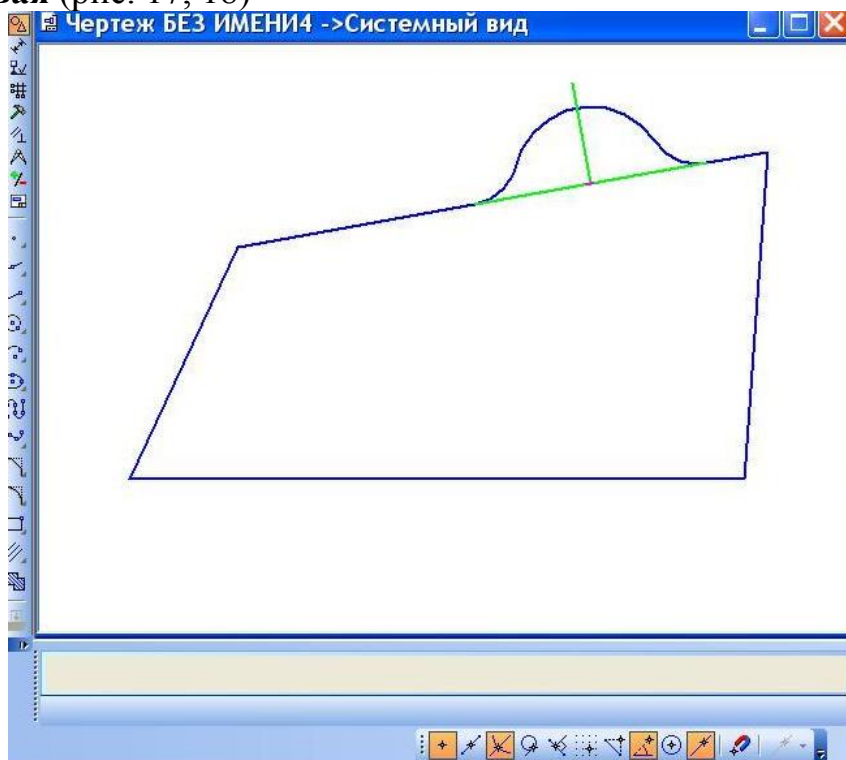


Рис. 17. Выделение отрезков.

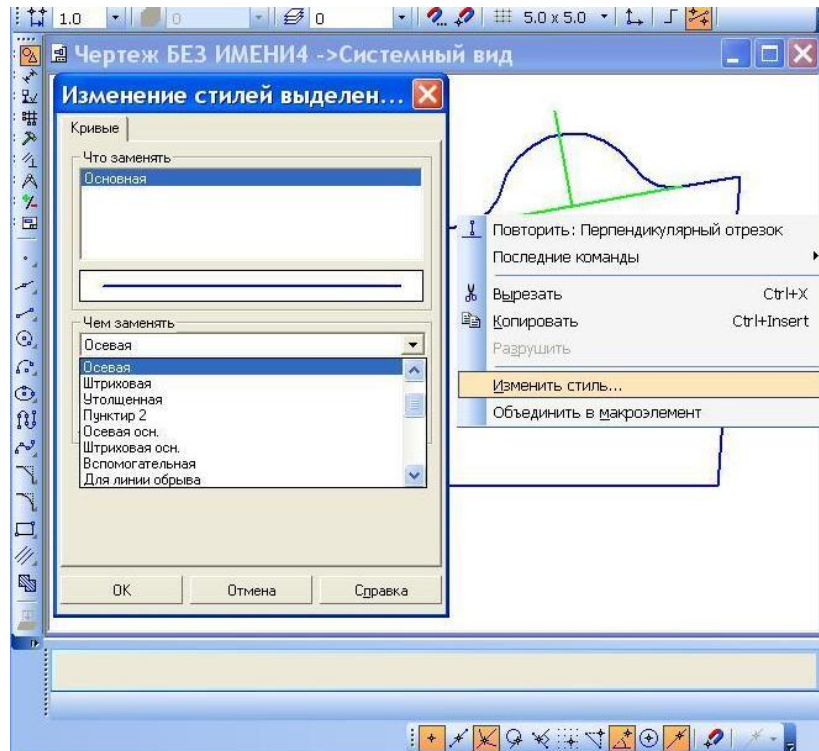



Рис. 18. Вызов контекстного меню для изменения стиля линии.

46. Через **Инструменты - Размеры - Линейные** вызовите команду **Линейный размер**.

47. На **Панели свойств** активизируйте **Тип**

48. Щелчком ЛКМ укажите точки 1 и 2 и  установите фантом размера в нужное место (рис. 19).

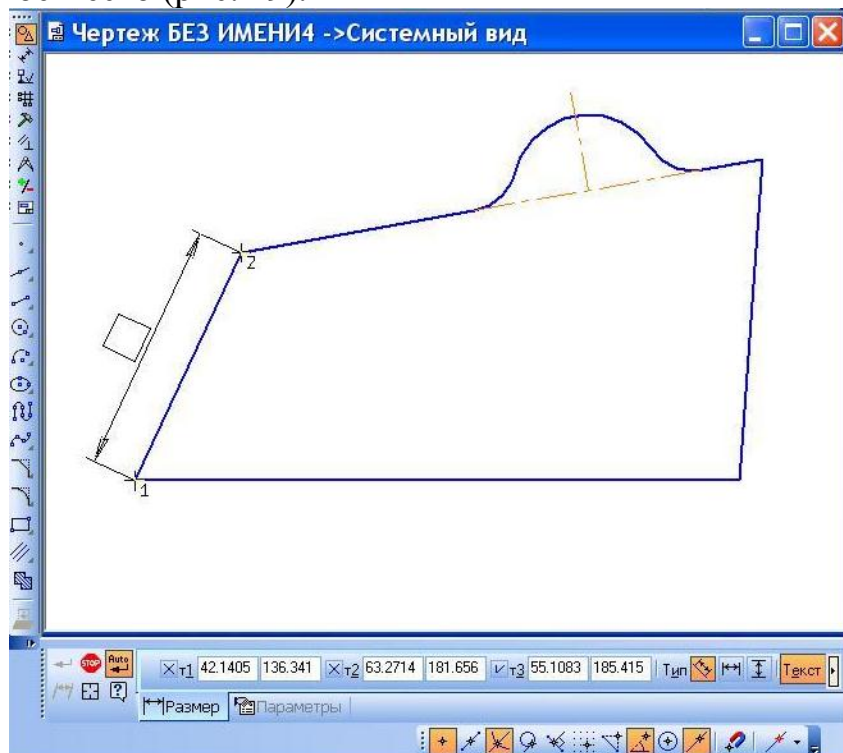


Рис. 19. Фантом размера перед вводом.

49. ЛКМ зафиксируйте размер.
50. Не прерывая команду, укажите точки и нанесите размеры (Рис. 20, 21).

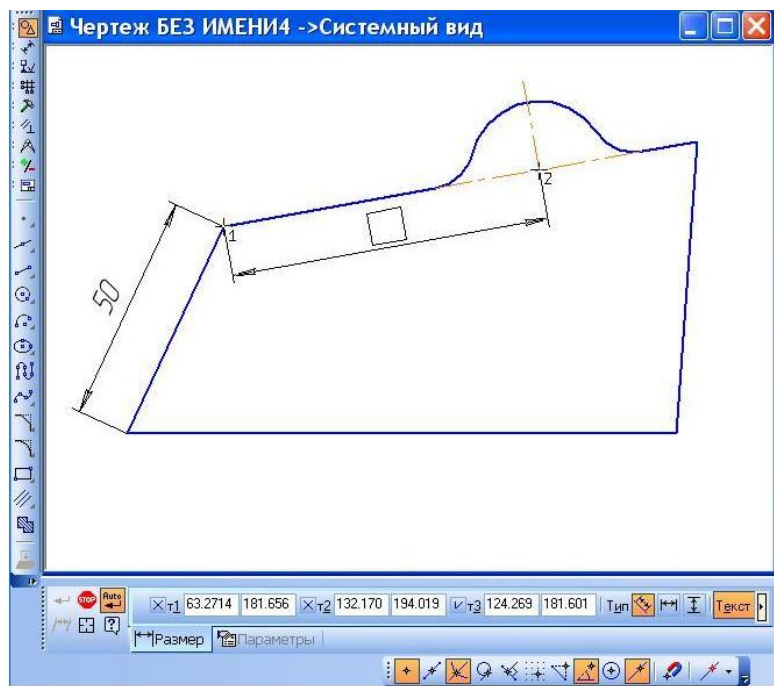


Рис. 20. Фантом размера перед вводом.

