

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

ИНФОРМАТИКА

*Методические указания
к практическим работам для студентов 1 курса специальности
08.03.01 «Строительство»*

Мурманск

2021

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное автономное
образовательное учреждение высшего образования
«Мурманский государственный технический
университет»

Кафедра автоматики и
вычислительной техники

Практические работы
по дисциплине Б1.О.10.02 Информатика

Методические указания для бакалавров по
направлению подготовки 08.03.01
«Строительство», профиль
«Промышленное и гражданское
строительство»

Мурманск
2021

Составитель:

Бучкова Зоя Алексеевна,
Старший преподаватель кафедры автоматики
и вычислительной техники ФГАОУ ВО
«Мурманский государственный технический
университет»

Методические указания рассмотрены и
одобрены кафедрой автоматики и
вычислительной техники 23 июня 2021 г.,
протокол № 7

*Электронное издание подготовлено в
авторской редакции*

Мурманский государственный технический университет
183010, Мурманск, ул. Спортивная д. 13 тел. (8152) 25-40-72

© Мурманский государственный
технический университет, 2021

ОГЛАВЛЕНИЕ

1. Практическая работа №1	5
2. Практическая работа №2	15
3. Практическая работа №3	19
4. Практическая работа №4	35
5. Практическая работа №5	43
6. Практическая работа №6	49
7. Практическая работа №7	56
8. Практическая работа №8	62
7. Практическая работа №9	68
9. Практическая работа №10	83
10. Практическая работа №11	104
11. Практическая работа №12	125
12. Практическая работа №13	136
13. Практическая работа №14	145
14. Практическая работа №15	150
15. Литература	150

Введение

Методические указания предназначены для проведения практических (лабораторных) работ по дисциплине «Информатика».

Цель практических занятий - предоставление возможностей для углубленного изучения теории, овладения практическими навыками и выработки самостоятельного творческого мышления у обучающихся.

Практические работы позволяют решать следующие задачи:

- отражение в учебном процессе современных достижений науки;
- углубление теоретической и практической подготовки студентов;
- формирование умения применять полученные знания на практике, осуществлять вычисления и расчеты;
- развитие инициативы и самостоятельности студентов;
- способности представлять результаты проведенного исследования, умения вести дискуссию;
- формирование общекультурных и общепрофессиональных компетенций;
- контроль за освоением учебной дисциплины.

В конце каждой темы приведены задания для самостоятельных работ и контрольные вопросы для закрепления изученного материала. Это позволяет использовать данные методические указания не только для аудиторной, но и для самостоятельной работы студентов.

Каждый студент должен отчитаться за практическую часть работы, выполнив индивидуальное задание по указанию преподавателя. По результатам выполнения индивидуального задания выставляется отметка (баллы) в журнале.

Отчёт о выполнении работы. Работа считается полностью выполненной после защиты отчёта и при наличии отметки о выполнении практической части. Отчёт о практической работе – индивидуальный творческий документ о её выполнении, который должен содержать:

- Краткое изложение основных теоретических положений по изучаемой теме;
- Чётко сформулированный вариант индивидуального задания;
- Практические результаты в виде таблиц, графиков, расчётов и т.д. с описанием способов их получения.

Приведённая в отчёте информация должна быть релевантной (полезной) и полной, т.е. достаточной, чтобы дать ответы на все поставленные вопросы и продемонстрировать, насколько глубоко пользователь разобрался в изучаемом вопросе.

Для защиты отчёта необходимо правильно и полно ответить на 2-3 вопроса преподавателя (из числа контрольных) либо решить задачу.

Практическая работа №1. Позиционные системы счисления. Двоичная арифметика. Основы машинной арифметики.

Цель: изучить способы перевода чисел из одной позиционной системы счисления в другую.

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.

Все системы счисления можно разделить на позиционные и непозиционные.

Непозиционная система счисления – система, в которой символы, обозначающие то или иное количество, не меняют своего значения в зависимости от местоположения в изображении числа.

Запись числа A в непозиционной системе счисления D может быть представлена выражением: $A_D = D_1 + D_2 + \dots + D_N = \sum_{i=1}^N D_i$, где A_D – запись числа A в системе счисления D ; D_i – символы системы [доп. 1].

К *непозиционной системе счисления* относится, например, римская, символы алфавита которой и обозначаемое ими количество представлены в таблице.

Римские цифры	I	V	X	L	C	D	M
Значение (обозначаемое количество)	1	5	10	50	100	500	1000

Запись чисел в этой системе счисления осуществляется по следующим правилам:

- 1) если цифра слева меньше, чем цифра справа, то левая цифра вычитается из правой (IV: $1 < 5$, следовательно, $5 - 1 = 4$, XL: $10 < 50$, следовательно, $50 - 10 = 40$);
- 2) если цифра справа меньше или равна цифре слева, то эти цифры складываются (VI: $5 + 1 = 6$, VIII: $5 + 1 + 1 + 1 = 8$, XX: $10 + 10 = 20$).

Пример 1. Число 1964 в римской системе счисления имеет вид MCMLXIV (М – 1000, CM – 900, LX – 60, IV – 4).

Систему счисления, в которой значение цифры определяется её местоположением в изображении числа, называют *позиционной*.

Возможно бесчисленное множество позиционных систем счисления: двоичная, троичная, четверичная и т.д. Запись чисел в каждой из систем счисления с основанием p означает сокращённую запись выражения [доп. 1]:

$$A(p) = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + \dots + a_{-m} p^{-m} = \sum_{k=-m}^n a_k p^k,$$

где a_i - цифры системы счисления;

n и m – число целых и дробных разрядов, соответственно;

A_p - запись числа A в p -ичной системе счисления.

Алфавиты некоторых систем счисления:

Основание	Система счисления	Алфавит системы счисления
2	Двоичная	0,1
3	Троичная	0,1,2
4	Четверичная	0,1,2,3
5	Пятеричная	0,1,2,3,4
8	Восьмеричная	0,1,2,3,4,5,6,7
10	Десятичная	0,1,2,3,4,5,6,7,8,9
12	Двенадцатеричная	0,1,2,3,4,5,6,7,8,9,A,B
16	Шестнадцатеричная	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Сложение, вычитание, умножение в двоичной системе счисления:

Сложение	Вычитание	Умножение
0+0=0	0-0=0	0*0=0
0+1=1	1-0=1	0*1=0
1+0=1	1-1=0	1*0=0
1+1=10	10-1=1	1*1=1

Сложение в восьмеричной системе счисления:

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11

3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Сложение в шестнадцатеричной системе счисления:

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Умножение в восьмеричной системе счисления:

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Умножение в шестнадцатеричной системе счисления:

×	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78

9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Пример 1. Десятичное число 35 в системах счисления с основанием P=12, 10, 8, 4, 3, 2 будет иметь вид:

$$2 \text{ B}_{12} = 2 \cdot 12^1 + \text{B} \cdot 12^0$$

$$3 \text{ 5}_{10} = 3 \cdot 10^1 + 5 \cdot 10^0$$

$$4 \text{ 3}_8 = 4 \cdot 8^1 + 3 \cdot 8^0$$

$$2 \text{ 0 3}_4 = 2 \cdot 4^2 + 0 \cdot 4^1 + 3 \cdot 4^0$$

$$1 \text{ 0 2 2}_3 = 1 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 2 \cdot 3^0$$

$$1 \text{ 0 0 0 1 1}_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Пример 2. Перевести данное число $A_{10} = 47$ из десятичной системы счисления в двоичную:

$$47 : 2 = 23(1);$$

$$23 : 2 = 11(1);$$

$$11 : 2 = 5(1);$$

$$5 : 2 = 2(1);$$

$$2 : 2 = 1(0);$$

$$1 : 2 = 0(1).$$

$$A_2 = 101111.$$

Пример 3. Перевести данное число в десятичную систему счисления.

$$\text{a) } 1000001_{(2)}.$$

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

Замечание. Очевидно, что если в каком-либо разряде стоит нуль, то соответствующее слагаемое можно опускать.

$$\text{б) } 1000011111,0101_{(2)}.$$

$$1000011111,0101_{(2)} = 1 \times 2^9 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-2} + 1 \times 2^{-4} = 512 + 16 + 8 + 4 + 2 + 1 + 0,25 + 0,0625 = 543,3125_{(10)}.$$

в) $1216,04_{(8)}$.

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

г) $29A,5_{(16)}$.

$$29A,5_{(16)} = 2 \times 16^2 + 9 \times 16^1 + 10 \times 16^0 + 5 \times 16^{-1} = 512 + 144 + 10 + 0,3125 = 656,3125_{(10)}.$$

Пример 4. Перевести число $0,2_{10}$ в двоичную систему счисления:

$$0,2 \cdot 2 = 0,4 = 0 + 0,4 \Rightarrow b_{-1} = 0;$$

$$0,4 \cdot 2 = 0,8 = 0 + 0,8 \Rightarrow b_{-2} = 0;$$

$$0,8 \cdot 2 = 1,6 = 1 + 0,6 \Rightarrow b_{-3} = 1;$$

$$0,6 \cdot 2 = 1,2 = 1 + 0,2 \Rightarrow b_{-4} = 1 \text{ и т. д.}$$

$A_2 = 0$. (0011) – периодическая дробь.

Пример 5. Перевести число $0,36_{10}$ в восьмеричную систему счисления:

$$0,36 \cdot 8 = 2,88 = 2 + 0,88 \Rightarrow b_{-1} = 2;$$

$$0,88 \cdot 8 = 7,08 = 7 + 0,08 \Rightarrow b_{-2} = 7;$$

$$0,08 \cdot 8 = 0,64 = 0 + 0,64 \Rightarrow b_{-3} = 0 \text{ и т. д.}$$

$A_8 = 0,270$.

Пример 6. Перевести число $67532,107_8$ в двоичную систему счисления.

Заменяем каждую цифру трёхзначным двоичным числом:

6 7 5 3 2 1 0 7

| | | | | | | |

110 111 101 011 010, 001 000 111

т.е. $67532,107_8 = 110111101011010, 001000111$

Пример 7. Перевести число $10111011101,1101$ в восьмеричную систему счисления.

Заменяем каждую триаду восьмеричным числом:

010 111 011 101, 110 100

| | | | | |

2 7 3 5 6 4

т.е. $10111011101,1101 = 2735,64$.

Пример 8. Перевести число $35B,451E_{16}$ в двоичную систему счисления.

Заменяем каждую шестнадцатеричную цифру двоичной тетрадой:

3	5	B,	4	5	1	F

0011 0101 1011, 0100 0101 0001 1111

т.е. $35B,451E_{16} = 1101011011,0100010100011111$.

Пример 9. Сложить числа:

а) $10001101,1_{(2)} + 111011,11_{(2)} = 11001001,01_{(2)}$.

б) $17_{(8)} + 6_{(8)} = 25_{(8)}$.

в) $3B3,6_{(16)} + 38B,4_{(16)} = 73E,A_{(16)}$.

10001101,1	17	3B3,6
+ 111011,11	+ 6	+ 38B,4
-----	----	-----
11001001,01	25	73E,A

Пример 10. Выполните вычитание:

а) $11001001,01_{(2)} - 111011,11_{(2)} = 10001101,10_{(2)}$.

б) $311,2_{(8)} - 73,6_{(8)} = 215,4_{(8)}$.

в) $C9,4_{(16)} - 3B,6_{(16)} = 8D,8_{(16)}$.

11001001,01	311,2	C9,4
- 00111011,11	- 73,6	- 3B,6
-----	-----	-----
10001101,10	215,4	8D,8

Пример 11. Выполнить умножение:

а) $100111_{(2)} \times 1000111_{(2)} = 101011010001_{(2)}$.

б) $1170,64_{(8)} \times 46,3_{(8)} = 57334,134_{(8)}$.

в) $61,A_{(16)} \times 40,D_{(16)} = 18B7,52_{(16)}$.

100111	1170,64	61,A
*1000111	* 46,3	*40,D
-----	-----	-----
100111	355 234	4F 52
+ 100111	+ 7324 70	+ 1868
100111	47432 0	-----
100111	-----	18B7,52
-----	57334,134	
101011010001		

Пример 12. Выполнить деление:

а) $1011011101001_{(2)} : 1110011_{(2)} = 110011$.

```

101101110100 |1110011
-1110011      110011
  1000100
-1110011
  10101100
- 1110011
   1110011
- 1110011
     0

```

Основы машинной арифметики

Цель: изучить представления чисел в прямом, обратном и дополнительном кодах и арифметические операции над ними.

Методические указания

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

Целые числа без знака. Обычно занимают в памяти компьютера один или два байта. В однобайтовом формате принимают значения от 00000000_2 до

11111111₂. В двухбайтовом формате от 00000000 00000000₂ до 11111111 11111111₂.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$0 \dots 2^8 - 1$	0...255
2	$0 \dots 2^{16} - 1$	0...65535

Число 72₁₀ в однобайтовом формате: 01001000₂

Число 72₁₀ в двухбайтовом формате: 0000000001001000₂.

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

Диапазон значений целых чисел со знаком

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$-2^7 \dots 2^7 - 1$	- 128...127
2	$-2^{15} \dots 2^{15} - 1$	-32768...32767
4	$-2^{31} \dots 2^{31} - 1$	-2147483648...2147483647

В цифровых автоматах 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:

$$X_{10} = -13, X_2 = -1101, [X_2]_{\text{пр}} = 1|1101,$$

$$X_{10} = 9, X_2 = 1001, [X_2]_{\text{обр}} = 0|1001.$$

Вертикальная линия отделяет знаковый разряд от абсолютного значения 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:

$$X_{10} = 12, X_2 = 1100, [X_2]_{\text{пр}} = [X_2]_{\text{обр}} = 0|1100,$$

$$X_{10} = -7, X_2 = -0111, [X_2]_{\text{обр}} = 1|1000.$$

Дополнительный код положительных чисел такой же, как и прямой код. У отрицательных чисел дополнительный код равен результату суммирования обратного кода числа с единицей младшего разряда: $X_{10} = 18$, $X_2 = 10010$, $[X_2]_{\text{пр}} = [X_2]_{\text{обр}} = [X_2]_{\text{доп}} = 0|10011$, $X_{10} = -11$, $X_2 = -1011$, $[X_2]_{\text{доп}} = [X_2]_{\text{обр}} + 1 = 1|0100 + 1 = 1|0101$.

Использование чисел со знаком (прямого кода представления чисел) усложняет структуру ЭВМ. В этом случае операция сложения двух чисел, имеющих разные знаки, должна быть заменена на операцию вычитания меньшей величины из большей и присвоения результату знака большей величины. Поэтому 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 1. Сложение обратных кодов.

1) X и Y положительные.

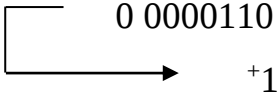
3	0 0000011
+ 7	+ 0 0000111
10	0 0001010

2) X положительное, Y отрицательное и по абсолютной величине больше, чем X.

+3	0 0000011	
-10	+ 1 1110101	обратный код числа -10
-7	1 1111000	обратный код числа -7

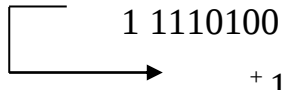
Получен корректный результат в обратном коде. При переводе в прямой код биты цифровой части результата инвертируются: $1\ 0000111 = -7_{10}$.

3) X положительное, Y отрицательное и по абсолютной величине меньше, чем X.

+10	+0 0001010
<u>-3</u>	<u>1 1111100</u> обратный код числа -3
7	0 0000110
	
	0 0000111

Компьютер исправляет полученный первоначально некорректный результат (6 вместо 7) *переносом единицы* из знакового разряда в младший разряд суммы.

4) X и Y отрицательные.

-3	+1 1111100	обратный код числа -3
<u>+ -7</u>	<u>1 1111000</u>	обратный код числа -7
-10	1 1110100	
		
	1 1110101	обратный код числа -10

При переводе результата в прямой код биты цифровой части числа инвертируются:

$$1\ 0001010 = -10_{10}.$$

При сложении может возникнуть ситуация, когда старшие разряды результата не помещаются в отведённой для него области памяти. Эта ситуация называется переполнением разрядной сетки формата числа.

5) X и Y положительные, сумма $X+Y$ больше либо равна 2^{n-1} , где n – количество разрядов формата чисел (для однобайтового формата $n = 8$, $2^{n-1} = 2^7 = 128$).

+ 65	+ 0 1000001	
<u>97</u>	<u>0 1100001</u>	
162	0 0100010	Переполнение

Семи разрядов цифровой части числового формата недостаточно для размещения 8 – разрядной суммы ($162_{10} = 10100010_2$), поэтому старший разряд суммы оказывается в знаковом разряде. Это вызывает несовпадение

знака суммы и знаков слагаемых, что является случаем переполнения разрядной сетки.

6) X и Y отрицательные, сумма абсолютных величин X и Y больше либо равна 2^{n-1} .

+ -63	+11000000	обратный код числа -63
<u>-95</u>	<u>10100000</u>	обратный код числа -95
-158	01100000	переполнение

Задачи и упражнения:

Выполните сложение чисел в обратном и дополнительном кодах. Результат переведите в прямой код:

Число X	Число Y
X = - 100101	Y = 11101
X = - 110101	Y = 11101
X = -1000111	Y = 11101
X = -1010001	Y = 10011
X = -1101001	Y = 10111
X = -101001	Y = 10111
X = - 110101	Y = 10011

Практическая работа №2. Логические основы работы ПК.

Цель: научиться составлять таблицы истинности и преобразовывать формулы алгебры логики.

Методические указания

Алгебра логики – раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Высказывание – некоторое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Высказывания, которым приписаны численные значения 1 или 0, называются *логическими переменными*.

В алгебре логики определены следующие логические операции:

$\neg, \bar{}$ - логическое отрицание (“НЕ”, инверсия);

$*, \&, \wedge, \text{AND}$ – логическое умножение (“И”, конъюнкция);

$+, \vee, \text{OR}$ – логическое сложение (“ИЛИ”, дизъюнкция);

$\oplus, \nabla$ - логическое исключающее “ИЛИ” (исключающая дизъюнкция);

$\rightarrow$ - логическое следование (импликация);

$\leftrightarrow, =$ - эквивалентность (двойная импликация);

$|$ - функция Шеффера;

$\downarrow$ - стрелка Пирса, или функция Вебба °;

$\overrightarrow{} \text{ и } \overleftarrow{}$ - обратная импликация, или коимпликация правая и левая;

$\sim$ - эквивалентность или функция тождества;

Значение каждой логической функции описывается общей таблицей истинности. *Таблица истинности* представляет собой таблицу, устанавливающую соответствие между возможными значениями наборов переменных и значениями функции.

Теоремы алгебры логики:

1. $\overline{\overline{A}} \equiv A$ - закон двойного отрицания.
2. $A \& B \equiv B \& A$ - коммутативный закон для конъюнкции.
3. $A \vee B \equiv B \vee A$ - коммутативный закон для дизъюнкции.
4. $(A \& B) \& C \equiv A \& (B \& C)$ - ассоциативный закон для конъюнкции.
5. $(A \vee B) \vee C \equiv A \vee (B \vee C)$ - ассоциативный закон для дизъюнкции.
6. $A \& (B \vee C) \equiv (A \& B) \vee (A \& C)$ - дистрибутивный закон.
7. $A \vee (B \& C) \equiv (A \vee B) \& (A \vee C)$ - дистрибутивный закон.
8. $A \& A \equiv A$ - закон идемпотентности для конъюнкции.
9. $A \vee A \equiv A$ - закон идемпотентности для дизъюнкции.
10. $(A \& B) \vee A \equiv A$ - закон поглощения.
11. $(A \vee B) \& A \equiv A$ - закон поглощения.

12. $\overline{A \& B} \equiv \overline{A} \vee \overline{B}$ - закон де Моргана.

13. $\overline{A \vee B} \equiv \overline{A} \& \overline{B}$ - закон де Моргана.

14. $A \& 1 \equiv A$ - закон единицы для конъюнкции.

15. $A \& 0 \equiv 0$ - закон нуля для конъюнкции.

16. $A \vee 1 \equiv 1$ - закон единицы для дизъюнкции.

17. $A \vee 0 \equiv A$ - закон нуля для дизъюнкции.

Пример 1. Построить таблицу истинности для функции [доп. 13]

$$F(x_1, x_2, x_3) = (x_1 \rightarrow x_2) \rightarrow \overline{x_3}$$

Вначале выписываются значения, которые могут принимать набор переменных в этой функции. В общем случае, если переменных n , то различных n -мерных наборов переменных существует 2^n . Затем вычисляется значение функции на каждом наборе. Любая, рассматриваемая логическая функция представляет собой суперпозицию элементарных логических функций и может быть вычислена последовательно при помощи подстановок определённых ранее значений.

Итак, определим значение наборов переменных. Их восемь:

$x_1 \ x_2 \ x_3$	$x_1 \rightarrow x_2$	$\overline{x_3}$	$(x_1 \rightarrow x_2) \rightarrow \overline{x_3}$
0 0 0	1	1	1
0 0 1	1	0	0
0 1 0	1	1	1
0 1 1	1	0	0
1 0 0	0	1	1
1 0 1	0	0	1
1 1 0	1	1	1
1 1 1	1	0	0

$$A \cdot B + \overline{A} \cdot B = B,$$

Пример 2. Доказать тождество $(A + B) \cdot (\overline{A} + B) = B$

Доказательство первого тождества проводится с использованием первого закона дистрибутивности:

$$A \cdot B + \bar{A} \cdot B = B \cdot (A + \bar{A}) = B \cdot 1 = B$$

Доказательство второго тождества проводится с использованием второго закона дистрибутивности: $(A + B) \cdot (\bar{A} + B) = A \cdot \bar{A} + B = B$.

Пример 3. Доказать закон свёртки логического выражения [доп. 4]:

$$A \cdot B + \bar{A} \cdot C + B \cdot C = A \cdot B + \bar{A} \cdot C.$$

Данное тождество можно доказать, последовательно используя законы работы с логическими константами, дистрибутивности, идемпотентности и склеивания:

$$\begin{aligned} A \cdot B + \bar{A} \cdot C + B \cdot C &= A \cdot B \cdot (C + \bar{C}) + \bar{A} \cdot C \cdot (B + \bar{B}) + B \cdot C \cdot (A + \bar{A}) = \\ &= A \cdot B \cdot C + A \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C + A \cdot B \cdot C + \bar{A} \cdot B \cdot C = \\ &= (A \cdot B \cdot C + A \cdot B \cdot \bar{C}) + (\bar{A} \cdot B \cdot C + \bar{A} \cdot \bar{B} \cdot C) + (\bar{A} \cdot \bar{B} + \bar{A} \cdot B \cdot C) = \\ &= A \cdot B + \bar{A} \cdot C + \bar{A} \cdot C = A \cdot B + \bar{A} \cdot C \end{aligned}$$

Пример 4. Используя основные равносильности алгебры логики, а также равносильности

$$A \rightarrow B \equiv \bar{A} \vee B \quad \text{и} \quad A \leftrightarrow B \equiv (A \rightarrow B) \& (B \rightarrow A) \equiv (A \& B) \vee (\bar{A} \& \bar{B}),$$

упростить формулу $(X \rightarrow Y) \rightarrow Y$.

Решение.

$$\begin{aligned} (X \rightarrow Y) \rightarrow Y &\equiv (\bar{X} \vee Y) \rightarrow Y \equiv \overline{\bar{X} \vee Y} \vee Y \equiv (\bar{\bar{X}} \& \bar{Y}) \vee Y \equiv (X \& \bar{Y}) \vee Y \equiv \\ &\equiv (X \vee Y) \& (\bar{Y} \vee Y) \equiv (X \vee Y) \& 1 \equiv X \vee Y \end{aligned}$$

Задачи и упражнения:

1. Упростить выражение $S = \overline{(A \cdot B) \cdot (A + B)}$.

2. Известен следующий закон свёртки логического выражения:

$(A \cdot B) | (\bar{A} \cdot C) | (B \cdot C) = (A \cdot B) | (\bar{A} \cdot C)$, т.е. третье слагаемое логической суммы в левой части выражения не влияет на конечный результат. Проверить этот закон с помощью таблицы истинности.

3. Докажите второй закон де Моргана с помощью таблиц истинности:

$$\overline{x \wedge y} = \bar{x} \vee \bar{y}.$$

4. Даны два высказывания:

A1={если одно слагаемое делится на 3 и сумма делится на 3, то и другое слагаемое делится на 3}

A2={если одно слагаемое делится на 3, а другое не делится на 3, то сумма не делится на 3}.

Формализуйте эти высказывания, постройте таблицы истинности и убедитесь, что результирующие столбцы совпадают.

5. Для заданного логического выражения:

а) Построить таблицу истинности;

б) Упростить высказывания, используя законы алгебры логики;

с) Полученный результат проверить, построив для него таблицу истинности.

1	$(A \leftrightarrow B) \vee A\bar{B} \vee C$
2	$(AC \rightarrow B) \vee A\bar{C}$
3	$(A\bar{B} \vee (A \leftrightarrow C)B)$
4	$(\bar{A} \leftrightarrow B)(A \rightarrow BC)$
5	$(B \rightarrow C) \vee (B \rightarrow AC)$
6	$(AB \rightarrow C) \vee A\bar{C}$
7	$(AC \rightarrow \bar{B}) \vee B\bar{C}$
8	$(A \rightarrow B)(C\bar{A} \rightarrow \bar{B})$

Практическая работа №3. Прикладное программное обеспечение.

Текстовый процессор Microsoft Word.

Компьютерные технологии обработки текстовой информации в MS Word

Цель: изучение основных операций процессора *Microsoft Word*.

Наработка практических навыков по созданию и оформлению текстовых документов.

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. Основными параметрами редактора *Microsoft Word* по работе с документами являются:

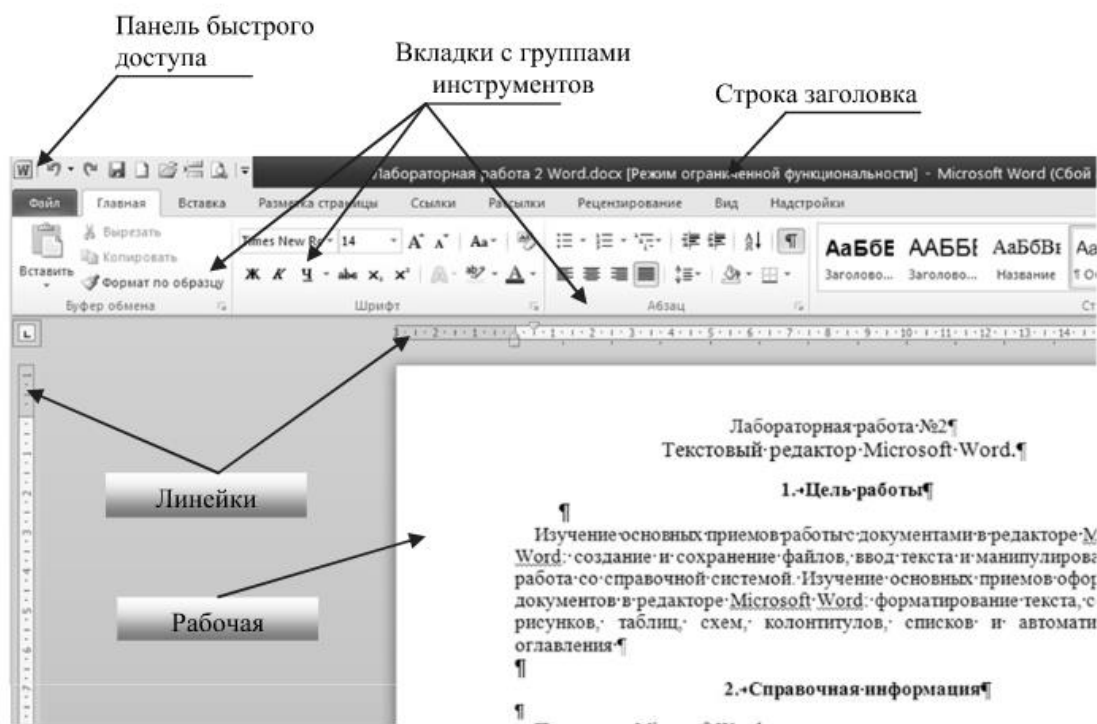
- набор (ввод) текста;
- редактирование содержимого документа (внесение каких-либо изменений в текст документа: изменение взаимного расположения отдельных частей документа, поиск и замена символов, слов и участков текста);
- форматирование содержимого документа (оформление документа таким образом, чтобы он стал более наглядным, удобочитаемым, эстетичным и соответствовал правилам оформления документов данного типа).

Кроме того, *Microsoft Word* обладает такими возможностями, как:

- 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.

Абзац – фрагмент текста от одного нажатия клавиши Enter до следующего. В ячейках таблицы абзацем является фрагмент от начала ячейки до ближайшего нажатия клавиши Enter или знака конца ячейки¹.

¹ Асташова, Т.А. Информатика : учебное пособие : [16+] / Т.А. Асташова ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2017. – 108 с. : ил., табл. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=574622>



Структура окна Microsoft Word

При установке параметров для одного абзаца выделять его необязательно. Достаточно, если в этом абзаце будет находиться курсор. Если же оформляется сразу несколько абзацев, их необходимо выделить.

Раздел – часть документа, в которой применяется определённый формат страницы. Например, для изменения количества колонок необходимо вставить в документ новый раздел с помощью команды вкладки Вставка/Разрыв.

Списки – это фрагменты текста, пункты которого отмечены специальными знаками. Списки могут быть маркированными, нумерованными, многоуровневыми.

Колонтитулы – это небольшие идентификаторы, которые указываются вверху и внизу во всём документе и содержат важные сведения о нём. В них содержатся такие сведения, как номера страниц, даты, название книги или главы, а также имя автора.

Стили – это наборы различных вариантов форматирования, которые отображаются в виде эскизов в коллекции экспресс-стилей. Коллекция экспресс-стилей используется для быстрого применения стилей к абзацам или фрагментам текста документа.

Форматирование – оформление текста, связанное с изменением его внешнего вида. Различают три основные операции форматирования:

- форматирование абзацев – изменение полей абзацев, изменение интервалов между строками и выравнивание абзацев;

- форматирование символов - изменение шрифта отдельных букв, слов, предложений, абзацев;
- форматирование страниц – выбор размера, ориентации и полей страниц.

Использование формул в таблицах

Команда **Формула** находится в разделе Работа с таблицами на вкладке Макет в группе Данные.

Основные функции для работы с данными

Основные функции	Описание
=AVERAGE()	Вычисление среднего арифметического
=COUNT()	Подсчет количества
=MAX(), =MIN()	Нахождение максимума, минимума
=SUM()	Подсчет суммы
<i>Ключевое слово</i>	<i>Местонахождение данных</i>
BELOW, ABOVE	Под ячейкой и над ячейкой
LEFT, RIGHT	Слева и справа от ячейки

Сноски предназначены для добавления к тексту комментариев, объяснений, указания источника информации. Сноски бывают обычными (в конце страницы) и концевыми (в конце всего текста).

Оглавление – это список заголовков документа. Для того, чтобы быстро сделать оглавление, документ должен быть отформатирован согласно встроенным форматам уровней структуры или стилей заголовков.

Гиперссылка – это выделенный цветом или подчеркнутый текст или графический элемент, на который можно нажать для открытия одного из следующих элементов:

- файла;
- местоположение в файле;
- веб-страницы в Интернете;
- веб – страницы в интрасети;
- узла Gopher, Telnet или FTP.

Основные операции с текстом:

1. Ввод и исправление текста.
2. Работа со шрифтами.
3. Установка параметров страницы. Проверка документа. Форматирование символов и абзацев. Стили документов. Форматирование и сортировка списков.
4. Шаблоны документов. Колонки. Таблицы.
5. Панели инструментов и меню. Импорт и создание графики. Размещение текста и графики.

Задание №1. С помощью редактора формул наберите следующие выражения:

[illegible]

$$2. \quad \lim_{x \rightarrow 0+0} \ln y = \lim_{x \rightarrow 0+0} \frac{\ln x}{\frac{1}{x}} \left(\frac{\infty}{\infty} \right) = \lim_{x \rightarrow 0+0} \frac{1}{x \left(-(x)^{-2} \frac{1}{\cos^2 x} \right)}$$

$$3. f(x) = \begin{cases} \frac{x+4}{x^2-16} & \text{при } x < 0 \\ \frac{\sin x}{x^2-9} & \text{при } x > 0 \\ \frac{\sin(x-2)}{x^2-4} & \text{при } x = 0 \end{cases}$$

1. Создайте надписи и заголовки.
2. Все элементы рисунка должны быть сгруппированы.
3. При создании рисунка используйте копирование, свободное вращение (для поворота нижней половины гиперболоида после копирования) и рисование дуг.