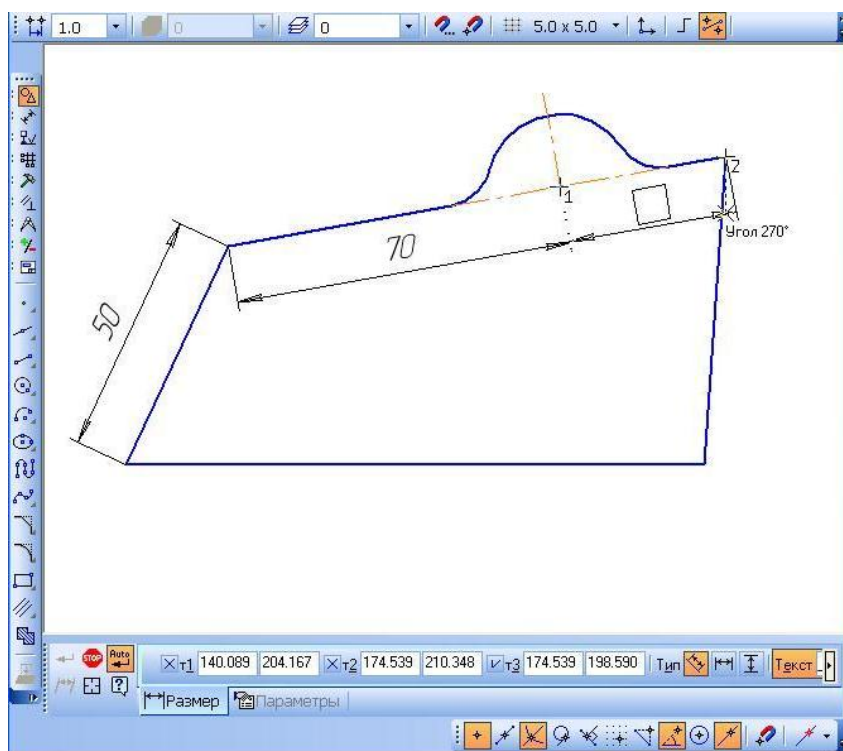


Рис. 21. Фантом размера перед вводом.

51. Прервите команду **Размеры** (нажмите **Esc**).
52. Вызовите команду **Ввод текста** (**Инструменты - Ввод текста**).

53. ЛКМ укажите расположение и введите поочередно буквы А, В, С, Е, D. Ввод каждой буквы завершите нажатием пиктограммы кнопки ввода на ПАНЕЛИ СВОЙСТВ.

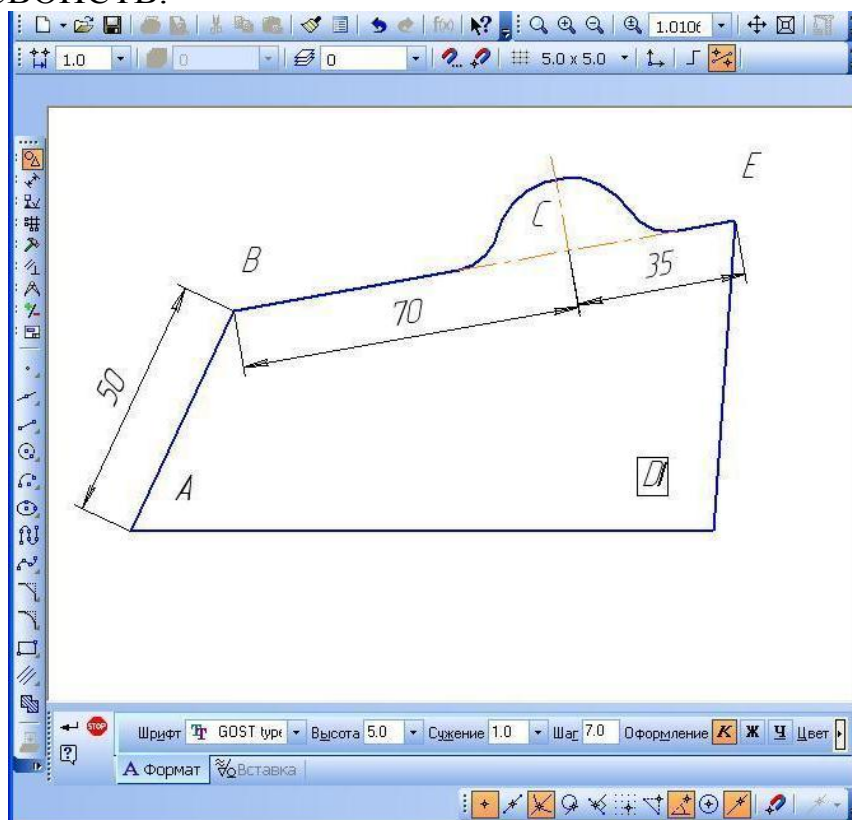


Рис. 22. Перед завершением ввода буквы D.

54. Прервите команду **Ввод текста** (нажмите **Esc**).

55. Вызовите команду **Радиальный размер (Инструменты-Размеры-Радиальные-Радиальный размер)**. Нанесите радиальные размеры. Прервите команду (нажмите **Esc**).

56. Вызовите команду **Угловой размер (Инструменты - Размеры - Угловые –Угловой размер)**. Нанесите угловой размер (65°). Прервите команду (нажмите **Esc**).

57. Нанесите недостающие линейные размеры.

58. Сохраните чертеж (Рис. 23) в своей папке.

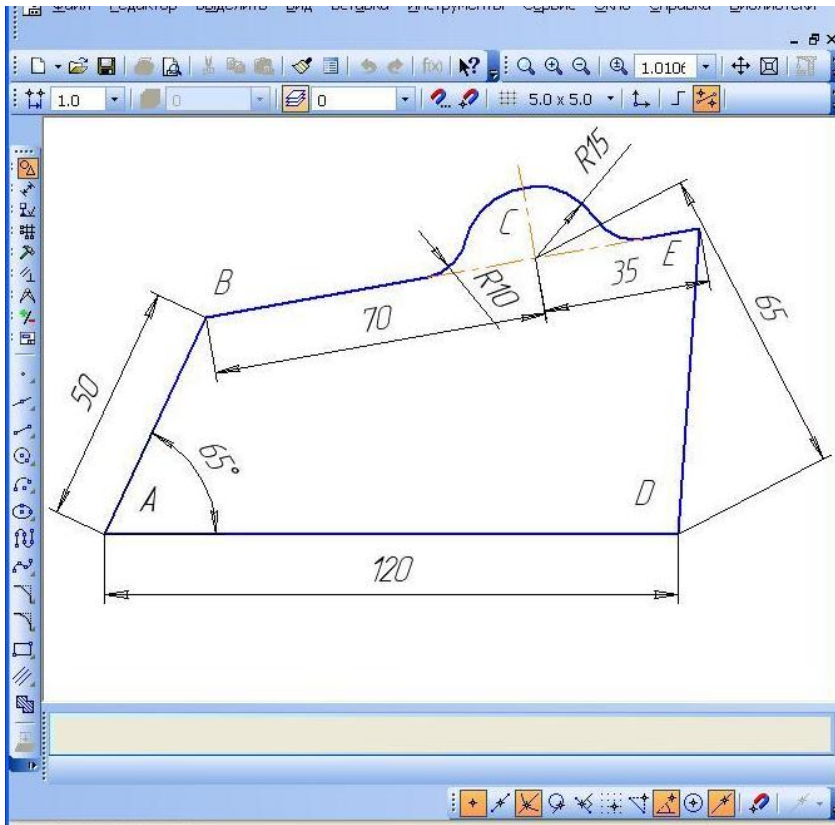


Рис. 23. Геометрическая фигура с нанесенными размерами.

59. Вызовите команду **Вид - Показать все** или нажмите F9. Переместите фигуру в верхнюю часть чертежа. Для этого:
60. Вызовите команду **Выделить рамкой** (пункт меню **Выделить-Рамкой**).
61. Нажмите ЛКМ, выделите фигуру рамкой (рис. 24).