Задание №3. Создайте таблицу в соответствии с образцом:

Сведения об успеваемости студентов ИМА МГТУ																	
№ п/п	Учебная дисциплина	Группа	Средний балл	Всего сдавало	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно	Неявки	Учебная дисциплина	Средний балл	Всего сдавало	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно	Неявки
1	Информатика	371	3,88	32	12	10	6	3	1	Информатика	4,4	32	12	10	6	3	1
2		372	3,52	27	7	9	6	3	2		3,4	20	6		5	2	0
3		373	3,43	28	9	8	3	5	3		3,9	23	9	8	3	5	5
4		374	3,52	29	8	8	8	3	2		3,5	29	8	4	6		4
	Итого			116	36	35	23	14	8	Итого	116	36	35	20	13	10	

Задание №4. Создайте нижний колонтитул на всех страницах, кроме первой, в который вставьте из автотекста колонтитула номера страниц, дату создания документа и рисунок.

Задание №5. Создайте бланк документа.

Документ (документированная информация)- зафиксированная на материальном носителе информация с реквизитами, позволяющими её идентифицировать².

Бланк - это стандартный лист бумаги с воспроизведённой на нём постоянной информацией документа и местом, отведённым для переменной информации. Требования к бланкам документов, включая бланки документов с воспроизведением Государственного герба РФ, изложены в IV разделе ГОСТ 6.30-2003. Образцы бланков документов приведены в приложении Б к Государственному стандарту.

² Журавлёва И.В., Журавлёва М.В. Оформляем документы на персональном компьютере: грамотно и красиво. – М.: ИНФРА-М, 2013. – 187 с. – (Просто, кратко, быстро).

Формат бланка и размер полей документа

Государственным стандартом Р 6.30-2003 установлены два формата бланков документов- А4 (210×297 мм) и А5 (148×210 мм). Стандартом установлен минимальный размер полей документа: верхнее, нижнее и левое поле-20 мм, правое -10 мм.

Однако, следует отметить, что:

- в верхнем поле документа печатается номер страницы, и при размере верхнего поля 20 мм он может оказаться слишком близко к тексту;
- смещение листа при проходе через принтер при печати может привести к тому, что правое поле в нижней части листа уменьшится до 3-5 мм вместо установленных минимальных 10 мм;
- размер левого поля (20 мм) часто недостаточен для подшивки документов в дело, т.к. если папка с документами имеет толщину 3...4 см, на последних листах текст, расположенный слева, будет трудно прочитать.

В связи с этим рекомендуется устанавливать следующие размеры полей: верхнее и нижнее поля- по 20-25 мм; левое поле- 30 - 35 мм, правое-15 мм.

Документы оформляются на белой бумаге или бумаге светлых тонов.

В зависимости от расположения реквизитов Государственным стандартом установлено два варианта бланков – угловой и продольный.

Схемы расположения реквизитов и границы зон на угловом и продольном бланках приведены в Приложениях 2 и 3. В п. 4.3 Государственного стандарта Р 6.30-2003 содержится очень важная для нас информация: «...**Ориентировочные границы** зон расположения реквизитов обозначены пунктиром. Каждая зона определяется совокупностью входящих в неё реквизитов». Из этой формулировки можно сделать вывод: положение пунктирных границ зон расположения реквизитов можно изменять – переносить их выше или ниже, левее или правее.

Угловой бланк отличается тем, что реквизиты с 01 по 14 (в зависимости от вида документа, для которого проектируется бланк) располагаются в левом верхнем углу листа. На продольных бланках эти реквизиты расположены вдоль верхнего поля документа.

Создаём бланки на стандартном листе формата А4.

Ориентация страницы – **книжная**.

Работаем в режиме **Разметка страницы** (меню Вид/Разметка страницы).

Включим непечатаемые знаки (меню Главная/Отобразить все знаки).

Установим размер полей (меню Макет/Поля):

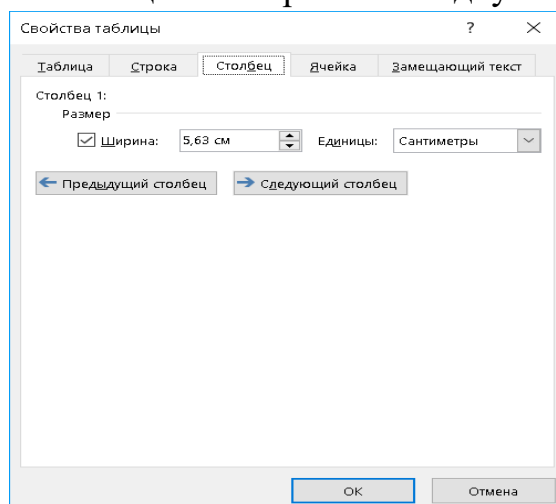
- левое поле - 3,0 см;
- правое поле – 1,5 см;
- верхнее поле – 2,0-2,5 см;
- нижнее поле – 2,0-2,5 см.

Проектирование продольного бланка с центрированным расположением реквизитов

Вставьте таблицу из трёх столбцов и семи строк (меню **Вставка/Таблица**):

Установите ширину столбцов таблицы. Для этого:

1. Выделите первый столбец таблицы, в меню **Таблица** выберите команду **Свойства таблицы...** Откройте вкладку столбец:

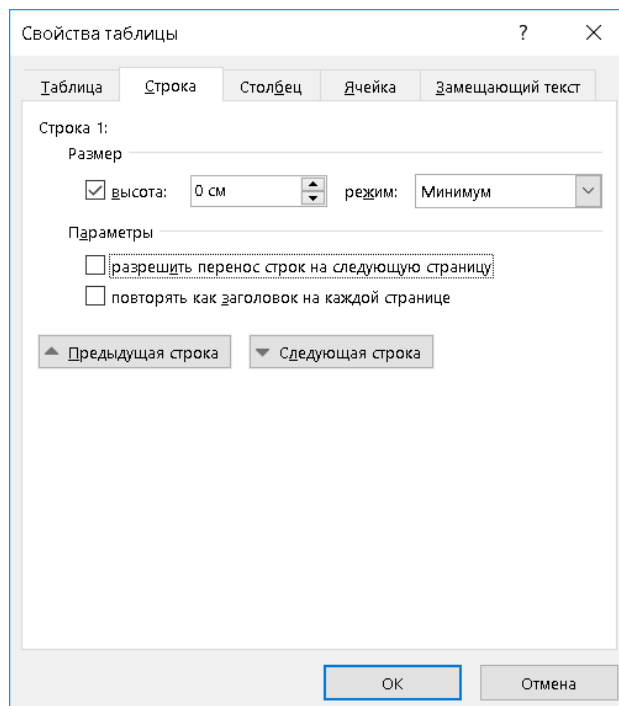


2. Исправьте цифру в поле **Ширина** на **7,3** см и нажмите кнопку **Следующий столбец**. Для второго столбца установите ширину **1,4** см, для третьего столбца – **7,8** см.

Выделите первую строку таблицы и объедините три ячейки этой строки (меню **Таблица** /команда **Объединить ячейки**). Затем поочерёдно объедините по три ячейки второй, четвёртой и седьмой строки).

Установите необходимую высоту строк таблицы. Для этого:

1. Выделите первую строку таблицы. Откройте диалоговое окно **Свойства таблицы** и перейдите на вкладку **Строка**:



Обратите внимание: в диалоговом окне нужно установить флажок в поле **Высота**, выбрать режим **Минимум** и снять флажок в полях **Разрешить перенос на следующую страницу** и **Повторять как заголовок на каждой странице**.

- Используя кнопку **Следующая строка** для перехода к работе с каждой следующей строкой, установите требуемую высоту строк:

- строка 1 - минимум 1,7 см;
- строка 2 - минимум 2,0 см;
- строка 3 - минимум 1,0 см;
- строка 4 - минимум 1,0 см;
- строка 5 - минимум 1,2 см;
- строка 6 - минимум 1,8 см;
- строка 7 - минимум 1,3 см.

В результате получится такая таблица:

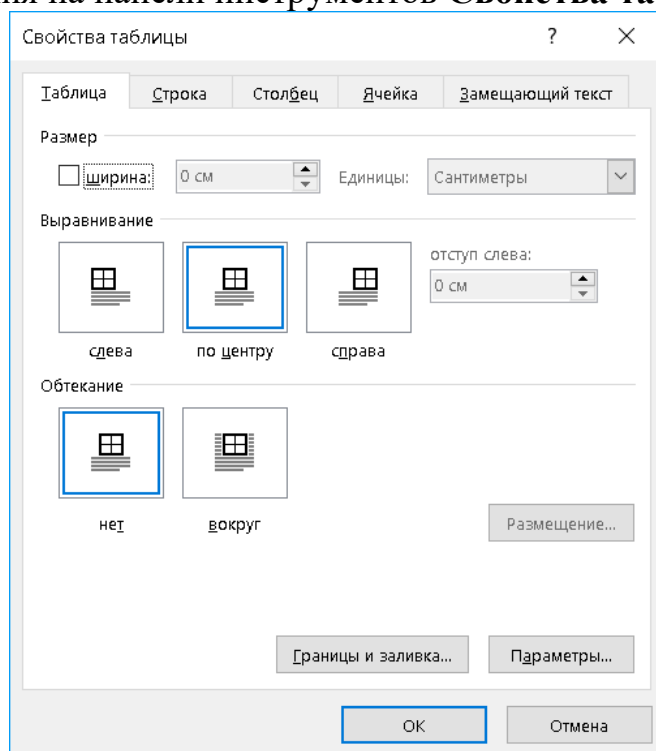
1		
2		
3	4	5
6		
7	8	9
10	11	12
13		

Ячейки таблицы пронумерованы для удобства дальнейшего описания работы по созданию бланка.

Последняя строка таблицы (ячейка 13) предназначена для создания интервала между заголовком к тексту и текстом документа – текст будет печататься непосредственно под таблицей.

Поочерёдно выделите и объедините ячейки 8 и 11, затем 9 и 12.

В объединённой ячейке 9-12 будет расположен либо реквизит «Адресат», либо «Гриф утверждения». В этой ячейке можно установить выравнивание **По центру** или **Слева**. В ячейках 1 и 2 установите выравнивание **По центру**. Для этого можно воспользоваться полем выбора способа выравнивания на панели инструментов **Свойства таблицы**:



Используя этот же инструмент, в ячейке **6** установите выравнивание **По центру**, а в ячейке **10** – **Слева**.

Установите размер и начертание шрифта: в ячейке **2** (для наименования организации)- 14-18 пт, полужирное начертание, в остальных ячейках таблицы – 12-14 пт.

До размещения в ячейках таблицы постоянных реквизитов бланка нужно сделать невидимыми границы таблицы.

Для этого выделите всю таблицу. В поле выбора границ на панели инструментов **Форматирование** или **Границы и заливка** щёлкните по кнопке **Нет границы**. Если контуры таблицы не видны, работать с такой таблицей неудобно. В этом случае нужно выбрать команду **Отображать сетку**. Сетка таблицы не печатается, она видна только на экране компьютера:

	1	
	2	
3	4	5
	6	
7	8	9
10	11	12
13		

В зависимости от назначения проектируемого бланка в ячейках таблицы могут быть размещены реквизиты, указанные в таблице.

Номер ячейки	Общий бланк	Бланк письма	Бланк приказа	Бланк справки с места работы
	Номера и наименование реквизитов в соответствии с ГОСТ Р 6.30-2003			
1	01 – государственный герб РФ, или 02 – Герб субъекта РФ, или 03 – Эмблема организации ³			
2	05-ОГРН; 08- наименование организации ⁴	04-Код по ОКПО; 05-ОГРН; 06- наименование организации; 09- справочные данные об организации.	08– наименование организации; 10-наименование вида документа.	04-Код ОКПО; 05-ОГРН; 06-(ИНН/КПП); 08- наименование организации; 09- справочные данные об организации; 10-наименование вида документа;
3	11-дата документа	11-дата документа 12- регистрационный номер документа 13-ссылка на номер и дату инициативного документа	11- дата документа	11- дата документа
4	Знак «№»	Знак «№»	Знак «№»	Знак «№»
5	12- регистрационный номер документа		12- регистрационный номер документа	12- регистрационный номер документа
6	14- место составления или издания документа		14- место составления или издания документа	
10	18- заголовок к тексту			
9-12	15- адресат	15- адресат		15- адресат

³ Эмблема организации может размещаться в области верхнего колонтитула, в таком случае первая строка таблицы не нужна, её следует удалить.

⁴ Если проектируется бланк структурного подразделения, в этом поле ниже наименования организации должно быть указано наименование структурного подразделения. Если проектируется бланк должностного лица, после наименования организации будет указано наименование должности руководящего работника, для которого проектируется бланк.

(объединённая)				
----------------	--	--	--	--

Для общего бланка и бланка конкретного вида документа необходимо установить ширину ячеек **3** и **5** равной приблизительно 2 см, ориентируясь по горизонтальной линейке⁵.

Установите выравнивание в ячейках **3** и **5** **снизу по левому краю**, в ячейке **4** – **справа по нижнему краю**. Поместите курсор в ячейку **3** и добавьте нижнюю границу.

Повторите эту операцию в ячейке **5**. В ячейке **4** напечатайте знак «№».

На бланке письма в ячейке **3** необходимо разместить реквизиты **11** (дата документа) и **12** (регистрационный номер документа), а также реквизит **13** (ссылка на регистрационный номер и дату документа). Для размещения этих реквизитов ячейку **3** надо предварительно подготовить:

1. Поместите курсор в ячейку **3**. В меню **Таблица** выберите команду **Разбить ячейки...** На экране появится окно **Разбиение ячеек**. В поле **Число столбцов**: укажите **1**, а в поле **Число строк**: **-2**. Щёлкните по кнопке **ОК**. В ячейке **3** образуется две строки.
2. Поместите курсор в верхнюю строку ячейки **3**, откройте диалоговое окно **Разбиение ячеек**. В поле **Число столбцов**: укажите **3**, а в поле **Число строк**: оставьте **1**. Щёлкните по кнопке **ОК**.
3. Поместите курсор в нижнюю строку ячейки **3**, откройте диалоговое окно **Разбиение ячеек**. В поле **Число столбцов**: укажите **4**, а в поле **Число строк**: оставьте **1**. Щёлкните по кнопке **ОК**.

Теперь ячейка **3** выглядит так:

				1	
				2	
A	B	C		4№	5
D	E	F	G		
				6	
7				8	9
10				11	12
13					

Для удобства дальнейшего описания работы в получившихся после разбиения ячейках поставлены латинские буквы.

4. Перетаскивая правые границы ячеек и ориентируясь по горизонтальной линейке, установите требуемые ширины ячеек:

ячейка **A** 2,6 см;
ячейка **B** 0,6 см;
ячейка **D** 1,12 см;

⁵ Если на экране нет горизонтальной линейки, в меню Вид нужно выбрать команду **Линейка**.

ячейка F 0,6 см;

Для того, чтобы установить точную ширину ячейки, при перемещении её правой границы надо удерживать нажатой клавишу **ALT**, при этом горизонтальная линейка изменится так, что с её помощью можно будет с большой точностью установить ширину столбцов.

Ширину остальных ячеек изменять не надо.

Если в дальнейшем при работе с бланком потребуется увеличить или уменьшить ширину каких-то ячеек, это легко можно сделать, перемещая правые границы ячеек.

5. Поместите курсор в ячейку **A** и добавьте нижнюю границу. Повторите операцию для ячеек **C**, **E**, **G**.

6. В ячейке **B** напечатайте знак «№». В ячейку **D** введите «**На №**», в ячейке **F** напечатайте предлог «**от**».

7. В ячейках **B** и **F** установите выравнивание **Снизу по центру**.

В результате ячейка **З**, подготовленная для бланка письма, будет выглядеть так:

		1	
		2	
	№	4№	5
На №	от		
		6	
7		8	9
10		11	12
13			

Для ввода в ячейки таблицы постоянных реквизитов можно дать следующие рекомендации:

1. тип шрифта, в соответствии с рекомендациями Типовой инструкции по делопроизводству в органах Федеральной исполнительной власти, выбираем Times New Roman;
2. для наименования организации можно использовать шрифт размером от 14 до 22 пт, можно добавить полужирное начертание;
3. для справочных данных об организации выбираем шрифт размером от 9 до 12 пт, в зависимости от объёма вводимых данных;
4. при подготовке бланка конкретного вида документа наименование документа печатается прописными (заглавными) буквами, можно добавить полужирное начертание (например, ПРИКАЗ или СПРАВКА);
5. для того чтобы наименование вида документа зрительно отделялось от названия организации, нужно сделать следующее: щёлкнуть мышкой в строке с наименованием вида документа, в меню **Главная** выбрать

команду **Абзац**, перейти на вкладку Отступы и интервалы и **увеличить Интервал перед и после** на 6-12 пт;

6. для даты и регистрационного номера документа также можно использовать полужирное начертание: они будут хорошо зрительно выделяться.

Самое широкое распространение продольный бланк с центрированным расположением реквизитов получил при изготовлении общих бланков организаций.

Заметим, что если на бланке для писем и справок должно быть выделено достаточно места для оформления реквизита «Адресат», то на бланке для приказов надобность в таком поле отпадает, поэтому из формы бланка можно удалить строку, в которой находится ячейка 7.

Пример 1. Заголовочная часть справки с места работы:



Общество с ограниченной ответственностью «ПИЛИГРИМ»
Проспект Андропова, д. 34/2, Москва, Россия, 125315 Тел. (495) 123-45-67. Факс (495) 123-45-68 Р/сч. 12345678901234567890 в ОАО в «Банк Москвы», к/сч. 12345678912345678912; БИК 123456789. ОКПО 12345678; ОГРН 1234567890123; ИНН/КПП 1234567890/321654987

СПРАВКА

02.02.2019

№ 15

Единый информационно-
расчётный центр ЮАО г. Москвы

Задание №6. Слияние документов Word и Excel.

Слияние документов Word и Excel пригодится при заполнении отчётов, писем, рассылок, уведомлений, договоров, и т.д.

Задание. Написать уведомление клиентам, список которых представлен в следующей таблице.

Таблица Excel должна соответствовать следующим правилам:

1. Таблица Excel должна иметь шапку таблицы (название столбцов).
2. Не должно быть одинаковых названий столбцов.
3. В таблице не должно быть объединённых ячеек, чтобы в документе Word не было смещения строк.

Фамилия	Имя	Отчество	Год рождения	Пол	Срок предоплаты	Срок оплаты
Иванова	Светлана	Михайловна	1990	ая	15.02.2018	15.04.2018
Григорьев	Иван	Владимирович	1986	ый	12.01.2018	15.04.2018
Иванов	Иван	Иванович	1978	ый	15.04.2018	27.07.2018

--	--	--	--	--	--	--

Бланк Word

Уважаем ! Администрация центра «Радуга» напоминает Вам, что срок окончательной оплаты вашей учёбы истекает Просим оплатить учёбу до числа. С уважением, директор центра «Радуга» Иванов И.О.

Это заготовка бланка – одинаковый текст для всех клиентов, без имён, без окончаний слова «Уважаемый».

Теперь выполним слияние документов **Word** и **Excel**. Для этого воспользуемся мастером слияния **Word**. В **Word**, где заготовка нашего бланка на закладке **Рассылки** в разделе **Начать слияние** нажимаем кнопку **Начать слияние**. Из списка выбираем функцию **Пошаговый мастер слияния**.

Справа экрана появится окно мастера слияния. Остаётся выполнить всё, что он вам скажет.

1. Выберем **Тип документа** – **Письма**. Нажимаем внизу этого диалогового окна кнопку **Далее**.
2. В диалоговом окне **Выбор документа** ставим – **Текущий документ**. Т.к. мы открыли мастер слияния на странице Word, на которой написали шаблон бланка и хотим заполнить данными этот бланк – текущий документ. Нажимаем кнопку **Далее**. **Выбор получателей**.
3. У нас уже есть список получателей этого послания в таблице Excel, поэтому выбираем **Использование списка**. А чтобы найти нашу таблицу, нажимаем кнопку **Обзор**. Вот наш список. Теперь в этом списке нужно установить галочки напротив тех людей, которым будет рассылать это письмо с напоминанием оплаты. Если список большой,

то можно воспользоваться фильтром. Поставим в таблице галочки у нужных фамилий, всё проверили, нажимаем **Ок**. Нажимаем кнопку **Далее. Создание письма** в диалоговом окне мастера слияний.

4. Теперь будем расставлять в нашем шаблоне бланка нужные слова в нужных местах документа. Ставим курсор перед восклицательным знаком – здесь мы напишем фамилию. В диалоговом окне **Создание письма** нажимаем кнопку **Другие элементы**. В появившемся диалоговом окне из списка выбираем слово *Фамилия*. Нажимаем кнопку **Вставить**. Нажимаем кнопку **Ок**. Делаем один пробел. Снова нажимаем на кнопку **Другие элементы**, выбираем слово *Имя*, нажимаем **Вставить**, **Ок**. После слова «*Уважаем*» ставим слово *Пол*. Так помечаем все места нужными словами. У нас получилась такая заготовка документа:

Уважаем «Пол» «Фамилия» «Имя» «Отчество»!
Администрация центра «Радуга» напоминает Вам, что срок окончательной оплаты вашей учёбы истекает «Срок_оплаты»
Просим оплатить учёбу до «Срок_оплаты» числа.
С уважением, директор центра «Радуга» Сидоров И.О.

Нажимаем кнопку в окне мастера слияния **Далее. Просмотр писем**.

5. Просматриваем как получились письма. Нажимаем на стрелку в диалоговом окне у слова «Получатель: 1», перелистываем письма. Получилось так :

Уважаемый Григорьев Иван Юрьевич!
Администрация центра «Радуга» напоминает Вам, что срок окончательной оплаты вашей учёбы истекает 15.04.20148 числа.
Просим оплатить учёбу до 4/15/20148 числа.
С уважением, директор центра «Радуга» Сидоров И.О.

А дата написана не так как мы привыкли читать. Нажимаем правой кнопкой мыши на дату в нашем письме и выбираем из контекстного меню функцию **Коды/Значения полей**. Вместо даты образовалось такое поле:

Просим оплатить учёбу до {MERGEFIELD “Срок оплаты”} числа.

Код менять не будем, но допишем формат даты внутри фигурных скобок:

Просим оплатить учёбу до {MERGEFIELD “Срок оплаты”\@”DD.MM.YYYY”}
числа.

Нажимаем снова правой кнопкой на этот код и выбираем функцию **Обновить поле**. Получилось так:

Просим оплатить учёбу до 15.04.2018 числа.

Нажимаем кнопку в мастере слияния **Далее. Завершение слияния**.

6. Здесь можно выбрать разные функции. **Печать** – распечатать письма. Или **Изменить часть писем**. Можно сохранить письма. Всё, документ готов. Закрывать окно мастера слияния. На каждом этапе можно вернуться к предыдущему этапу, можно выбрать другой список. Например, в процессе работы, мы изменили список.

Уважаемый Григорьев Иван Юрьевич!

Администрация центра «Радуга» напоминает Вам, что срок окончательной оплаты вашей учёбы истекает 15.04.2014 числа.

Просим оплатить учёбу до 4/15/20148 числа.

С уважением, директор центра «Радуга» Сидоров И.О.

Контрольные вопросы:

1. Что может содержать сложный документ?
2. Объясните основные понятия документа: абзац, раздел, форматирование.
3. Для чего используют гиперссылки в текстовом документе?
4. Каковы возможности использования формул в таблицах текстового редактора?
5. В чём отличие понятий форматирования и редактирования?
6. Назначение колонтитулов в текстовом документе.

Практическая работа №4. Табличный процессор Microsoft Excel.

Использование электронных таблиц в решении математических задач.

Цель: изучение основных приёмов работы с табличным процессором MS

Excel

MS Excel – это компьютерная программа, предназначенная для обработки данных. Она позволяет не только создавать таблицы, но и автоматизировать обработку табличных данных.

К возможностям процессора относят:

- создание и редактирование электронных документов;
- получение выборки данных, удовлетворяющих определённым критериям;
- сортировку данных по определённому принципу;
- составление сводных отчётов с помощью нескольких способов определения итогов и объединения данных;
- построение графиков и диаграмм;
- обработку данных с использованием аппарата функций и формул;
- исследование различных факторов на данные.

Основные приёмы выполнения вычислений

Цель: научиться простым приёмам выполнения вычислений.

Методические указания

MS Excel – это компьютерная программа, предназначенная для обработки данных. Она позволяет не только создавать таблицы, но и автоматизировать обработку табличных данных⁶.

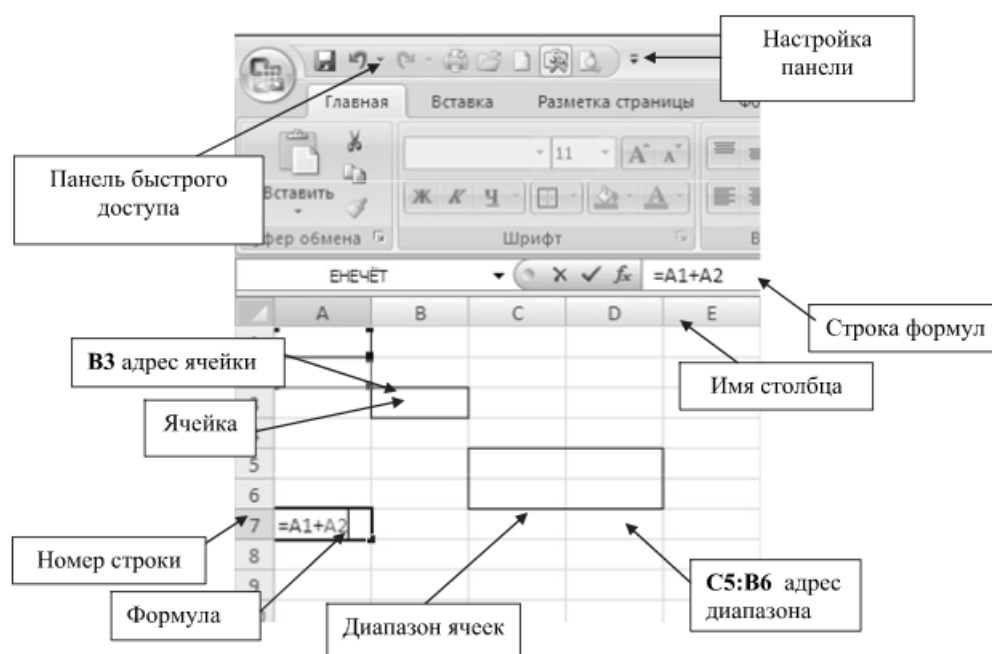
К возможностям процессора относят:

- создание и редактирование электронных документов;
- получение выборки данных, удовлетворяющих определённым критериям;
- сортировку данных по определённому принципу;

⁶ Асташова, Т.А. Информатика : учебное пособие : [16+] / Т.А. Асташова ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2017. – 108 с. : ил., табл. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=574622>

- составление сводных отчётов с помощью нескольких способов определения итогов и объединения данных;
- построение графиков и диаграмм;
- обработку данных с использованием аппарата функций и формул;
- исследование различных факторов на данные.

Основными элементами MS Excel являются: ячейка, диапазон ячеек, строка, столбец, адрес ячейки, формула, абсолютная и относительная адресация.



Основные элементы MS Excel

Вычисления в электронных таблицах осуществляются при помощи формул. Формула может содержать числовые константы, ссылки на ячейки и функции Excel, соединённые знаками математических операций.

Ячейки в MS Excel – это основной элемент таблицы, образованный пересечением столбца и строки.

Адрес ячейки – имя столбца и номер строки, на пересечении которых находится ячейка, представляют собой координаты, определяющие расположение этой ячейки на листе.

Диапазон ячеек – это группа рядом стоящих ячеек, образующая прямоугольник.

Относительные ссылки – это ссылки, которые при копировании формулы изменяются автоматически в соответствии с относительным расположением исходной ячейки и создаваемой копией (A5).

Абсолютные ссылки – это ссылки, которые при копировании не изменяются (\$F\$5).

Смешанные ссылки – это ссылки, которые сочетают в себе и относительную и абсолютную адресацию (A\$7, \$A7).

Формулой Excel считается всё, что начинается со знака «=». В формуле можно использовать различные типы операторов (арифметические и т.п.), текст, ссылки на ячейку или диапазон ячеек, круглые скобки, именованные диапазоны. В формулах соблюдается приоритет выполнения операций. Для изменения порядка выполнения операций используются круглые скобки.

Если в формуле используется текст, то он обязательно должен быть заключён в двойные кавычки.

Ошибки в формулах

- #ДЕЛ/0 Деление на нуль;
- #Н/Д Недопустимое значение;
- #ИМЯ? Не распознаётся имя;
- #ПУСТО! Формула использует пересечение непересекающихся областей;
- #ЧИСЛО! Проблема с числовым значением;
- #ССЫЛКА! Неверная ссылка;
- #ЗНАЧЕНИЕ! Неверный тип аргумента или операнда;

Переполнение ячейки

Пример 1. Вычислить функции: $Y = e^{3x+1}$, $Y = \ln(1 + 3x)$, $Y = \sqrt{5 \sin x}$.

3. Открыть Книгу1 и выполнить ввод аргумента $x = 0,8$ в ячейку A1.
4. Найти нужную функцию (EXP, LN, КОРЕНЬ).
5. В диалоговом окне Аргументы функции ввести аргумент, например, $3 \cdot x + 1$, где x -ссылка на ячейку A1.
6. Если в качестве аргумента используется другая функция, то в поле аргумента ввести эту функцию, а затем в следующее диалоговое окно – адрес ячейки, т.е. A1.

Задачи и упражнения [доп. 13]:

1. Составьте таблицу значений функции $\sin x$ для $-3,14 \leq x \leq 3,14$ в 20 точках.
2. Для заданной таблицы найдите максимальное, минимальное и среднее значения. Значения X – 10; 20 30 40. Значения Y – 100; 110; 120 130.

	X1	X2	X3	X4
Y1	Y1-X1	Y1-X2	Y1-X3	Y1-X4
Y2	Y2-X1	Y2-X2	Y2-X3	Y2-X4
Y3	Y3-X1	Y3-X2	Y3-X3	Y3-X4
Y4	Y4-X1	Y4-X2	Y4-X3	Y4-X4

3. В таблице представлены трёхкратные измерения каждого из 5 параметров.

0,05	0,04	0,98
1,96	1,72	1,75
2,05	2,70	3,98
3,91	3,06	4,23
4,55	4,18	1,67

Для каждого параметра вычислите:

- Среднее значение всех измерений $I_{\text{ср}}$.
- Отклонение каждого измерения от среднего значения $O_i = I_i - I_{\text{ср}}$.
- Среднее отклонение $O_{\text{ср}}$.
- Максимальное отклонение $O_{\text{макс}}$.
- Сумму произведений измерений на их отклонения от среднего значения $\sum I_i O_i$.

- Сумму квадратов отклонения от среднего значения $\sum O_i^2$.

Массивы

Цель: научиться выполнять операции с матрицами.

Пример 1. Пусть имеется неоднородная система линейных уравнений (т.е. свободные члены отличны от нуля):

$$x_1 + 2x_2 + 3x_3 = 4$$

$$2x_1 - 3x_2 - 4x_3 = 1$$

$$x_1 + 5x_2 - 5x_3 = 7.$$

Решим эту систему матричным методом, согласно которому исходную систему уравнений можно представить следующим образом:

$$A \cdot X = B,$$

где A – основная матрица, состоящая из коэффициентов при неизвестных;

B – матрица свободных членов,

X – матрица неизвестных.