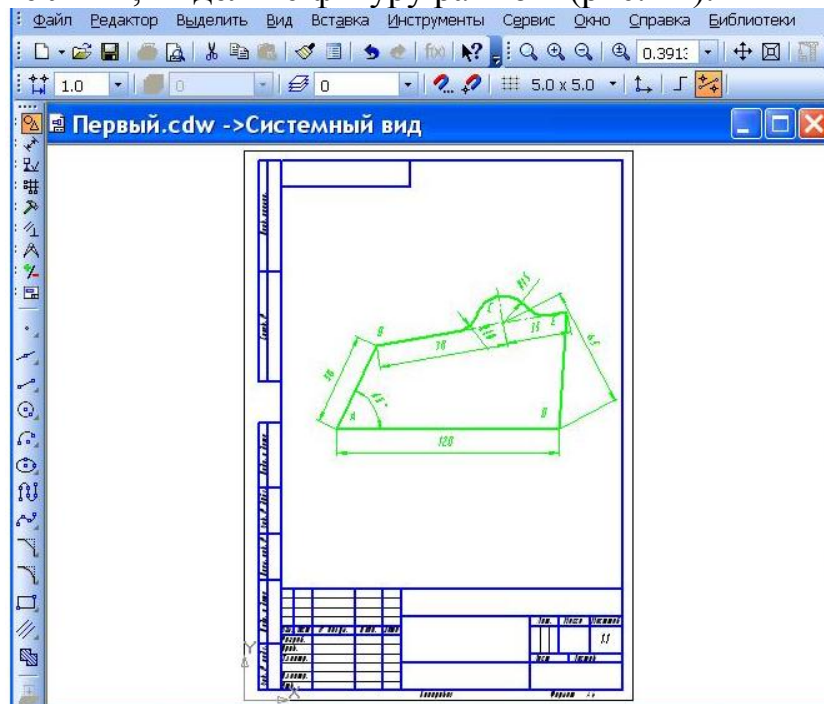


Рис. 24. Вид на чертеж после выделения фигуры рамкой.

62. Нажмите ЛКМ и переместите фигуру на новое место на листе (рис. 25, 26).

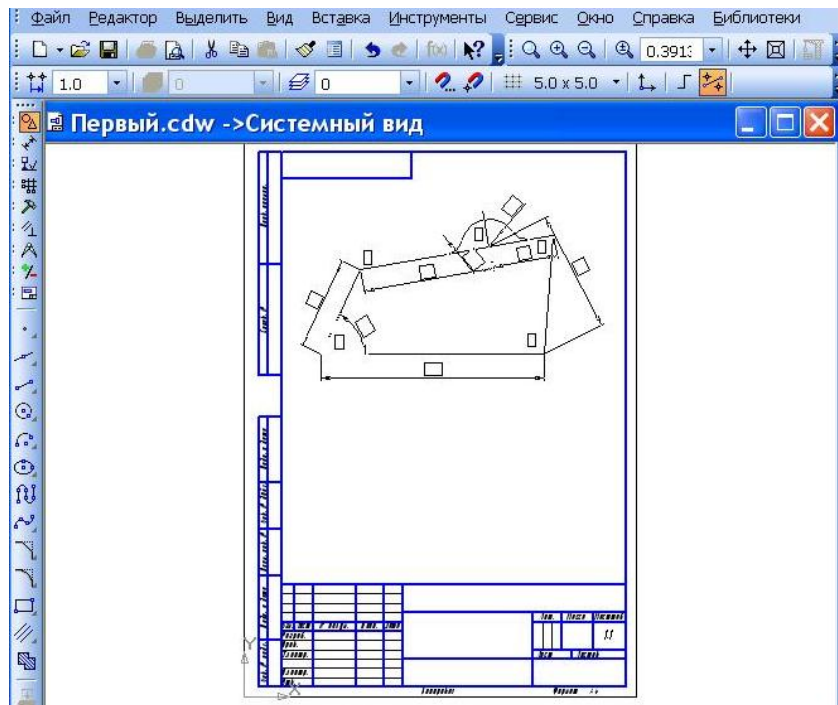


Рис. 25. Чертеж в момент перемещения фигуры в верхнюю часть листа.

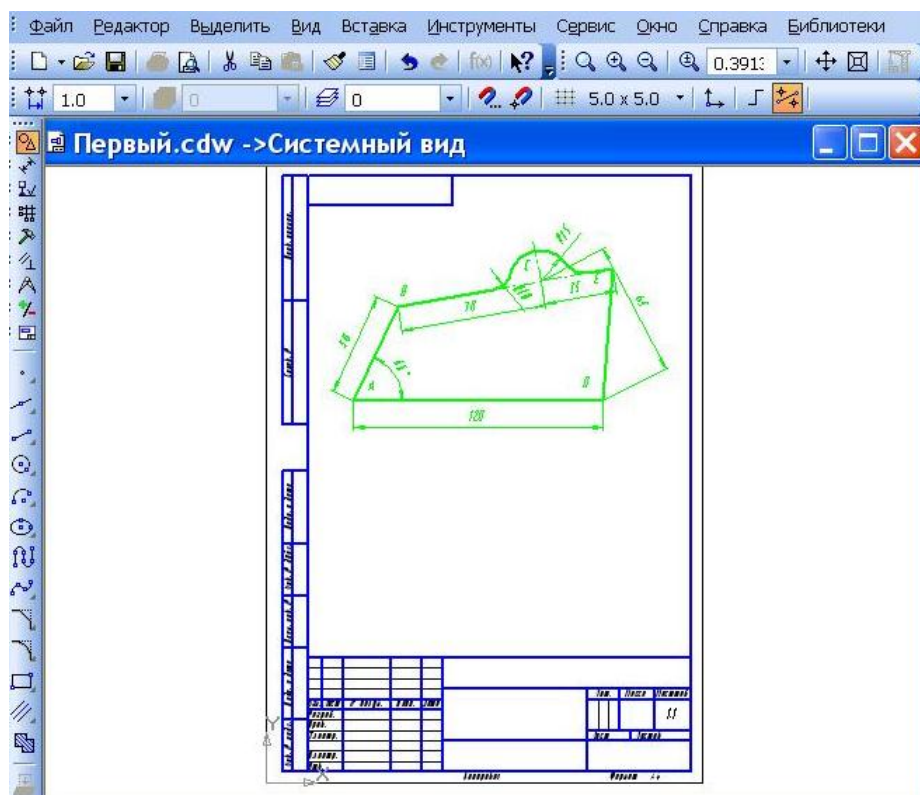


Рис. 26. Фигура установлена в требуемое место.

63. Произведите щелчок ЛКМ на поле чертежа. Сохраните чертеж.

Практическая работа №9. Разработка алгоритмов. Программирование с использованием языков высокого уровня. Линейный и разветвляющийся алгоритмы.

Цель: Сформировать навыки составления алгоритмов решения задач и написания программ.

Методические указания

Алгоритм – это определенная последовательность действий для решения задачи.

Способы записи алгоритма: словесная, графическая, программная.

Программирование – процесс разработки программ.

В случае, когда необходимо работать с набором данных, в программе применяют массивы.

Язык программирования – совокупность средств и правил записи алгоритма в форме, пригодной для выполнения вычислительной машиной.

Pascal (Паскаль) относится к процедурным языкам программирования высокого уровня.

Алфавит языка программирования Pascal включает заглавные и строчные латинские буквы, символ подчеркивания, цифры (от 0 до 9), специальные символы (+ - * / = > < . , ; : @ ' () [] { } # \$ ^).

Структура программы (основные разделы):

Program <имя программы>;

Uses <список имен модулей⁵>;

Label <имена меток⁶>;

Const <описание констант⁷>;

Type <описание пользовательских типов данных>;

Var <описание переменных>;

Procedure <имя процедуры⁸>;

Function <имя функции⁹>;

Begin

{последовательность операторов, разделенных точкой с запятой}

End.

Стандартные процедуры и функции

Стандартная функция	Выполняемое действие	Стандартная функция	Выполняемое действие
Abs(x)	x	Dec(x,n)	Процедура уменьшает значение x на n
Sqr(x)	x ²	Dec(x)	Процедура уменьшает значение x на 1
Sqrt(x)	√x	Inc(x,n)	Процедура увеличивает значение x на n

⁵ Подключение стандартных и пользовательских библиотечных модулей

⁶ Метка – совокупность идентификаторов, предназначенных для организации передачи управления операторам, отмеченных метками.

⁷ Константа – данные, не меняющие своего значения в ходе выполнения программы.

⁸ Процедура – способ оформления вспомогательного алгоритма (подпрограммы).

⁹ Функция – способ оформления вспомогательного алгоритма (подпрограммы).

Exp(x)	e^x	Inc(x)	Процедура увеличивает значение x на 1
Ln(x)	Ln(x)	Frac(x)	Функция вычисляет дробную часть x
pi	Число π	Int(x)	Функция вычисляет целую часть x
Sin(x)	Sin(x)	Trunc(x)	Функция возвращает ближайшее целое число, меньшее или равное вещественному x для $x \geq 0$, и больше или равное для $x < 0$
Cos(x)	Cos(x)	Round(x)	Функция возвращает значение x, округленное до ближайшего целого числа
Arctan(x)	Arctg(x)	Random (диапазон)	Функция возвращает случайное число x, удовлетворяющее условию $0 \leq x < \text{диапазон}$
		Random	Функция возвращает случайное число x, удовлетворяющее условию $0 \leq x < 1$

Вывод информации на экран осуществляется процедурами *write* и *writeln*.