Введём в ячейки Excel две матрицы $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & -3 & -4 \\ 1 & 5 & -5 \end{pmatrix}$ и $B = \begin{pmatrix} 4 \\ 1 \\ 7 \end{pmatrix}$. Искомое

решение будет находиться по формуле:

$$X = A^{-1} \cdot B.$$

Для его нахождения необходимо выполнить следующие действия:

1. Найти обратную основной матрицу A^{-1} при помощи функции *МОБР* из категории *Математические* (перед вводом функции необходимо выделить диапазон, содержащий достаточно ячеек для размещения всех результатов; после завершения ввода формулы нажмите клавиши *Ctrl+Shift+Enter*)

Обратная матрица A^{-1} равна:

0,406977 0,290698 0,011628

0,069767 -0,09302 0,116279

0,151163 -0,03488 -0,0814

2. С использованием функции *МУМНОЖ* выполнить перемножение матриц A^{-1} и B , в результате получим искомое решение:

2

1

1,11022E-16

1. Excel позволяет так же транспонировать матрицу (функция *ТРАНСП* из категории *Полный алфавитный перечень*) и вычислить детерминант матрицы (функция *МОПРЕД*).

Задачи и упражнения:

1. Вычислите определитель матрицы A . Найдите обратную матрицу A^{-1} .

Решите систему линейных уравнений $Ax=b$ [доп. 13].

$$1. A = \begin{pmatrix} -9 & 2 & 3 & -5 \\ -3 & 1 & 3 & 5 \\ 3 & 1 & 4 & 2 \\ 1 & 2 & 4 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad 2. A = \begin{pmatrix} -7 & 2 & 5 & -5 \\ -1 & 1 & 1 & 7 \\ 3 & 3 & 4 & 4 \\ -1 & 2 & 4 & 5 \end{pmatrix}, \quad b = \begin{pmatrix} -2 \\ 3 \\ 2 \\ 8 \end{pmatrix}$$

$$3. A = \begin{pmatrix} -10 & 2 & 2 & -5 \\ -4 & 1 & 4 & 4 \\ 3 & 0 & 4 & 1 \\ 2 & 2 & 4 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 11 \end{pmatrix} \quad 4. A = \begin{pmatrix} -10 & 2 & 2 & -1 \\ -4 & 1 & 4 & 4 \\ 3 & 0 & 4 & 1 \\ 2 & 2 & 4 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 11 \end{pmatrix}$$

$$5. A = \begin{pmatrix} -5 & 2 & 7 & -5 \\ 1 & 1 & -1 & 9 \\ 3 & 5 & 4 & 6 \\ -3 & 2 & 4 & 7 \end{pmatrix}, \quad b = \begin{pmatrix} -4 \\ 5 \\ 4 \\ 6 \end{pmatrix} \quad 6. A = \begin{pmatrix} -11 & 2 & 7 & -2 \\ -5 & 1 & 5 & 3 \\ 3 & -1 & 4 & 0 \\ 3 & 2 & 4 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ -1 \\ -2 \\ 12 \end{pmatrix}$$

$$7. A = \begin{pmatrix} -4 & 2 & 8 & 5 \\ 2 & 1 & -2 & 10 \\ 3 & 6 & 4 & 7 \\ -4 & 2 & 4 & 8 \end{pmatrix}, \quad b = \begin{pmatrix} 7 \\ 6 \\ 5 \\ 5 \end{pmatrix} \quad 8. A = \begin{pmatrix} -3 & 2 & 9 & 6 \\ 3 & 1 & -3 & 11 \\ 3 & 7 & 4 & 8 \\ -5 & 2 & 4 & 9 \end{pmatrix}, \quad b = \begin{pmatrix} 8 \\ 7 \\ 6 \\ 4 \end{pmatrix}$$

Нахождение корней уравнения методом Подбора параметра

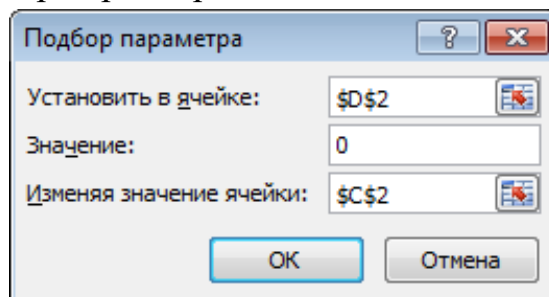
Цель: научиться применять метод подбора параметров для нахождения корней уравнений в MS Excel.

Задание

Найти корни полинома третьей степени $x^3 - 2x^2 + 3x - 5 = 0$ методом Подбора параметра.

Порядок выполнения работы:

1. Протабулируем полином на интервале $[0;3]$ с шагом 0,2;
2. В ячейку B2 введём формулу: $=A2^3-2*A2^2+3*A2-5$ и скопируем на весь диапазон;
3. Видно, что полином меняет знак на интервале $[1,8;2]$, поэтому там имеется корень;
4. В ячейку C2 запишем начальное приближение к корню уравнения 1,9;
5. В ячейку D2 запишем функцию, для которой ведётся поиск корня, причём вместо неизвестной у этой функции должна указываться ссылка на ячейку, отведённую под искомый корень, т.е. $=C2^3-2*C2^2+3*C2-5$;
6. Найдём точное значение корня. Для этого выберем: Данные/Анализ “что если”/Подбор параметра:



Результат

A	B	C	D	E
x	f(x)	корень	f(x)	
0	-5	1,843734	1,2E-07	
0,2	-4,472			
0,4	-4,056			
0,6	-3,704			
0,8	-3,368			
1	-3			
1,2	-2,552			
1,4	-1,976			
1,6	-1,224			
1,8	-0,248			
2	1			
2,2	2,568			

Варианты заданий для самостоятельной работы

Решить уравнения методом Подбора параметра [5]:

1. $x^3 - 3x^2 + 9x - 8 = 0$
2. $x^3 + 2x^2 - 5x + 1 = 0$
3. $x^3 - 3x + 1 = 0$
4. $x^4 - 18x^2 + 6 = 0$
5. $x^3 - 4x^2 - 3x = 0$
6. $3x^4 + 4x^3 - 12x^2 - 5 = 0$
7. $x^3 - 0,2x^2 + 0,3x - 3 = 0$
8. $3x^4 + 8x^3 + 6x^2 - 5 = 0$
9. $x^3 + 4x + 2 = 0$
10. $x^3 - 3x^2 + 9x - 8 = 0$

Практическая работа №5. Графические возможности Microsoft Excel.

Цель: научиться строить графики функций и поверхности.

Методические указания

Для построения графика выполните следующие действия:

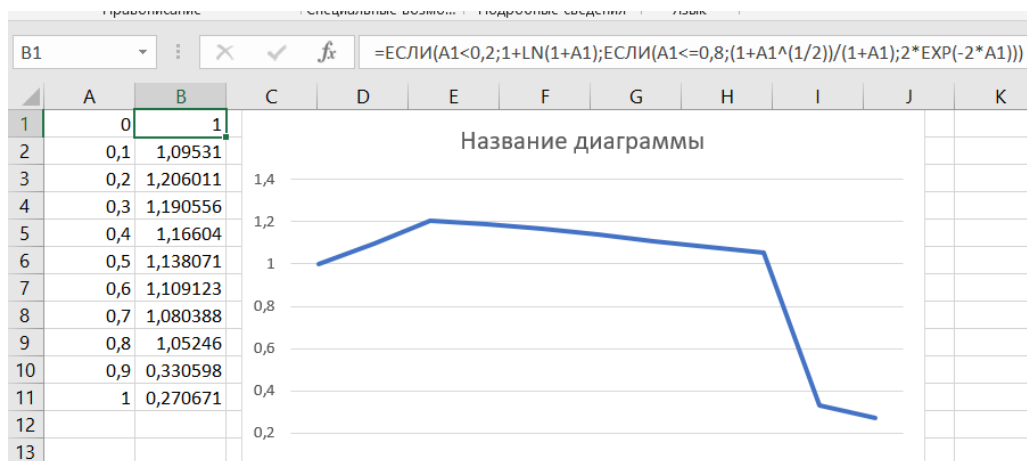
- 1) Ввести в один из столбцов таблицы значения аргумента на симметричном интервале;
- 2) В ячейки соседнего столбца ввести формулу, используя мастер функций;
- 3) Выделив значения функции в таблице, построить график функции с помощью Мастера диаграмм.

Пример 1. Построить график с тремя условиями на $[0;1]$:

$$y = \begin{cases} 1 + \ln(1+x), & \text{если } x < 0,2 \\ \frac{1+x^{1/2}}{1+x}, & \text{если } x \in [0,2; 0,8] \\ 2e^{-2x}, & \text{если } x > 0,8 \end{cases}$$

Введите в ячейку B1 следующее логическое выражение, используя функцию ЕСЛИ:

=ЕСЛИ (A1<0,2; 1+LN(1+A1); ЕСЛИ (A1<=0,8; (1+A1^(1/2))/(1+A1); 2*EXP(-2*A1)))

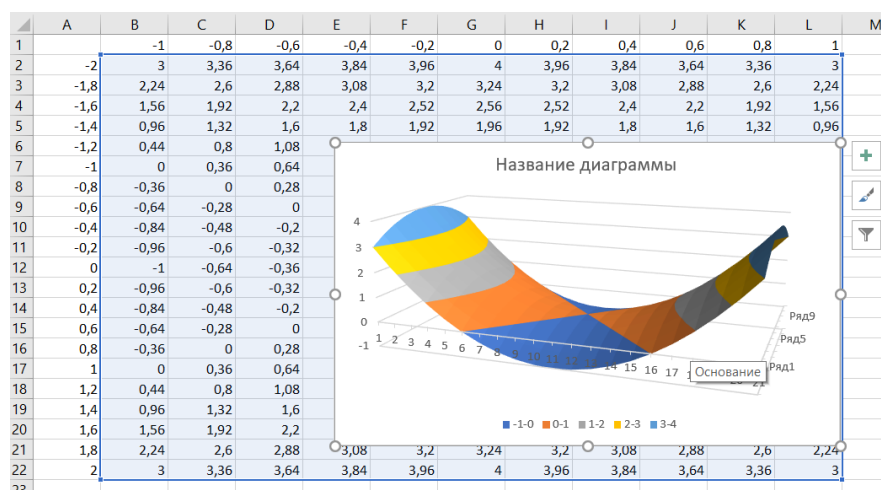


Пример 2. Построить поверхность $S = x^2 - y^2$

при $x \in [-2; 2]$, $y \in [-1; 1]$.

1. В ячейку A2 введите значение -2, в ячейку A3 – значение -1,8.
2. Выделите A2:A3 и с помощью маркера автозаполнения протяните на диапазон A4:A22. Т. о, значения аргумента x протабулированы от -2 до 2 с шагом 0,2.

- В ячейку B1 введите значение -1, в ячейку C1 введите значение -0,8. Аналогично протяните на диапазон D1:L1.
- В ячейку B2 введите формулу: $=\$A2^2 - B\1^2 .
- Скопируйте формулу на диапазон B2:L22.
- По полученному диапазону значений постройте поверхность.



Задания для самостоятельной работы:

Постройте графики функций:

а). $x \in [-2; 2] \quad y = \frac{1 + x^2}{1 + 2x^2}$

б). $y = \begin{cases} 3 \sin(x) - \cos^2(x), & x \leq 0 \\ 3\sqrt{1 + x^2}, & x > 0 \end{cases}$

в). $z = \begin{cases} \frac{1 + x}{\sqrt[3]{1 + x^2}}, & x \leq 0 \\ -x + 2e^{-2x}, & x \in (0; 1) \\ |2 - x|^{1/3}, & x \geq 1 \end{cases}$

г). Два графика в одной системе координат $x \in [-2; 2]$

$$y = 2 \sin(\pi x) - 3 \cos(\pi x)$$

$$y = \cos^2(2\pi x) - 2 \sin(\pi x)$$

д). Построить поверхность, $x, y \in [-1; 1]$ $z = 3x^2 - 2 \sin^2(y)y^2$, используя смешанные ссылки.

Контрольные вопросы:

1. Что представляет собой электронная таблица?
2. Как формируется адрес ячейки?
3. Что называется диапазоном ячеек? Как он задаётся?
4. Что такое трёхмерный диапазон?
5. Чем различаются относительная и абсолютная ссылки?
6. Какие типы данных встречаются в электронных таблицах?
7. Перечислите и поясните существующие форматы представления числовых данных в ячейках электронной таблицы.
8. Как защитить содержимое ячеек электронной таблицы от несанкционированного доступа и внесения изменений?
9. Что такое макросы и для чего они используются?
10. Что такое автозаполнение?
11. Что такое консолидация таблиц?
12. Как посмотреть и отредактировать формулу, содержащуюся в ячейке?
13. Как осуществляется фильтрация списков?
14. Какие вы знаете методы обработки и анализа данных в электронных таблицах?

Построение и редактирование диаграмм

Цель: изучить принципы построения и редактирования диаграмм.

Методические указания

1. В новой рабочей книге на рабочем листе **Лист 1** создать данные, подобные по форме показанным в таблице. Присвоить листу имя **Данные⁷**.

	Район 1	Район 2	Район 3	Район 5
Янв	12	13	17	200
Фев	13,5	12	12	150
Мар	15	14	15	170,0
Апр	16,5	15	11	225
Май	18	12	14	325
Июн	19,5	20	16	268

2. Выделите диапазон A1:D7.
3. Нажать клавишу F11. На новом р/л **Диаграмма1** появляется диаграмма, построенная с принятым по умолчанию типом и форматом.
4. Выполнить команду **Диаграмма/Параметры диаграммы**.
5. В диалоговом окне **Параметры диаграммы** выбрать карточку **Заголовки** и в поле **Название диаграммы** ввести **Учебная диаграмма**. **ОК**.
6. На панели инструментов **Диаграммы** щёлкнуть первую кнопку слева **Формат заголовка диаграммы** или в контекстном меню заголовка выполнить команду **Формат заголовка диаграммы**.
7. Провести редактирование заголовка.

Связывание названия диаграммы с ячейкой рабочего листа:

1. Ввести в любую ячейку р/л, на котором находятся данные, новое название диаграммы, например в ячейку A10 ввести **Новая диаграмма**, в ячейку A11- **Другая диаграмма**.
2. Выделить диапазон A1:D7.
3. Щёлкнуть кнопку **Мастер диаграмм**.
4. Выбрать тип диаграммы – **График**.
5. На третьем шаге **Мастера** ввести название диаграммы – **Учебная диаграмма**.
6. По окончании построения выделить название диаграммы.
7. В строке формул набрать знак «=».
8. Щёлкнуть ячейку с основным названием (A10 или A11).
9. **ОК**.

Добавление названий к осям диаграммы

1. Перейти на лист **Диаграмма1**.
2. Выполнить команду **Диаграмма/Параметры диаграммы**.
3. В диалоговом окне **Параметры диаграммы** выбрать карточку **Заголовки** и ввести названия осей.
4. **Готово**.

⁷ Соболев, Б. В. Практикум по информатике: / Б. В. Соболев [и др.]; под ред. Б.В. Соболева. – Ростов н / Д: Феникс, 2009. – 509 [1] с. – (Высшее образование).

Добавление меток к элементам данных

1. Перейти на лист **Диаграмма1**.
2. Щёлкнуть один раз любой маркер данных за февраль, затем щёлкнуть этот же маркер ещё раз.
3. Выполнить команду **Формат/Выделенный элемент данных**.
4. В диалоговом окне **Формат элемента данных** перейти на карточку **Подписи данных**.
5. Установить опцию **значения**.
6. Нажать **ОК**.
7. Повторить шаги 2-6 для любой метки данных за июнь.

Добавление плавающего текста

Ни один элемент диаграммы не должен быть выделен, поле имени должно быть пустым. Выделить область построения. Щёлкнуть строку формул и набрать нужный текст (если текст слишком длинный, то CTRL+Enter позволит переместить его в несколько строк). Завершить ввод набранного текста. Текстовое поле с введённым текстом можно перенести в любое место диаграммы.

Форматирование текста на диаграмме выполняется по команде **Формат/Выделенный объект** в соответствующих диалоговых окнах на карточках **Вид, Шрифт, Выравнивание, Защита**.

Перемещение, редактирование и форматирование легенды

Перемещение легенды

1. Выделить легенду.
2. Выполнить команду **Формат/Выделенная легенда**.
3. В диалоговом окне **Формат легенды** перейти на карточку **Размещение**.
4. С помощью выбора соответствующей радиокнопки легенду можно переместить в любое место, даже на диаграмму.