Ввод информации осуществляется процедурами *read* и *readln*.

Условный оператор может записываться в полной и неполной формах:

<i>полная форма условного оператора</i>	<i>неполная форма условного оператора</i>
If <условие> Then <оператор 1> Else <оператор 2>;	If <условие> Then <оператор>;

При написании сложных условий простые условия заключаются в скобки.

Пример 1. Определить значение целочисленных переменных a и b после выполнения фрагмента программы:

```
a := 3 + 8*4;
b := (a div 10) + 14;
a := (b mod 10) + 2;
```

Решение: Выполним действия, предписываемые алгоритмом:

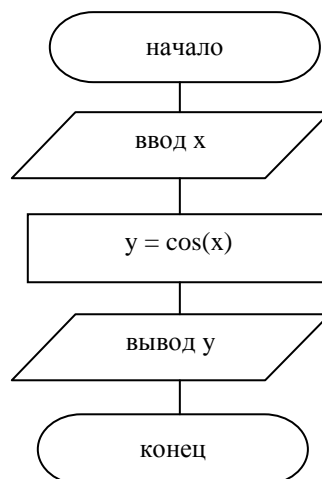
```
a = 3 + 8*4 = 35;
b = (35 div 10) + 14 = 3 + 14 = 17;
a = (17 mod 10) + 2 = 7 + 2 = 9;
```

Ответ: a = 9; b = 17.

Пример 2. Составить алгоритм вычисления значения функции $y = \cos(x)$. Значение переменной x вводится с клавиатуры.

Решение:

```
Program primer_2;
Var  x:integer;
     y: real;
Begin
writeln('введите число');
readln(x);
y := cos(x);
writeln('y= ', y)
End.
```



Пример 3. Составить алгоритм вычисления значения функции $y(x)$. Значение переменной x вводится с клавиатуры.

$$y = \begin{cases} x^2, & \text{если } x \leq 0 \\ \sqrt{x}, & \text{если } x > 0 \end{cases}$$

Решение:

```

Program primer_3;
Var x:integer;
    y: real;
Begin
writeln('введите число');
readln(x);
if x<=0 then y := sqr(x)
    else y:=sqrt(x);
writeln('y= ', y)
End.
  
```

Пример 4. Определить значение переменной c после выполнения следующего фрагмента программы:

```

a := 100; b := 30;
a := a - b*3;
if a > b then c := a - b else c := b - a;
  
```

Решение: Выполним действия, предписываемые алгоритмом:

```

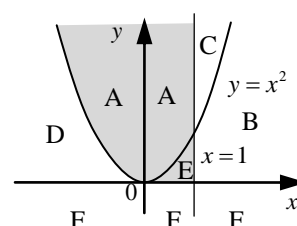
a = 100; b = 30;
a = 100 - 30*3 = 10;
if a > b then c := a - b else c := b - a;
  
```

Т.к. условие $a > b$ ложно, то выполняется оператор $c := b - a = 30 - 10 = 20$.

Ответ: 20.

Пример 5. Требуется написать программу, которая вводит с клавиатуры координаты точки на плоскости (x, y – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы.

Решение:



```

var x, y: real;
begin
readln(x, y);
if ((y >= 0)and (x <= 1) and (y >= x*x))
or ((y >= 0)and (x <= 1)and (x>=0)) then write('принадлежит')
else write('не принадлежит');
end.

```

Пример 6. Вывести на экран наибольшее из двух данных чисел.

Решение:

```

Program primer_1;
Var x, y : integer;
Begin
  WriteLn ( ' Введите 2 числа ' );
  ReadLn (x, y);
  If x>y Then WriteLn (x)
Else If x<y Then WriteLn (y)
      Else Writeln('Числа равны');
ReadLn
End.

```

Задания для выполнения

- 1) Определить значение целочисленных переменных a и b после выполнения фрагмента программы:


```

a := 6*12 + 3;
b :=(a div 10)+ 5;
a :=(b mod 10)+ 1;

```
- 2) Определить значение переменной c после выполнения следующего фрагмента программы:


```

a := 6; b := 15;
a := b - a*2;
if a > b then c := a + b
      else c := b - a;

```
- 3) С клавиатуры вводятся два действительных числа x , y . Требуется написать программу вычисления значения выражения

$$\frac{1 + \sin^2(x + y)}{2 + |x - 2x/(1 + x^2 y^2)} + x$$
- 4) Дано уравнение $a|x|+b=0$. Написать программу, находящую решение этого уравнения или выдающую сообщение, что решения не существует.

Практическая работа №10. Разработка алгоритмов. Программирование с использованием языков высокого уровня. Циклический алгоритм.

Цель: Сформировать навыки составления алгоритмов решения задач и написания программ.

Методические указания

При разработке алгоритмов решения большинства задач возникает необходимость многократного повторения однотипных команд. Для организации в программах повторов однотипных команд язык Паскаль предлагает специализированные операторы повтора, называемые *операторами цикла*.

Разновидности оператора *цикла*:

- 1) цикл с параметром
- 2) цикл с предусловием
- 3) цикл с постусловием.

Оператор цикла с параметром

For i:= A to B do <оператор>;

где *i* - параметр цикла (счетчик повторов)- переменная целого типа (integer);
A и *B* - начальное и конечное значения параметра цикла (выражения того же типа, что и параметр цикла);

оператор - любой простой или составной оператор, который требуется повторить несколько раз.

Оператор цикла типа For...to...do предусматривает последовательное увеличение на единицу параметра цикла "i" от начального значения "A" до конечного значения "B" и выполнение входящего в цикл алгоритма при каждом значении параметра цикла.

В качестве иллюстрации применения оператора цикла For...to...do рассмотрим пример вычисления суммы всех целых чисел от 1 до N (целое число N вводится с клавиатуры).

```
Program Primer;  
  Var i, N, S:integer;  
  Begin  
    Write('N= ');  
    Readln(N); {С клавиатуры ввели целое число в переменную "N"}  
    S:=0; {Задали начальное значение суммы}  
    For i:=1 to N do S:=S+i; { Во время каждого из повторов }  
    {значение суммы "S" увеличивается на новую величину счетчика "i" }  
    Writeln('S= ', S:6);  
    Readln  
  End.
```

В этой программе оператор "S:=S+i" выполняется "N" раз, при различных значениях параметра цикла "i".

В некоторых случаях бывает удобно, чтобы параметр цикла принимал последовательно убывающие, а не возрастающие значения. Для этого предусмотрена следующая разновидность оператора цикла:

For i:=B downto A do <оператор>;

где *i*, *A* и *B* имеют прежний смысл. Отличие от предыдущего варианта цикла в том, что в операторе цикла типа For...downto...do шаг наращивания параметра равен -1, при этом начальное значение счетчика повторов "*B*" больше конечного значения "*A*".

Пример при нисходящем изменении значения параметра цикла:

```
Program Primer;
Var i,N,S:integer;
Begin
  Write('N= '); Readln(N);
  S:=0; {начальное значение суммы}
  For i:=N downto 1 do S:=S+i;
  Writeln('S= ',S:6);
  Readln
End.
```

Оператор цикла с предусловием

Если число повторений, выполняемых в цикле, заранее не известно или шаг приращения счетчика (параметра) цикла отличен от единицы, то необходимо использовать оператор цикла с предусловием. Оператор цикла этого вида имеет вид:

While <условие> do <оператор> ;

где *условие* - это логическое выражение, от значения которого зависит - продолжать повторы или завершить цикл;

оператор - любой простой или составной оператор.

Выполнение оператора начинается с вычисления значения логического выражения. Если оно имеет значение "True" (истина), то выполняется оператор (операторы), входящий в цикл. Выполнение цикла продолжается до тех пор, пока логическое выражение в его заголовке не примет значение "False" (ложно). Если выражение равно "False" при первом же витке цикла, то работа цикла завершится, а входящие в него операторы не выполнятся ни разу. Поскольку в цикле типа While...do условие завершения его работы проверяется до выполнения входящего в него оператора, такой цикл называется "оператор цикла с предусловием".

Пример использования оператора цикла с предусловием:

```
Program Primer;
Var K:integer;
Begin
  K:=0;
  While K<=10 do
    begin
```

```
    K:=K+2;
    Write('K= ',K:3)
end;
Readln
End.
```

Оператор цикла с постусловием

Цикл этой разновидности применяется в случаях, когда число повторений оператора, входящего в тело цикла, заранее не известно. Такой цикл похож на цикл с предусловием, но в данном случае условие завершения повторов проверяется после выполнения операторов, составляющих тело цикла. Общий вид оператора цикла с постусловием таков:

Repeat

оператор1, оператор2, ... , операторN

Until <условие>;

где оператор1, оператор2, ... , операторN - операторы тела цикла;
условие - логическое выражение, диктующее завершение повторов.

Оператор цикла с постусловием начинается с выполнения операторов внутри цикла. Затем проверяется истинность логического условия, стоящего после слова **Until**. Если это условие справедливо (**True**), то осуществляется выход из цикла. Если же значение логического выражения ложно (**False**), то выполнение операторов тела цикла повторяется, после чего снова проверяется истинность логического условия.

Пример программы, использующей оператор цикла с постусловием:

```
Program Primer;
Var K:Integer;
Begin
  K:=0;
  Repeat
    K:=K+2;
    Write('K= ',K:3)
  Until K>10;
  Readln
End.
```

При использовании операторов цикла следует учитывать следующие особенности:

- Операторы, входящие в цикл **Repeat...Until**, всегда выполняются хотя бы один раз, поскольку истинность логического выражения в цикле этого типа проверяется после операторов, входящих в тело цикла.

- При использовании цикла типа **While...do** могут быть ситуации, когда операторы, входящие в цикл, не будут выполнены ни разу, если логическое выражение изначально имеет значение **"False"**.

- Цикл Repeat...Until выполняется, пока логическое выражение имеет значение "False". Цикл While...Do выполняется, пока логическое выражение имеет значение "True".

<pre> Program W; Var i:integer; Begin i:=1; While i<=10 do begin Writeln('Привет'); i:=i+1 end; Readln End.</pre>	<pre> Program R; Var i:integer; Begin i:=1; Repeat Writeln('Привет'); i:=i+1 Until i>10; Readln End.</pre>
--	---

Если тело цикла "While...Do" состоит из нескольких операторов, их следует обрамлять операторными скобками "begin...end", образующими составной оператор. В цикле типа "Repeat...Until" операторные скобки не нужны.

Пример 1. Вычислить и вывести на экран значения функции $Z = \sin x / x$ для $x \in [0.1, 1]$ с шагом 0.1. Для решения задачи использован цикл с постусловием.

```

Program tabulirovanie_1;
Var x, dx, z: real;
Begin
  x:=0.1;  xk:=1;  dx:=0.1;
  z:= 0;
  Repeat
    z:= sin(x)/x;
    Writeln('x = ', x:5:2, 'z = ', z:7:3);
    x:= x+dx
  Until x>xk;
  Readln
End.
```

Пример 2. Вычислить и вывести на экран значения функции $Z = \sin x / x$ для $x \in [0.1, 1]$ с шагом 0.1. Для решения задачи использован цикл с предусловием.

```

Program tabulirovanie_2;
Var x, dx, z: real;
Begin
  x:=0.1;  xk:=1;  dx:=0.1;
  z:= 0;
  While x<xk+dx do
```

```

Begin
  z:= sin(x)/x;
  Writeln('x = ', x:5:2, 'z = ',z:7:3);
  x:= x+dx
End;
Readln
End.

```

Пример 3. Вычислить сумму целых чисел от 1 до N. Значение N вводится с клавиатуры. Для решения задачи использован цикл с параметром (счетчиком).

```

Program summa;
Var i, N, S: integer;
Begin
  Write('N= ');
  Readln(N); {С клавиатуры ввели целое число в переменную N}
  S:=0; {Задали начальное значение суммы}
  For i:=1 to N do S:=S+i; { Во время каждого из повторов }
  {значение суммы S увеличивается на новую величину счетчика i }
  Writeln('S= ', S:6);
  Readln
End.

```

Задания для выполнения

1. Определить, что будет выведено в результате работы фрагмента программы:

```

var n, s: integer;
begin
  n := 1; s := 0;
  while s <= 365 do begin
    s := s + 36; n := n * 2
  end; write(n)
end.

```

2. Определить значение переменной S после выполнения следующего фрагмента программы:

```

S:=0; i:=7;
while i > 1 do begin
  S := S + i div 2;
  i := i - 1;
end;

```

3. С клавиатуры вводятся N чисел. Составьте программу, которая определяет количество отрицательных, количество положительных и количество нулей среди введенных чисел. Значение N вводится с клавиатуры.
4. Вывести те из двузначных чисел, которые делятся на 4, но не делятся на 6.
5. Найти все двузначные числа, состоящие из одинаковых цифр.
6. Вычислить значения функции $y = \frac{x^2 - 3x + 2}{\sqrt{2x^3 - 1}}$ для $x = 1, 1.1, 1.2, \dots, 5$.
7. Дано натуральное N. Вычислить: $1/2^2 + 1/4^2 + \dots + 1/(2N)^2$.
8. Составить программу, определяющую максимальное из всех введенных пользователем чисел.
9. Дано натуральное число N. Составить программу, подсчитывающую количество и сумму цифр в записи числа N.

Практическая работа №11. Разработка алгоритмов. Программирование с использованием языков высокого уровня. Работа с массивами.

Цель: Сформировать навыки составления алгоритмов решения задач и написания программ.

Методические указания

Массив – это последовательность данных, состоящая из фиксированного числа элементов, имеющих один и тот же тип, и обозначаемая одним именем.

Массиву присваивается имя, посредством которого можно ссылаться на него, как на единое целое. Элементы массива упорядочены так, что каждому элементу соответствует совокупность номеров (индексов), определяющих его место в общей последовательности. Индексы представляют собой выражения порядкового типа (целочисленные, символьный, логический типы данных). Доступ к каждому отдельному элементу осуществляется обращением к имени массива с указанием индекса нужного элемента в квадратных скобках после имени массива: *<имя массива>[<индекс>]*. Индексы массива сами могут быть переменными или выражениями, обеспечивая доступ не к одному, а к последовательности элементов. Сами элементы массива иначе называют переменными с индексами (индексированными переменными).

Характеристики массива:

- 1) *тип* – общий для всех элементов массива тип;
- 2) *размерность* (ранг) – количество индексов массива;
- 3) *диапазон изменения индексов* (индекса) – определяет количество элементов в массиве.

Виды массивов:

- 1) *одномерный* (линейный, вектор), элементы которого нумеруются одним индексом;
- 2) *двумерный* (матрица), элементы которого нумеруются двумя индексами;
- 3) *трех-, четырехмерные* массивы.

Способы *описания* массивов:

- 1) предварительное описание типа массива

Type <имя типа >=**array**[<тип индекса>] **of** <тип элементов>;

Далее, в перечне переменных указывается имя массива, и через двоеточие указывается имя нового типа данных: **Var** <имя массива>:<имя типа >;

Пример: Type m=array[1..10,1..20] of real;

Var matrix: m;

- 2) в разделе описания типов данных:

Var <имя массива>: **array** [<тип индекса>] **of** <тип элементов>;

Чаще всего в качестве типа индекса используется интервальный целый тип.

Пример: Var a:array[1..10] of char;

b: array[1..3,1..3] of word;

- 3) в качестве типизированной константы в разделе описания констант:

Пример: Const x: array[1..5] of integer =(1,3,5,7,9);

Массив, описанный как типизированная константа, уже содержит данные. В остальных случаях массивы нужно заполнить данными, прежде чем начать выполнять с ними действия.

Способы *задания* значений элементов массива:

- 1) ввод данных с клавиатуры

Пример: а) for i:=1 to 5 do read(a[i]);

б) for i:=1 to 3 do

for j:=1 to 4 do readln(a[i,j]);

- 2) с помощью датчика случайных чисел

Пример: randomize;

for i:=1 to 5 do a[i]:=random(50);

- 3) присваиванием заданных значений

Пример: for i:=1 to 3 do

for j:=1 to 4 do a[i,j]:=1;

4) считывание значений элементов из файла.

Способы *вывода* значений элементов массива:

1) вывод элементов массива в одну строку. Пример: for i:=1 to 5 do
write(a[i]:3);

2) вывод элементов массива в столбец. Пример: for i:=1 to 5 do
writeln(a[i]);

3) вывод элементов матрицы

Пример: for i:=1 to 3 do
begin
for j:=1 to 4 do writeln(a[i,j]:4);
writeln
end;

Алгоритм решения задач с использованием массивов:

- Описание массива
- Заполнение массива
- Вывод (распечатка) массива
- Выполнение условий задачи
- Вывод результата

ПОИСК И СОРТИРОВКА МАССИВОВ

Сортировка – процесс упорядочения заданной последовательности объектов по определенному признаку.