Изменение текста легенды

Форматирование осуществляется с помощью выделения нужного ряда данных в раскрывающемся списке **Элементы диаграммы** панели инструментов **Диаграммы** и выполнения команды **Формат/Выделенный ряд**. Для форматирования отдельного текста внутри легенды нужно выделить его так же, как и отдельный маркер на диаграмме, и выполнить команду **Формат/Выделенный элемент легенды** или воспользоваться контекстным меню.

Форматирование легенды

3. Выделить легенду.
4. Команда **Формат/Выделенная легенда/карточки Вид и Размещение**.

Относительное расположение маркеров данных

На гистограммах или линейчатых диаграммах маркерами данных являются линейки или столбцы, которые группируются в кластеры. Каждый кластер содержит маркеры разных рядов данных, относящихся к одному значению категории:

1. Выделить ряд данных.
2. Выполнить команду **Формат/Выделенный ряд**.
3. В диалоговом окне **Формат ряда данных** на карточках **Вид, Порядок рядов, Параметры** устанавливаются внешний вид маркеров данных, порядок рядов в кластерах, перекрытие маркеров и зазор между кластерами.
4. Форма и расположение маркеров на объёмных гистограммах задаются в диалоговом окне **Формат ряда данных** на карточке **Параметры** тремя величинами: **Глубиной зазора** (измеряется от 0% до 500 %), **Ширина зазора** (изменяется от 0% до 500 %), **Глубина диаграммы** (изменяется от 20 % до 2000%).

Линии проекции и минимакс линии

Линии проекции - это прямые, которые опускаются из точки данных на ось категорий. Их можно добавить к диаграмме-графику с помощью команды **Формат/Выделенный ряд/карточка Параметры/флажок Линии проекции**.

Минимакс линии- это отрезки прямых, соединяющие точки с наименьшим и наибольшим значениями в одном кластере. Их можно добавить с помощью команды **Формат/Выделенный ряд – карточка Параметры/флажок Минимум-максимум**.

Практическая работа № 6. Прикладное программное обеспечение.

Создание презентаций в MS Power Point.

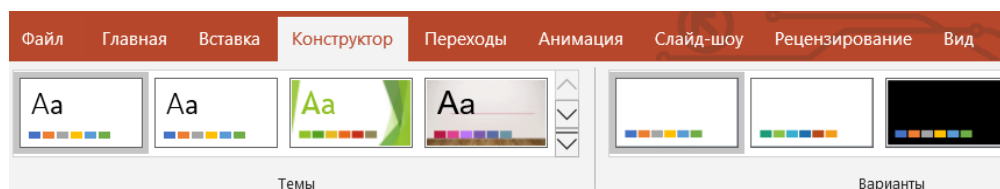
Цель: научиться создавать и редактировать презентацию.

Презентация – мультимедийный инструмент, используемый в ходе докладов или сообщений для повышения выразительности выступления, более убедительной и наглядной иллюстрации описываемых фактов и явлений. MS PowerPoint- программа для создания и проведения презентаций.

MS PowerPoint позволяет создавать наглядные презентации, интегрируя текст, графику, видео и другие элементы на отдельных страницах, называемых слайдами.

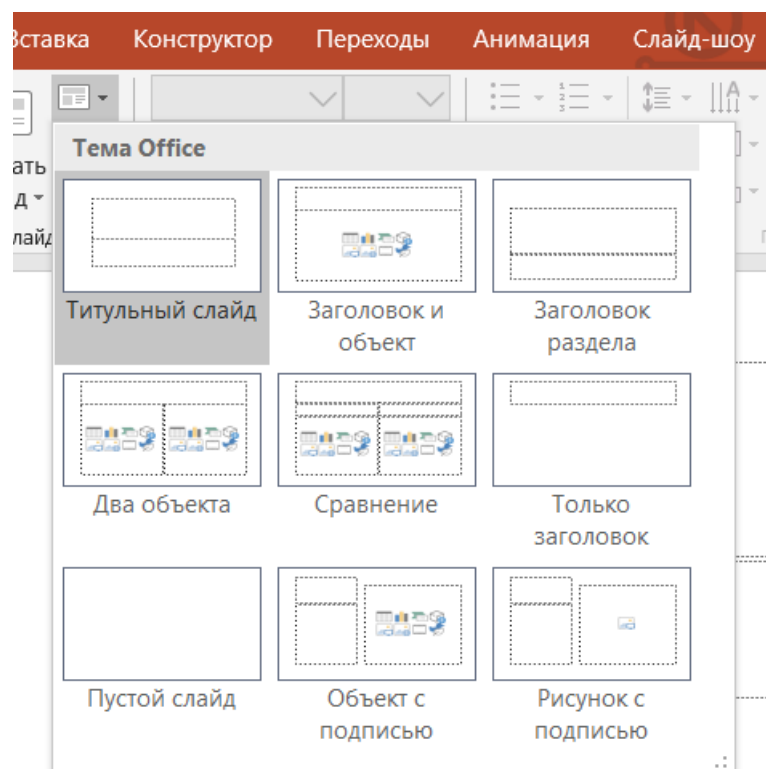
Тема презентации

Тема – набор унифицированных элементов, определяющих внешний вид документа с помощью цвета, шрифтов и графических объектов. Внешний вид презентации можно легко изменить в любой момент, применив другую тему.



Макет слайда (вкладка Главная/Макет)

Макеты слайдов определяют форматирование, размещение и заполнители для всего содержимого на слайде.



Добавление гиперссылок (Вставка/Гиперссылка)

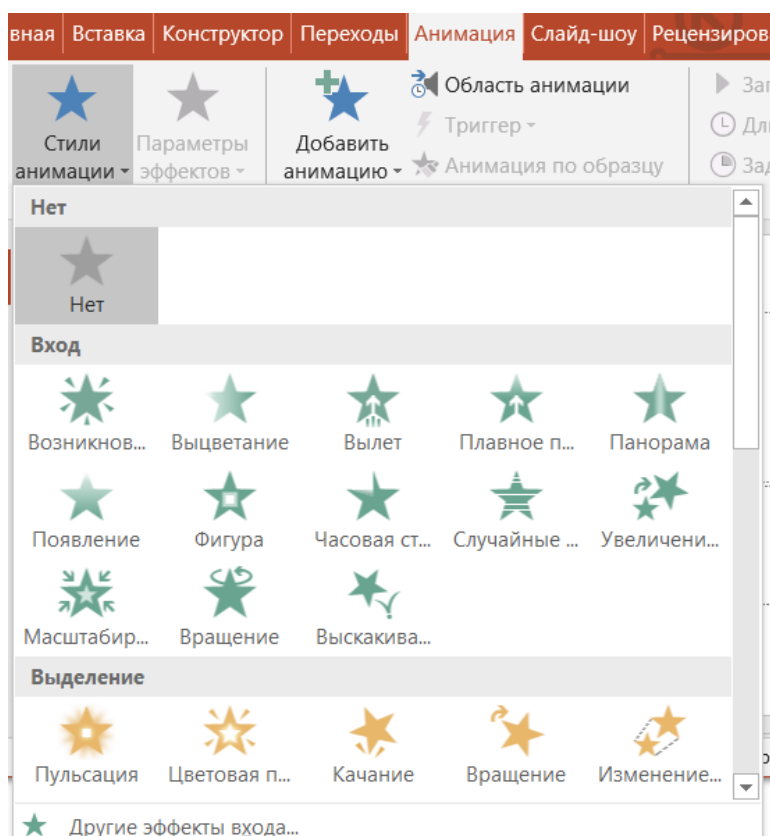
Инструмент Гиперссылка предназначен для перехода с одного слайда на другой, к ресурсу в локальной сети или в Интернете, либо даже к другому файлу или программе.

Чтобы создать гиперссылку выполните следующие действия:

1. Выделите текст, по которому нужно нажать для активации гиперссылки либо можно выделить объект (рисунок smart Art или клип);
2. Перейдите Вставка/Ссылки/Гиперссылка;
3. В диалоговом окне Вставка гиперссылки нажать на соответствующую кнопку в поле Мои адреса, чтобы задать назначение ссылки (т.е. место, на которое указывает ссылка).

Эффекты анимации слайдов (Вкладка Анимация)

Анимационная схема – предопределённый тип смены слайдов и библиотека анимационных эффектов, применяемых к объектам слайда



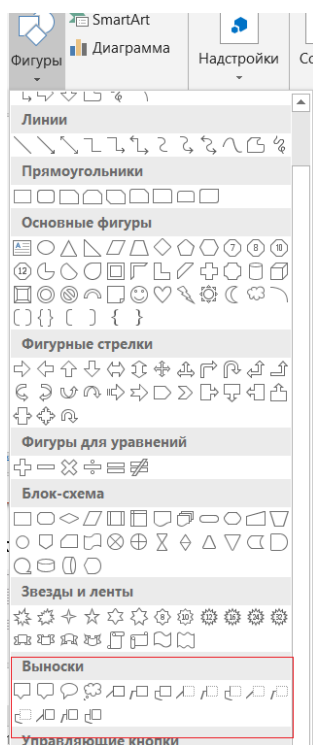
Триггеры

Триггер – это интерактивное средство анимации, позволяющее задать действие выделенному элементу, анимация запускается по щелчку. Через триггер происходит запуск анимационного эффекта или группы эффектов. Триггер можно применить к любому объекту на слайде. Он, как и управляющая кнопка, срабатывает при наведении курсора на по щелчку

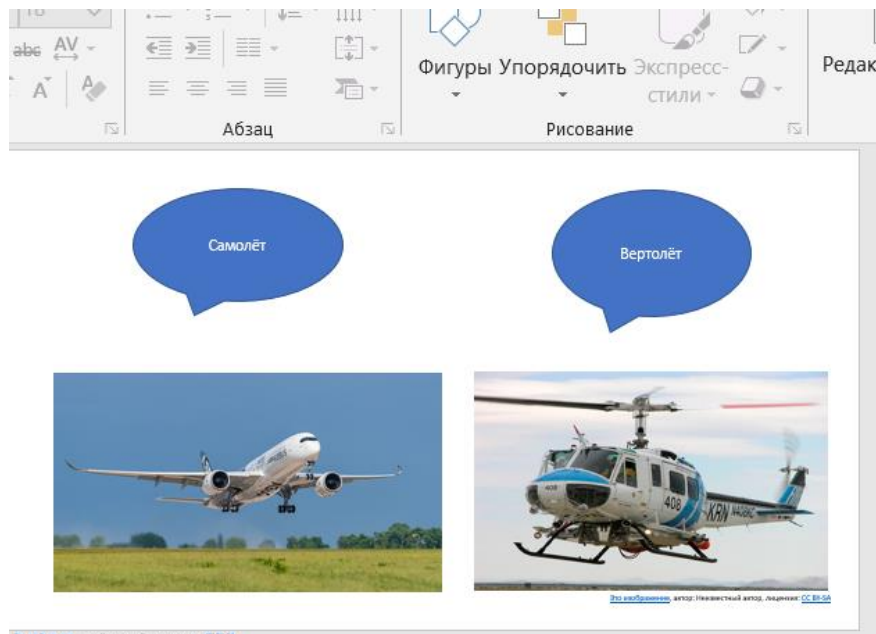
левой кнопки мыши, при этом в момент наведения сам курсор меняет внешний вид на «ладошку».

Для добавления триггер выполните следующие действия:

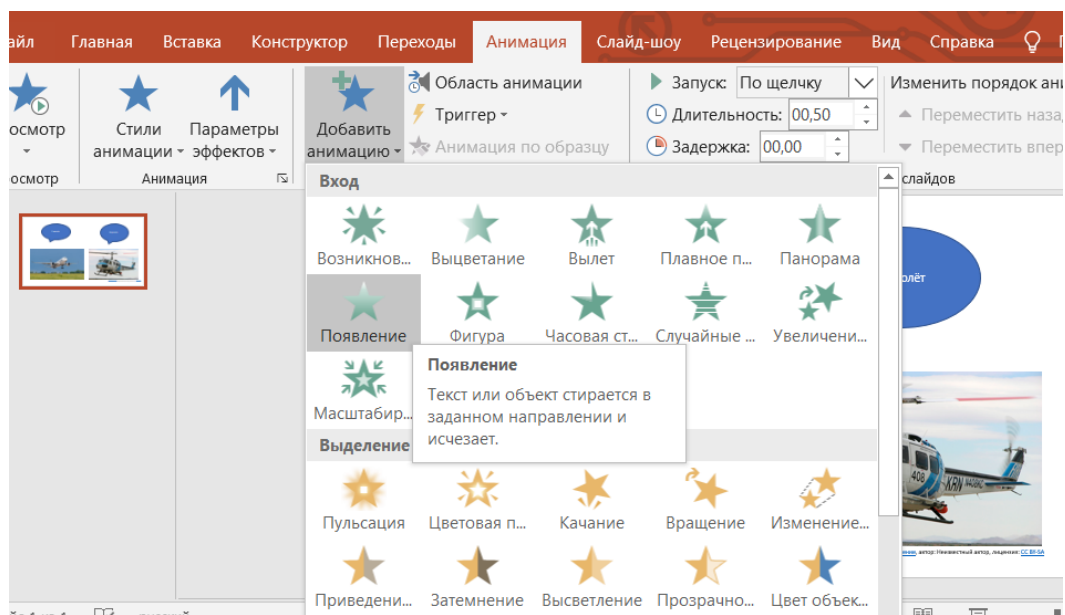
1. Наведите курсор на свободное место на слайде, кликните по нему один раз правой кнопкой мыши, выберите пункт Макет и Пустой слайд.
2. Добавляем изображение: Вставка/Изображение/Рисунки. В открывшемся диалоговом окне выберите необходимые изображения. Если одновременно надо добавить несколько рисунков, удерживайте клавишу CTRL.
3. Вставляем подсказки: выбираем меню Вставка/Фигуры/Выноска.



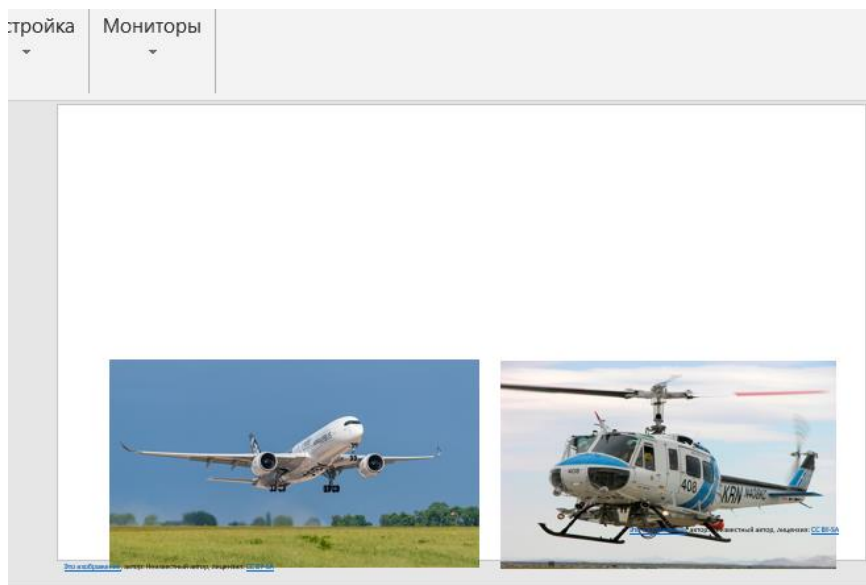
После вставляем в фигуру текст с подсказкой. Выполняем данные действия для двух фото. Должно получиться так:



4. Создаём анимацию. Для этого выделяем фигуру с текстом, выбираем меню Анимация/Добавить анимацию/Появление.



5. Настраиваем триггер. Для этого выделяем подсказку, далее Анимация/Область анимации/Триггер/По щелчку/Рисунок 3. Для второй подсказки применяем ту же схему, только в конце выбираем Рисунок 4. В итоге должно получиться следующее:



Точно также можно поставить триггеры на другие объекты слайда: надписи, рисунки, автофигуры, добавить анимацию, аудио- и видео – эффекты.

Инструмент Word Art (Вкладка Вставка/Word Art)

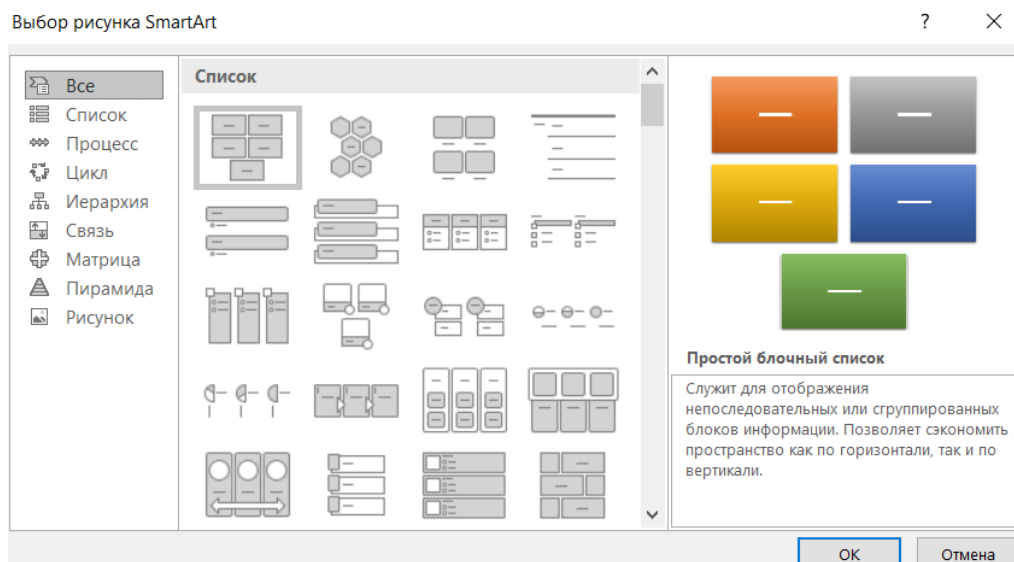
Word Art – это коллекция стилей текста, которые можно добавлять в документ office для создания декоративных эффектов.

Создание специальных эффектов при помощи инструмента word Art осуществляется с панели Вставка/Текст, для этого необходимо:

- Отобразить нужный слайд в режиме слайдов, а затем щёлкнуть на вкладке Вставить кнопку word Art и выбрать стиль текста;
- Создать необходимый текст в появившемся на слайде текстовом блоке;
- Настроить текст надписи, используя стили WordArt: тень, отражение, рельеф и т.д.

Инструмент SmartArt (Вкладка Вставка/SmartArt)

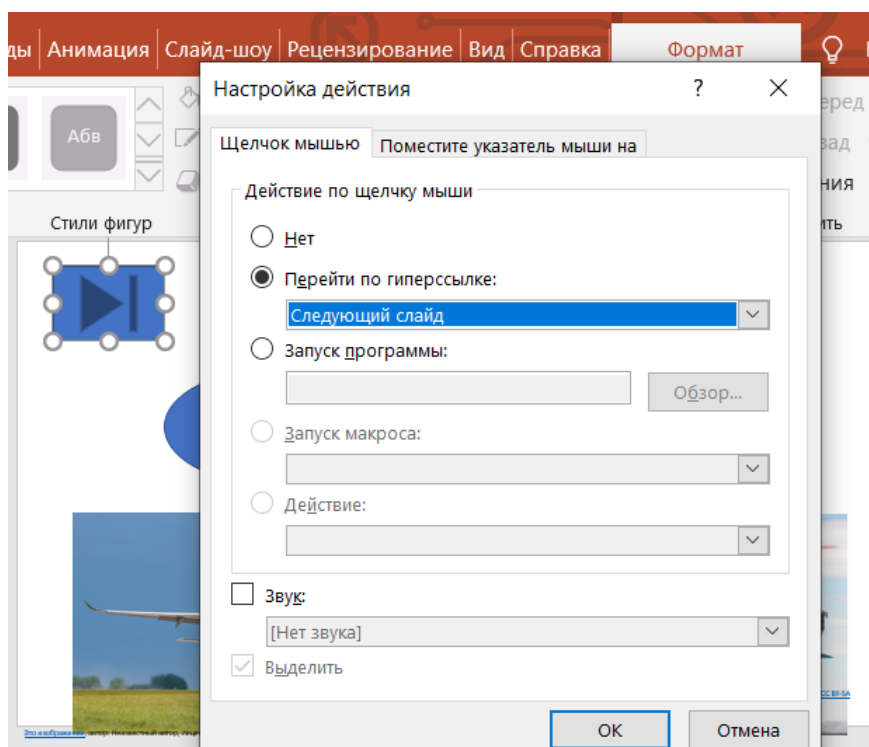
Рисунок SmartArt – это настраиваемое средство, позволяющее передать информацию зрительным образом.



Использование кнопок управления

Для создания кнопок управления необходимо:

1. Создать управляющую кнопку, используя Вкладку Вставка/Фигуры/Управляющие кнопки;
2. В диалоговом окне Настройка действия выбрать переключатель Перейти по гиперссылке- номер слайда.



Контрольные вопросы:

1. Для чего используют редактор создания презентаций MS PowerPoint?

2. Какие режимы просмотра презентации вы знаете?
3. Какие операции можно выполнять на вкладке Структура в левой панели экрана?
4. Как можно добавить колонтитул в нижнюю часть слайда и какие поля он содержит?
5. Как изменить цветовую схему слайдов презентации?
6. Как добавить в слайд звуковые и видеофайлы?
7. Как добавить в слайд графические объекты?
8. Как установить анимацию для перехода от одного слайда к другому?
9. Как установить анимацию для всех объектов слайда?
10. Какое назначение триггеров в MS Power Point?

Практическая работа №7. Создание реляционной базы данных в Microsoft Access.

Цель: Ознакомление с методами и средствами создания баз данных в среде MS Access.

База данных – это поименованная совокупность структурированных данных, относящихся к определённой предметной области.

Реляционная база данных – множество взаимосвязанных двумерных таблиц, каждая из которых содержит сведения об одном объекте предметной области⁸.

Таблицы состоят из полей (столбцов) и записей (строк)



⁸ Асташова, Т.А. Информатика : учебное пособие : [16+] / Т.А. Асташова ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2017. – 108 с. : ил., табл. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=574622>

Поле -элементарная единица логической организации данных, которая соответствует неделимой единице информации.

Запись – совокупность логически связанных полей.

Экземпляр записи – отдельная реализация записи, содержащая конкретные значения её полей.

Каждая таблица базы данных обладает следующими свойствами:

- 1) Все записи содержат одинаковые поля;
- 2) Данные одного поля для всех записей имеют один и тот же тип;
- 3) Каждый столбец имеет уникальное имя;
- 4) Одинаковые строки в таблице отсутствуют;
- 5) Порядок следования строк и столбцов произвольный.

Файл базы данных в программе MS Access содержит следующие объекты: таблицы, запросы, формы, отчёты, макросы, модули и страницы.

Основные данные хранятся в таблицах. Таблица состоит из записей. Записи разбиты на поля, в которых могут храниться данные различных типов.

Основные типы данных

В таблицах могут храниться данные следующих типов: текст, Поле Мемо, Дата/Время, Числовой, Денежный, счётчик, Логический и др.

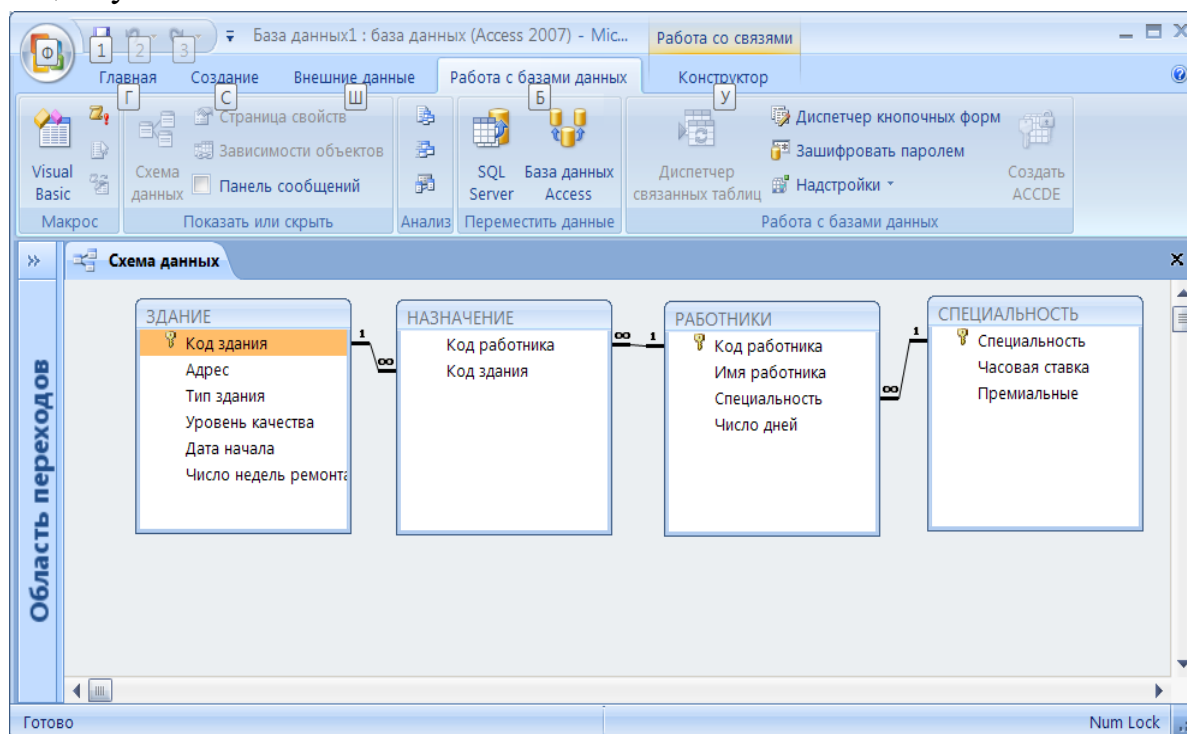
Запросы предназначены для получения информации на основе имеющихся данных.

Правила проектирования базы данных:

1. БД должна быть нормализована, т.е. в БД не должно быть повторяющихся данных. Это достигается разбиением таблиц на более мелкие (выделение данных в справочники).
2. В БД должна быть обеспечена целостность данных, т.е. при изменении одних данных автоматически происходит соответствующее изменение связанных с ним данных. Это достигается наличием логических связей между таблицами. Для связи таблиц между собой используют ключевое поле или простой ключ.

Простым ключом называют поле, каждое значение которого однозначно определяет соответствующую запись.

Задание. Создать базу данных «Строительная фирма» на основе следующей концептуальной схемы:



1. Запустите программу MS Access и создайте новую базу данных.
2. Задайте имя файла базы данных «**Строительная фирма**» и сохраните его в своей папке.
3. В окне базы данных откройте вкладку **Создание/Таблица**.
4. Перейдите в режим **Конструктор**. Введите название таблицы «**ЗДАНИЕ**».
5. В соответствии с таблицей №1 заполните последовательно характеристики всех полей.

Примечание: в открывшемся окне конструктора таблиц в каждой строке спецификации определяются характеристики одного поля записи. В колонке Имя поля определяется имя поля, в колонке Тип данных определяется вид информации, которая будет храниться в данном поле. В нижней части диалогового окна вводятся индивидуальные свойства каждого поля.

Таблица №1 «ЗДАНИЕ»:

Имя поля	Тип данных	Свойства поля
Код здания	Числовой	Размер поля – целое Формат поля – основной Число десятичных знаков – 0 Обязательное поле – Да Индексированное поле – Да (Совпадение не

		допускается)
Адрес	Текстовый	Размер поля – 30 Индексированное поле – Да (совпадения не допускаются)
Тип здания	Текстовый	Размер поля - 20 Индексированное поле – Да (Допускаются совпадения)
Уровень качества	Текстовый	Размер поля – 5 Индексированное поле – Да (Допускаются совпадения)
Дата начала	Дата/время	Формат – краткий формат даты
Число недель ремонта	Числовой	Размер поля – целое Формат поля – основной Число десятичных знаков - 0

- Установите курсор на первую строку таблицы и щёлкните мышью на кнопке **Ключевое поле** на панели инструментов.
- Сохраните созданную таблицу в текущей базе данных под именем «ЗДАНИЕ».
- Закройте окно конструктора таблицы.
- Аналогичным образом создайте таблицы «НАЗНАЧЕНИЕ», «СПЕЦИАЛЬНОСТЬ» и «РАБОТНИКИ».

Таблица №2 «НАЗНАЧЕНИЕ»:

Имя поля	Тип данных	Свойства поля
Код работника	Числовой	Размер поля – целое Формат поля – основной Число десятичных знаков – 0 Обязательное поле – Да Индексированное поле – Да (Допускается совпадение)
Код здания	Числовой	Размер поля – целое Формат поля – основной Число десятичных знаков – 0 Обязательное поле – Да Индексированное поле – Да (Допускается совпадение)

Таблица №3 «СПЕЦИАЛЬНОСТЬ»:

Имя поля	Тип данных	Свойства поля
Специальность	Текстовый	Размер поля – 25

		Индексированное поле – Да (Не допускаются совпадения)
Часовая ставка	Денежный	Формат поля – Денежный Число десятичных знаков – 0 Индексированное поле - Нет
Премии	Денежный	Формат поля – Денежный Число десятичных знаков – 0 Индексированное поле - Нет

Таблица №4 «РАБОТНИКИ»:

Имя поля	Тип данных	Свойства поля
Код работника	Числовой	Размер поля – целое Формат поля – основной Число десятичных знаков – 0 Обязательное поле – Да Индексированное поле – Да (Совпадения не допускаются)
Имя работника	Текстовый	Размер поля – 30 Обязательное поле – Нет Индексированное поле – Нет
Специальность	Текстовый	Размер поля – 25 Индексированное поле – Да (Допускаются совпадения)
Число дней	Числовой	Размер поля – целое Формат поля – основной Число десятичных знаков – 0

Связывание таблиц

1. Выберите меню **Работа с базами данных/ Схема данных**. На экране появятся диалоговые окна **Схема данных** и **Добавление таблицы**.
2. В списке таблиц выделите все таблицы и нажмите кнопку **Добавить**. Закройте окно **Добавление таблицы**.
3. Переместите с помощью мыши используемое для связи поле **Код здания** из таблицы **ЗДАНИЕ** к соответствующему полю таблицы **НАЗНАЧЕНИЕ**. На экране появится диалоговое окно **Изменение связей**.
4. Активируйте опции **Обеспечение целостности данных**, **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**. В поле **Тип отношения** должна быть установка **«один-ко-многим»**.

5. Нажмите кнопку **Создать**. Связь отобразится графически.
6. Переместите поле **Код работника** из таблицы **РАБОТНИК** к соответствующему полю таблицы **НАЗНАЧЕНИЕ**. Аналогично выполните связывание таблиц **РАБОТНИКИ** и **СПЕЦИАЛЬНОСТЬ**.
7. Закройте окно **Схема данных** с сохранением схемы данных.

Ввод записей

1. Откройте таблицу **ЗДАНИЕ** в режиме таблицы.
2. Установите курсор мыши в первую ячейку первой строки и заполните её данными:

Таблица **ЗДАНИЕ**

Имя поля	Данные				
Код здания	5375	6990	1001	9015	6090
Адрес	Крупской 57	Героев Рыбачьего 76	Шабалина 6	Баумана 7	Копытова 56
Тип здания	32/7	65/9	76/9	32/7	76/09
Уровень качества	2	6	4	3	2
Дата начала	06.09.01	09.11.01	08.07.02	12.09.02	12.12.00
Число недель	11	8	9	10	5

3. Аналогично заполните данными другие таблицы:

Таблица **РАБОТНИКИ**

Имя поля	Данные				
Код работника	123	124	125	126	127
Имя	Иванов Н.П.	Соколов Р.Г.	Петров Н.Н.	Сидоров И.Н.	Орлов Н.В.
Специальность					
Число проработанных дней	25	6	12	26	29

Таблица **СПЕЦИАЛЬНОСТЬ**

Имя поля	Данные				
Специальность	Слесарь	Столяр	Маляр	Каменщик	Подсобный
Часовая ставка	80	100	95	124	50
Премияльные	1000	1500	2000	3000	500

Таблица **НАЗНАЧЕНИЕ**

Имя поля	Данные				
Код работника	123	124	125	126	127
Код здания	5375	6990	5375	6990	9015

Практическая работа №8

Выполнение основных операций с базами данных

Цель: познакомиться с основными способами редактирования базы данных.