Виды сортировки:

- 1) внутренняя (сортировка массивов);
- 2) внешняя (сортировка элементов файлов).

Алгоритмы сортировки:

- сортировка методом простого обмена (сортировка методом «пузырька»);
- сортировка методом простого выбора (линейная сортировка);
- сортировка методом слияний;
- обменная сортировка с разделением (сортировка Хоара) и др.

Пример сортировки одномерного массива в порядке возрастания значения элементов методом «пузырька» (основная мысль алгоритма: сравниваются два соседних элемента, если порядок нарушен, то они меняются местами).

```
Program sort;  
Const n=15;  
Var a:array[1..n] of integer; i,k,buf:integer;
```



```

Begin
  Randomize;
  Write('Исходный массив ');
  For i:=1 to n do
    Begin
      a[i]:=random(100);
      Write(a[i]:5)
    End;
  Writeln;
  For k:=1 to n do
    For i:=1 to n-1 do if a[i]>a[i+1] then begin
      buf:=a[i];
      a[i]:=a[i+1];
      a[i+1]:=buf
    end;

    Write('Упорядоченный массив ');
    For i:=1 to n do write(a[i]:5)
  End.

```

Поиск – процесс нахождения в заданном множестве объекта, обладающего свойствами задаваемого априори эталона.

Алгоритмы поиска:

- линейный поиск;
 - линейный поиск с использованием барьера;
- бинарный поиск (поиск делением пополам) и пр.

МАССИВЫ ОДНОМЕРНЫЕ

Пример 1. Задан одномерный массив B из 10 целых чисел. Подсчитать количество элементов массива, больших заданного числа k .

```

Program massiv_1;
Var b:array [1..10] of integer; i, k, s : integer;
Begin
  s:=0;
  For i:=1 to 10 do
    Begin
      Write('Введите', i, '-й элемент массива ');
      Readln (b[i]);
      Write(b[i], ' ');
    End;
  Write('Введите число k');
  Readln(k);
  For i:=1 to 10 do if b[i]>k then s:=s+1; Write('Количество элементов', s);

```

End.

Пример 2. Определить самую высокую температуру и самый теплый день в мае.

```
Program massiv_2;
Var t:array [1..31] of integer;
    i, max, n : integer;
Begin
  For i:=1 to 31 do
    Begin
      t[i]:=random(20);
      Write(b[i], ' ');
    End;
  max:= t[1];
  n:=1;
  For i:=2 to 31 do
    Begin
      If t[i] > max then max:=t[i];
      n:=i ;
    End;
  Write('максимальная температура ', max, ' в ', n, ' день');
End.
```

Пример 3. Сортировка одномерного массива A из 15 целых чисел в порядке возрастания значения элементов методом «пузырька». Основная мысль алгоритма: сравниваются два соседних элемента, если порядок нарушен, то они меняются местами.

```
Program sort;
Const n=15;
Var A:array[1..n] of integer;
    i,k,buf:integer;
Begin
  Randomize;
  Write('Исходный массив ');
  For i:=1 to n do
    Begin
      A[i]:=random(100);
      Write(A[i]:5)
    End;
  Writeln;
  For k:=1 to n do
    For i:=1 to n-1 do if A[i]>A[i+1] then
      begin
```

```

        buf:=A[i];
        A[i]:=A[i+1];
        A[i+1]:=buf
    end;
    Write('Упорядоченный массив ');
    For i:=1 to n do write(A[i]:5)
End.

```

МАССИВЫ ДВУМЕРНЫЕ (МАТРИЦЫ)

Пример 4. Сформировать таблицу умножения (в виде двумерного массива) и вывести ее на экран.

```

Program primer_4;
Var A: array[1..9,1..9] of integer;
    i, k : integer;
Begin
    For i := 1 to 9 do
        For k := 1 to 9 do A[i, k] := i * k;
    For i := 1 to 9 do
        Begin
            For k := 1 to 9 do Write(A[i, k] : 5);
            Writeln;
        End;
    End.

```

Пример 5. Дан двумерный массив A[1..3,1..5]. Вычислить сумму его элементов. Организовать ввод элементов матрицы с клавиатуры.

```

Program primer_5;
Const n = 3; m = 5;
Var A: array[1..n,1..m] of integer;
    i, k, s : integer;
Begin
    Writeln('Введите 15 целых чисел ');
    For i := 1 to n do
        For k := 1 to m do read(A[i, k]); { Ввод значений элементов матрицы }
        s := 0;
        For i := 1 to n do
            For k := 1 to m do s := s + A[i, k]; { Вычисление суммы элементов
матрицы }
        Writeln('сумма элементов матрицы = ', s)
    End.

```

Пример 6. Дан двумерный массив $A[1..3,1..5]$. Вычислить произведение его элементов. Организовать заполнение матрицы с помощью функции генерации случайных чисел (Random).

```
Program primer_6;
Const n = 3; m = 5;
Var A: array[1..n,1..m] of integer;
    i, k, p : integer;
Begin
  Randomize;
  For i := 1 to n do
    For k := 1 to m do A[i, k] := Random(10)+1; { Заполнение матрицы }
  Writeln( ' Исходная матрица ' );
  For i := 1 to n do { Вывод матрицы в прямоугольной форме }
    Begin
      For k := 1 to m do Write(A[i, k] : 5);
      Writeln;
    End;
  p := 1;
  For i := 1 to n do
    For k := 1 to m do p := p*A[i, k]; { Вычисление произведения
элементов }
    Writeln( 'произведение элементов матрицы = ', p)
  End.
```

Задания для выполнения

1. Найти произведение элементов одномерного массива из 10 целых чисел, которые кратны 7 и имеют нечетные номера (индексы).
2. В одномерном массиве из 15 целых чисел заменить отрицательные элементы на положительные.
3. Дан одномерный массив A из 30 действительных чисел. Каждый элемент массива - это количество осадков (в миллиметрах), выпавших в течение месяца. Вычислить среднее количество осадков и отклонение от среднего для каждого дня.
4. В одномерном массиве из 15 целых чисел поменять местами минимальный и максимальный по значению элементы.
5. Дан массив $A[1..5, 1..3]$. Вывести номера его отрицательных элементов.
6. Вычислить сумму элементов каждой строки массива $A[1..3,1..3]$. Вывести

на экран строку с наибольшей суммой элементов

7. Дан целочисленный массив $B[1..5, 1..5]$. Упорядочить по возрастанию значений элементы, расположенные на главной диагонали матрицы.

Практическая работа №12. Применение компьютера в решении задач.

Цель: Сформировать навыки применения компьютера в решении задач.

Методические указания

Основные методы решения задач:

- *аналитические* (точные): решение задачи получено в результате выполнения математических преобразований исходных формул (зависимостей);
- *графические*: решение задачи получено путем построения соответствующего чертежа (графика функции и т.п.);
- *численные* (приближенные): решение задачи получено путем выполнения многочисленных вычислений с помощью компьютера. Численный метод реализуется всегда в виде вычислительного алгоритма.

Не всегда возможно решить задачу аналитическим методом, графический метод дает большую погрешность. В этих случаях применяют численные (приближенные) методы.

Численные методы применяются в различных прикладных направлениях. Численные методы позволяют свести решение задачи к выполнению конечного количества арифметических действий над числами, при этом результаты получаются в виде числовых значений.

В качестве примера рассмотрим задачу о приближённом нахождении значения определённого интеграла

$$I = \int_a^b f(x) dx.$$

Функция $f(x)$ непрерывна на отрезке интегрирования и имеет на этом отрезке производные до некоторого порядка.

Вычислять значение интеграла I будем по значениям функции $f(x)$ в некоторых точках отрезка x_i . Эти значения $y_i = f(x_i)$ будем предполагать известными, то есть предполагать, что есть некоторый эффективный способ вычисления значений функции с любой требуемой точностью. Формулы, позволяющие по известным значениям y_i приближённо определить значение I , называются *квадратурными формулами*.

Для наглядности прибегнем к геометрической интерпретации смысла определённого интеграла, как площади некоторой криволинейной трапеции, в случае функции $f(x) \geq 0$.

При $f(x) \geq 0$ вычислить интеграл I значит найти площадь под графиком $y = f(x)$, расположенную над отрезком $[a;b]$. Разобьём отрезок на части точками деления x_1, x_2, \dots, x_{n-1} и положим $x_0=a$ и $x_n=b$. Тогда разбиение отрезка $[a;b]$ состоит из отрезков $[x_{i-1};x_i]$ при $i = 1, \dots, n$. Вместо площади под графиком, равной I , будем приближённо находить суммарную площадь узких полосок, лежащих над отрезками разбиения $[x_{i-1};x_i]$ (см. рис.1).

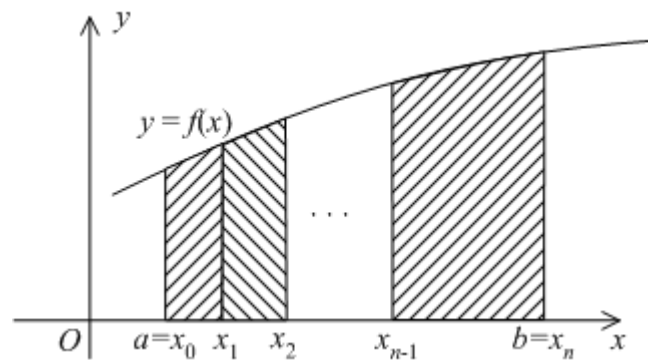


Рис. 1

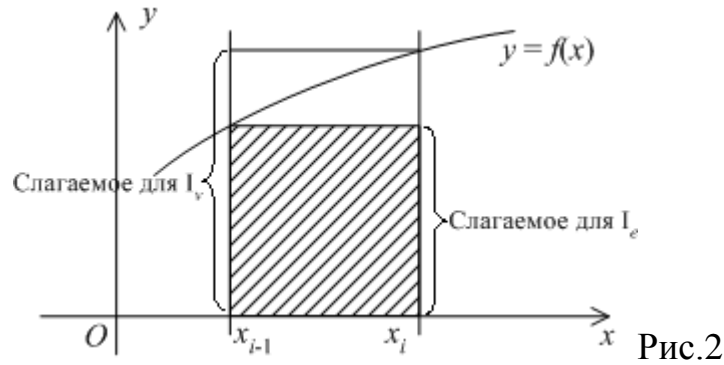
Самый простой метод приближённого вычисления площадей узких полосок – заменить их площадями S_i прямоугольников, основанием которых служит отрезок $[x_{i-1};x_i]$ на оси Ox , а высотой – отрезок, задающий значение функции в одном из концов основания, то есть либо в точке x_{i-1} , либо в точке x_i . Тогда в первом случае площадь $S_i = f(x_{i-1})(x_i - x_{i-1})$, а во втором $S_i = f(x_i)(x_i - x_{i-1})$.

Суммируя по всем отрезкам разбиения, то есть по i от $i = 1$ до $i = n$, получаем в первом случае *квадратурную формулу левых прямоугольников*:

$$I \approx I_l = \sum_{i=1}^n f(x_{i-1})(x_i - x_{i-1}),$$

а во втором случае *квадратурную формулу правых прямоугольников*:

$$I \approx I_r = \sum_{i=1}^n f(x_i)(x_i - x_{i-1}).$$



Снова рассмотрим отрезки разбиения $[x_{i-1}; x_i]$, где $i = 1, \dots, n$ и $x_0 = a$, $x_n = b$, и выберем в качестве точек разметки середины каждого из этих отрезков, т.е. точки

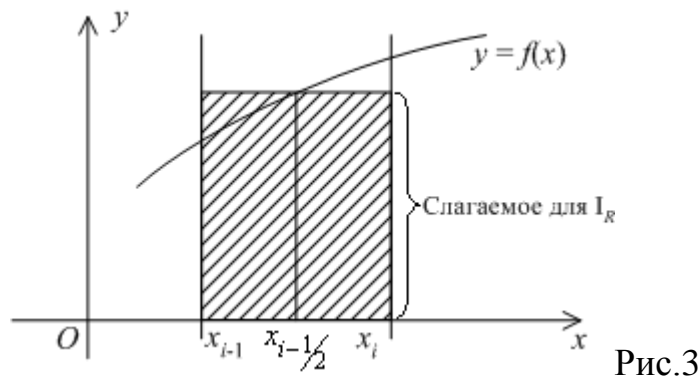
$$\bar{x}_i = \frac{1}{2}(x_{i-1} + x_i).$$

(будем эти середины обозначать $x_{i-1/2}$.)

Возьмём за приближённое значение интеграла интегральную сумму, построенную по такому размеченному разбиению. Каждое слагаемое в этой сумме, равно

$$S_i = f(x_{i-1/2})(x_i - x_{i-1}),$$

выражает площадь прямоугольника с основанием $[x_{i-1}; x_i]$ и высотой, равной значению функции в середине этого отрезка (см. рис. 3):



Получим тогда квадратурную формулу:

$$I \approx I_R = \sum_{i=1}^n f(x_{i-1/2})(x_i - x_{i-1}),$$

называемую *формулой центральных (средних) прямоугольников*.

Если взять все отрезки разбиения равной длины $h = \frac{b-a}{n}$, то эта квадратурная формула принимает вид

$$I \approx I_R = h \sum_{i=1}^n f(x_{i-\frac{1}{2}}).$$

В этом случае $x_{i-\frac{1}{2}} = x_i - \frac{h}{2} = a + ih - \frac{h}{2}$

Пусть снова взято разбиение отрезка $[a;b]$ на части $[x_{i-1};x_i]$, $i = 1, \dots, n$. Приблизённо заменим площадь под графиком $y = f(x)$, лежащую над промежутком разбиения $[x_{i-1};x_i]$, на площадь трапеции, параллельными основаниями которой служат отрезки, задающие значения функции в концах промежутка, то есть $f(x_{i-1})$ и $f(x_i)$ (см. рис.4).

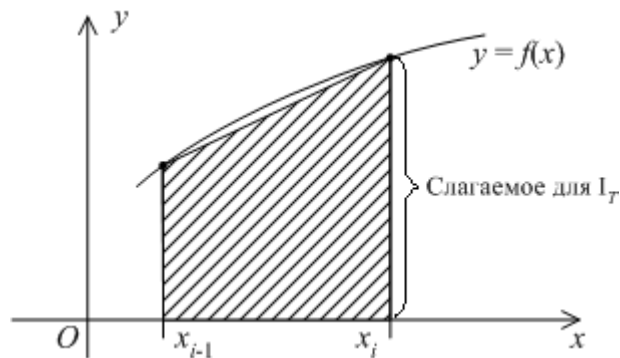


Рис.4

Тогда площадь такой трапеции равна

$$S_i = \frac{1}{2}(f(x_{i-1}) + f(x_i))(x_i - x_{i-1}).$$

Суммируя все площади S_i , получаем квадратурную формулу трапеций:

$$I \approx I_T = \frac{1}{2} \sum_{i=1}^n (f(x_{i-1}) + f(x_i))(x_i - x_{i-1}).$$

Если все отрезки разбиения выбираются одинаковой длины $h = \frac{b-a}{n}$, то формула трапеций приобретает вид

$$I \approx I_T = \frac{h}{2} \sum_{i=1}^n (f(x_{i-1}) + f(x_i)).$$

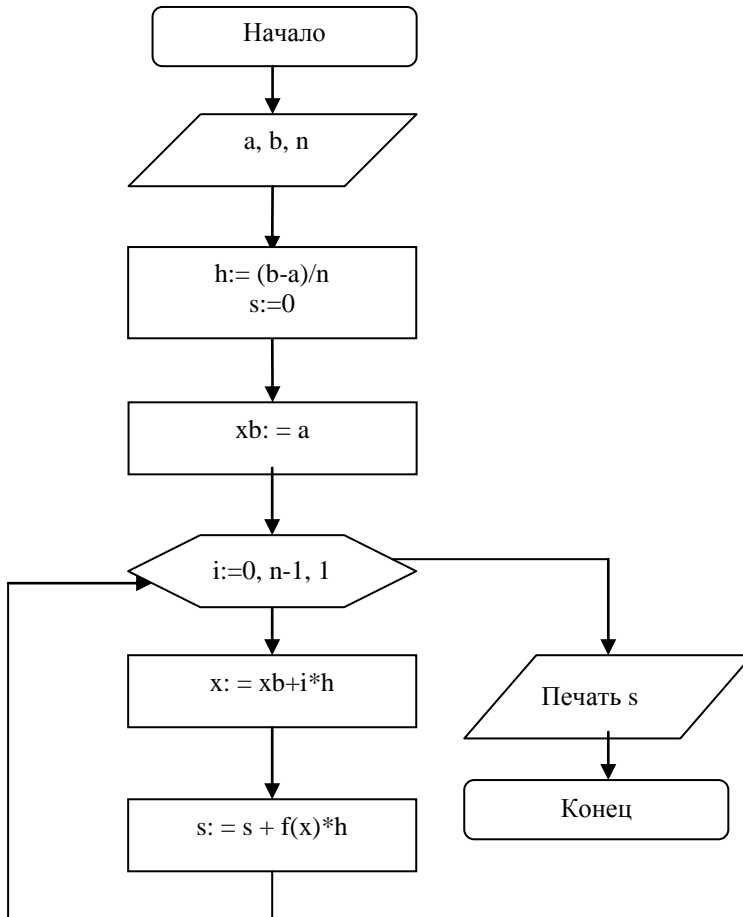
Все значения функции $f(x_i)$, кроме $f(x_0) = f(a)$ и $f(x_n) = f(b)$, встречаются в этой формуле по два раза. Поэтому, объединяя равные слагаемые, мы можем записать формулу трапеций в виде

$$I \approx I_T = \frac{h}{2} \left(f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right) = \frac{h}{2} (f(a) + f(b)) + h \sum_{i=1}^{n-1} f(x_i),$$

где $x_i = a + ih$, $i = 1, \dots, n-1$.

Пример 1. Найти значение определенного интеграла $I = \int_0^2 \frac{1}{x} \sin(x) dx$ методом левых прямоугольников.

Решение: Схема алгоритма



Текст программы на Pascal

```

program integral_1;
var i,n: integer;
    a,b,h,x,xb,s: real;
function f(x:real):real;
begin
f:= (1/x)*sin(x);
end;
begin
write('Введите верхний предел
интегрирования'); readln(a);
write('Введите нижний предел
интегрирования'); readln(b);
write('Введите количество
отрезков'); readln(n);
h:=(b-a)/n; s:=0; xb:=a;
for i:=0 to n-1 do
begin
x:= xb+i*h; s:= s+f(x)*h;
end;
writeln('Интеграл равен',
s:7:2);
end.
  
```

Пример 2. Найти значение определенного интеграла $I = \int_0^2 \frac{1}{x} \sin(x) dx$ методом трапеций.

Решение:

```

program integral_2;
var i,n: integer; a,b,h,x,s: real;
function f(x:real):real;
begin
f:= (1/x)*sin(x);
end;
begin
write('Введите верхний предел интегрирования'); readln(b);
write('Введите нижний предел интегрирования'); readln(a);
  
```

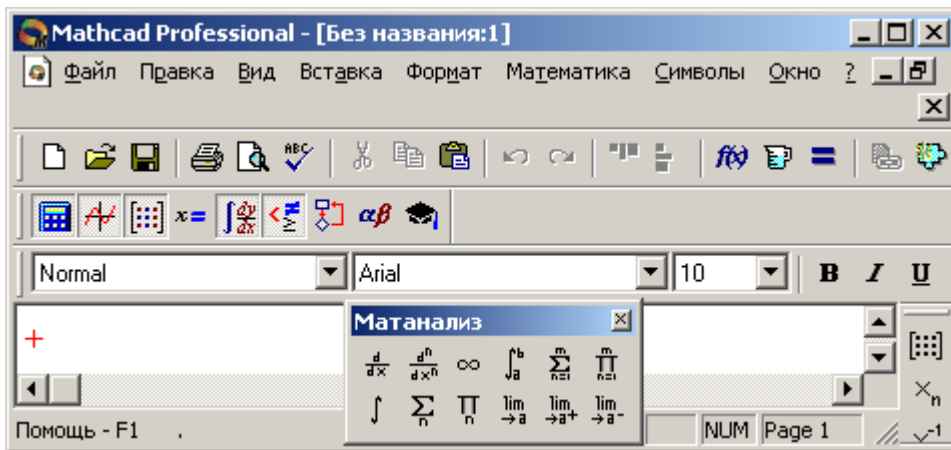
```

write('Введите количество отрезков'); readln(n);
h:=(b-a)/n; s:=0; x:=a;
for i:=1 to n-1 do begin
  x:= x+h; s:= s+f(x); end;
s:=h*((f(a)+f(b))/2 + s);
writeln('Интеграл равен', s:7:2);
end.

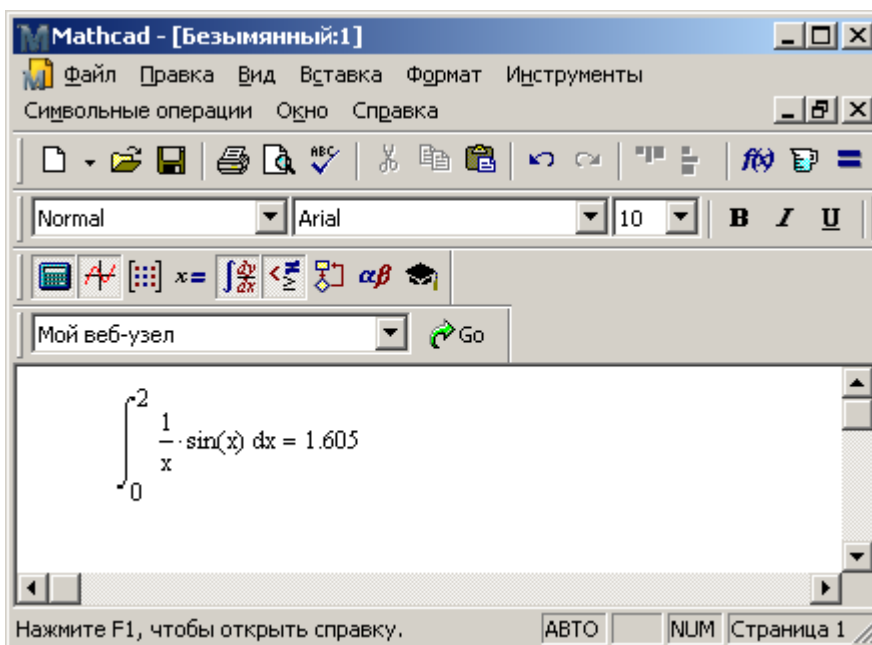
```

Пример 3. Найти значение определенного интеграла $I = \int_0^2 \frac{1}{x} \sin(x) dx$ средствами математического пакета MathCAD.

Решение: Для вычисления определенного интеграла необходимо выбрать знак интеграла из палитры инструментов.



После этого следует задать пределы интегрирования, подынтегральную функцию и переменную интегрирования, ввести знак = и нажать Enter.



Литература

1. Информатика : учебник для вузов / Н. В. Макарова, В. Б. Волков. - Санкт-Петербург [и др.] : Питер, 2012. - 573 с. : ил. - (Учебник для вузов) (Стандарт третьего поколения). Количество - 44.
2. Информатика. Базовый курс : учебник для вузов / под ред. С. В. Симоновича. - 3-е изд. - Санкт-Петербург [и др.] : Питер, 2012. - 637 с. : ил. - (Учебник для вузов) (Стандарт третьего поколения). Количество - 50.
3. Могилев, А. В. Информатика : учеб. пособие для пед. вузов / А. В. Могилев, Н. И. Пак, Е. К. Хеннер; под ред. Е. К. Хеннера. - 4-е изд., стер. - Москва : Академия, 2007. – 840 с. : ил. Количество - 97.
4. Основы программирования в среде Free Pascal [Электронный ресурс] : метод. указания для студентов и курсантов техн. специальностей / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматике и вычисл. техники ; сост. Н. И. Долюк, О. В. Нефедова. - Электрон. текстовые дан. (1 файл : 440 Кб). - Мурманск : Изд-во МГТУ, 2015. http://elib.mstu.edu.ru/2015/M_15_37.pdf.
5. Мурманский государственный технический университет. Информатика [Электронный ресурс] : опор. конспект лекций для студентов 1 курса техн. специальностей. Ч. 1 / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматике и вычисл. техники ; сост. Н. И. Долюк, О. В. Майорова. - Электрон. текстовые дан. (1 файл : 665 Кб). - Мурманск : Изд-во МГТУ, 2012. - Доступ из локальной сети Мурман. гос. техн. ун-та. http://elib.mstu.edu.ru/2012/U_12_11.pdf.
6. Информатика. ч. 2 [Электронный ресурс] : опор. конспект лекций для студентов 1 курса техн. специальностей / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматике и вычисл. техники ; сост. Н. И. Долюк, О. В. Нефедова. - Электрон. текстовые дан. (1 файл : 614 Кб). - Мурманск : Изд-во МГТУ, 2011. - Доступ из локальной сети Мурман. гос. техн. ун-та.
7. Компьютерная графика в системе КОМПАС-3D LT: методические указания к проведению лабораторных и практических работ/ Министерство науки и высшего образования РФ, Мурман. гос. техн. ун-т, Каф. Автоматике и вычисл. техники; сост. Лейко Н.Н., Майорова О.В. – Мурманск: Изд-во МГТУ, 2019 (файл находится на рабочих компьютерах в лабораториях)