Редактирование таблицы работник

1. Откройте таблицу **РАБОТНИК** в режиме таблицы и обратите внимание, что третье поле с каждой записи не заполнено.
2. Откройте в окне базы данных таблицу **РАБОТНИК** в режиме конструктора.
3. Щёлкните мышкой в колонке **Тип данных** поля Специальность и из списка доступных типов полей выберите элемент **Мастер подстановок**.
4. В первом диалоговом окне мастера выберите опцию Столбец подстановки будет использовать значения из таблицы или запроса и нажмите кнопку Далее.

5. В следующем окне выберите таблицу **СПЕЦИАЛЬНОСТЬ**. Нажмите кнопку Далее.
6. Из списка Доступные поля выберите поле Специальность и перенесите в список Выбранные поля. Нажмите кнопку Далее.
7. В следующем окне отформатируйте ширину столбца, если это требуется. Нажмите кнопку Далее.
8. Подтвердите название таблицы. Нажмите кнопку Далее.
9. Нажмите кнопку Готово. Подтвердите сохранение таблицы.
10. Перейдите в режим таблицы. Щёлкните мышью в поле Специальность таблицы **РАБОТНИК**. Откроется список, состоящий из колонки: Специальность.
11. Щёлкните по первой строке списка, и наименование специальности занесётся в поле ввода. Аналогично заполните поля всех моделей, выбирая их из списка в произвольном порядке.
12. Закройте таблицу **РАБОТНИК**.

Создание простого запроса

Задание 1. Создайте запрос на выборку данных о работниках, которые назначены на ремонт зданий.

1. В окне базы данных перейдите на вкладку Запросы и нажмите Простой запрос. Откроется первое окно Создание простых запросов.
2. Выберите из открывающегося списка Таблицы и запросы таблицу **ЗДАНИЕ**. В списке Доступные поля щёлкните дважды на поле Код здания. Поле переместится в Выбранные поля.
3. Выберите из списка Таблицы и запросы таблицу **Работник** и перенесите поля Имя и Специальность. Щёлкните на кнопке Далее.
4. Во втором диалоговом окне выделите строку создание подробного отчёта. Щёлкните на кнопке Далее.
5. Введите название запроса **Запрос_1** и выделите вариант Открыть запрос для просмотра данных.
6. Щёлкните на кнопке Готово. Мастер простых запросов закроется, а запрос откроется для просмотра в режиме таблицы.

7. Просмотрите полученную информацию. Следует заметить, что поле Специальность не, в данном запросе не даёт никакой информации, поэтому его можно скрыть.
8. Отобразите запрос в режиме конструктора, который позволяет редактировать запросы.
9. На бланке запроса в столбце Специальность щёлкните в ячейке Вывод на экран, чтобы убрать флажок.
10. Щёлкните в столбце Имя в ячейке Сортировка, щёлкните на стрелке и выделите в списке По возрастанию.
11. Переключитесь в режим таблицы.
12. Посмотрите результат запроса. Поле Специальность больше не отражается в режиме просмотра запроса.
13. Закройте окно запроса с сохранением информации.

Задание 2. Создайте запрос на выборку данных о работниках, назначенных на ремонт здания с кодом 6990.

1. Создайте копию запроса Задание_1 и переименуйте её в Задание_2.
2. Выделите запрос Задание_2 и нажмите кнопку Конструктор для возможности редактирования запроса.
3. На бланке запроса щёлкните в столбце Код здания в ячейке Условие отбора.
4. Введите в эту ячейку 6990, что является условием отбора. Нажмите клавишу Enter.
5. На панели инструментов конструктора перейдите в режим таблицы.
6. Закройте окно запроса с сохранением изменений.

Задание 3. Создайте запрос на выборку данных о зданиях, ремонт которых начат после 01.01.2000 и закончен до 01.12.00.

1. В окне базы данных перейдите на вкладку Запросы и создайте Новый запрос. Щёлкните дважды на строке Конструктор. Откроется окно бланка запроса и диалоговое окно Добавление таблицы.

2. В списке таблиц выделите таблицу **ЗДАНИЕ** и нажмите кнопку **Добавить**. Таблица отразится в бланке запроса. Закройте диалоговое окно.
3. В списке полей запроса щёлкните дважды на полях Код здания и Дата начала ремонта.
4. На бланке запроса щёлкните в столбце Дата начала в ячейке Условие отбора.
5. Введите в ячейку условие запроса: $>01.01.00 \text{ AND } <30.12.00$ нажмите клавишу Enter для ввода условия. Выражение преобразуется к виду $\#01.01.00 \text{ AND } \#30.12.00\#$.
6. Посмотрите результат в режиме таблицы.
7. Закройте окно запроса с сохранением изменений и сохраните созданный запрос под именем **Задание_3**.

Добавление в запрос вычисляемых полей

Задание 4. Добавьте в запрос **Задание_3** информацию об окончании ремонта зданий.

1. Создайте копию запроса **Задание_3** и переименуйте её в **Задание_4**.
2. Откройте запрос **Задание_4** в режиме конструктора.
3. Щёлкните правой кнопкой мыши в первом пустом поле бланка запроса. В появившемся контекстном меню выберите строку **Построить**. Появится диалоговое окно **Построитель выражений**.
4. Для создания нового поля в верхней части окна надо ввести следующее выражение: **Дата окончания:=Дата начала+Число недель*7**. Для этого введите текст **Дата окончания:**, щёлкните по значку **=**, дважды щёлкните на слове **Таблицы**, затем на названии таблицы **Здание**. В правом окне отобразятся данные этой таблицы.
5. Выделите поле **Дата начала** и укажите кнопку **Вставить**, затем щёлкните знак сложения **+**, вставьте **Число недель**, введите умножение на **7**. Нажмите кнопку **ОК**. В бланке запроса появится новое поле.
6. Перейдите в режим таблицы и посмотрите результаты. Сохраните запрос под именем **Задание_4**.

Задание для самостоятельной работы: (на выбор)

Задание №1

1. Создать базу данных «Библиотека», состоящую из таблиц: «Список книг»; «Читатели»; «Выдачи».
2. Предусмотреть в таблице «Список книг» поля:
 - ✓ № п/п (ключевое поле)
 - ✓ Тип книги
 - ✓ Название
 - ✓ Автор
 - ✓ Переводчик (если есть)
 - ✓ Издательство
 - ✓ Год издания
 - ✓ Тип переплёта
3. Предусмотреть в таблице «Читатели» поля:
 - ✓ № читателя (ключевое поле)
 - ✓ Фамилия
 - ✓ Имя
 - ✓ Отчество
 - ✓ Адрес телефон
4. Предусмотреть в таблице «Выдачи» поля:
 - ✓ № выдачи (ключевое поле)
 - ✓ Название (столбец подстановки)
 - ✓ Кому выдано (столбец подстановки)
 - ✓ Дата выдачи
 - ✓ Дата возврата
 - ✓ Примечания

6. Создать формы: «Список книг» для просмотра и редактирования списка книг и «Читатели книг» - сложную форму для учёта выдач книг. Создать сводную таблицу и диаграмму с количеством прочитанных книг у читателей.
7. Создать отчёты: «Список книг по алфавиту»; «Список книг по авторам» и «Читатели» (читатели + прочитанные книги).
8. Создать кнопочную форму для запуска форм и отчётов.
9. Создать и распечатать отчёт о проделанной работе (используя MS Word).

Задание №2

1. Создать базу данных «Оборудование», состоящую из таблиц: «Список оборудования», «Список помещений», «Список ответственных».
2. Предусмотреть в таблице «Список оборудования» поля:
 - № п/п (ключевое поле).
 - Тип.
 - Наименование.
 - Количество.
 - Стоимость.
 - Состояние (отл, хор, удовл, плох).
 - Помещение (присоед. столбец).
 - Ответственный (присоед. столбец).
3. Предусмотреть в таблице «Список помещений» поля:
 - № помещения (ключевое поле).
 - Примечания.
4. Предусмотреть в таблице «Список ответственных» поля:
 - № ответственного (ключевое поле).
 - Фамилия.
 - Имя.
 - Отчество.
 - Телефон.
5. Создать формы: «Список оборудования» для просмотра и редактирования списка оборудования и «Ответственный» - сложную форму для учёта оборудования.
6. Создать отчёты: «Список оборудования по ответственным», «Список оборудования по стоимости» и «Список изношенного оборудования».

7. Создать кнопочную форму для запуска форм и отчётов.
8. Создать и распечатать отчёт о проделанной работе (используя MS Word)⁹.

Контрольные вопросы:

1. Что такое база данных? Что такое СУБД? Что такое MS Access?
2. Назовите основные функции СУБД.
3. Что такое реляционная база данных?
4. Назовите основные элементы реляционной базы данных.
5. Что такое ключ и ключевое поле?
6. Назовите и понятие основным объектам базы данных.
7. Назовите два этапа процесса проектирования базы данных.
8. Что такое логическая организация данных?
9. Назовите основные типы связей между таблицами.

Практическая работа №9. Язык программирования Pascal ABC.NET.

Линейные и разветвляющиеся алгоритмические конструкции.

Цель: выработка навыков составления алгоритмов линейной и разветвляющейся структуры.

Методические указания

Алгоритм – точный порядок действий, определяющий процесс, ведущий от исходных данных к искомому результату и обладающий следующими свойствами: определённой, массовостью, результативностью, дискретностью, конечностью.

Наиболее распространёнными способами описания алгоритма являются: словесное описание, псевдокод, блок – схема и программа.

⁹ Соболев, Б. В. Практикум по информатике: / Б. В. Соболев [и др.]; под ред. Б.В. Соболева. – Ростов н / Д: Феникс, 2009. – 509 [1] с. – (Высшее образование).

Псевдокод – описание алгоритма на частично формализованном естественном языке. В псевдокоде используются служебные слова и конструкции, а также математическая символика, которая позволяет формализовать запись алгоритма.

Блок – схема – это графический способ описания алгоритма с помощью описания геометрических фигур (блоков), соединённых между собой линиями переходов, определяющими очередность выполнения действий.

Структура любой программной единицы (программы, процедуры или функции) в общем виде выглядит следующим образом:

<объявление программной единицы>;

Uses<список используемых модулей>;

Label<список меток>;

Const<список констант>;

Type<описание пользовательских типов>;

Var<список переменных>;

Procedure или **function** <раздел подпрограмм>;

Begin

<раздел исполняемых операторов>

End.

Типы данных

Тип объекта определяет множество допустимых значений, которые может иметь этот объект, а также множество допустимых операций, которые применимы к нему. Кроме того, тип определяет формат внутреннего представления данных в памяти компьютера¹⁰.

¹⁰ Информатика: практикум: [16+] / сост. С.О. Алтухова, З.А. Кононова ; Липецкий государственный педагогический университет имени П. П. Семенова-Тян-Шанского. – Липецк: Липецкий государственный педагогический университет имени П.П. Семенова-Тян-Шанского, 2018. – Ч. 1. – 72 с. : ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=576673>

К простым (скалярным) типам данных относятся типы данных, значения которых не содержат составных частей:

- Порядковый: целый, символьный (литерный), логический (булевый), перечисляемый, интервальный (тип диапазон);
- Вещественный.

Целый тип

Тип	Диапазон	Требования к памяти (в байтах)
ShortInt	-128..+127	1 со знаком
SmallInt	-32768..+32767	2 со знаком
Integer/LongInt	-2147483648..+2147483647	4 со знаком
Int64	$\sim -2^{63}..+2^{63}-1$	8 со знаком
Byte	0..255	1 без знака
Word	0..65535	2 без знака
Longword	0..4294967295	4 без знака

Символьный тип

Символьный тип **Char** занимает 1 байт и определяется множеством значений кодовой таблицы ЭВМ. Каждому символу приписывается целое число в диапазоне 0..255.

Логический тип

Логический тип данных Boolean представлен всего двумя значениями – true (истина)=1 и false (ложь)=0.

Перечисляемый тип

Перечисляемый тип задаётся перечислением всех доступных значений.

Формат:

Type<имя_типа>=(<значение1>,...,<значение N>);

Var<идентификатор>:<имя_типа>;

Интервальный тип (тип диапазон)

Интервальный тип описывается путём задания двух констант, определяющих границы допустимых для данного типа значений. Значение первой константы должно быть меньше значения второй.

Формат:

Type<имя_типа>=<мин. значение>..<макс.значение>;

Var<идентификатор>:<имя_типа>;

Пример:

Type Day1_31=1..31;
Var day1,day2:Day1_31;

Рациональнее определить интервальный тип более универсальным способом, задав границы диапазона не значениями констант, а их именами:

Const
 Min=1; max=31;
Type
 Day1_31=min..max;
Var day1, day2: Day1_31;

Интервальный и перечисляемый типы занимают в памяти 1 байт, поэтому не могут содержать более 256 символов.

Вещественные типы

Вещественные типы			
Тип	Диапазон	Размер (в байтах)	Точность (число цифр мантииссы)
Real	от 10^{-39} до 10^{38}	6	11-12
Single	от 10^{-45} до 10^{38}	4	7-8
Double	от 10^{-324} до 10^{308}	8	15-16
Extended	от 10^{-4932} до 10^{4932}	10	19-20

Функции преобразования типов

Round(x : real) : LongInt – округляет вещественное значение x до ближайшего целого числа; если число находится ровно посередине, то результат всегда четный.

Trunc(x : real) : LongInt – выделяет целую часть вещественного числа.

Floor(x : real) : LongInt – округляет вещественное число в сторону уменьшения.

Ceil(x : real) : LongInt – округляет дробь в сторону увеличения.

Frac(x : real) : Extended – возвращает дробную часть числа.

Арифметические выражения

Унарные арифметические операции: + (сохранение знака) и – (отрицание знака)- относятся к знаку числа и не меняют типа числа.

Бинарные арифметические операции: +(сложение), - (вычитание), * (умножение), /(деление)

Тип значения арифметического выражения определяется типом операндов и операциями. Если в операции участвуют целочисленные

операнды, то результат операции также будет целочисленного типа. Если хотя бы один из операндов принадлежит к вещественному типу, то результат также будет вещественным. Исключением является операция деления, результат которой всегда будет вещественным.

С целочисленными переменными дополнительно можно производить следующие арифметические операции:

Div – целочисленное частное от деления двух чисел;

Mod- целочисленный остаток от деления двух чисел.

Пример:

11 **div** 5=2

10 **div** 3=3

2 **div** 3=0

10 **mod** 5=0

11 **mod** 5=1

10 **mod** 3=1

14 **mod** 5=4

Логические выражения

Результатом логического выражения является значение True (Истина) или False (Ложь). Такие выражения чаще всего используются в условных операторах и операторах цикла. Логические выражения могут содержать:

- Логические константы True и False;
- Логические переменные типа Boolean;
- Операции сравнения (отношения);
- Логические операции;
- Круглые скобки.

Логические операции: And, Or, Xor и Not

Таблица истинности

Операция	Описание	Операнд 1	Операнд 2	Результат
Not	Отрицание	False		True
		True		False
And	Логическое И	False	False	False
		False	True	False
		True	False	False
		True	True	True
Or	Логическое ИЛИ	False	False	False
		False	True	True
		True	False	True
		True	True	True
Xor	Исключающее ИЛИ	False	False	False
		False	True	True
		True	False	True
		True	True	False

Приоритеты операций (в порядке убывания)

1. Унарные: +, -, Not («+», «-» - знаки числа, например +2, -2).
2. Мультипликативные: *, /, div, mod, And.
3. Аддитивные: +, -, Or, Xor.
4. отношения: =, <, >, <=, >=, <>.

В первую очередь выполняются операции, имеющие максимальный приоритет. Операции, имеющие одинаковый приоритет, выполняются в порядке встречаемости выражений слева направо. Скобки позволяют увеличить приоритет операции до максимального.

Арифметические функции

Для работы с арифметическими функциями, помеченными *, необходимо подключить модуль **Math**.

Арифметические функции

Формат функции	Описание функции	Математическая запись	Запись на языке Pascal
Abs(x):real/integer	абсолютное значение аргумента x	$f= 5-b $	<code>f:=abs(5-b);</code>
Sqr(x):real/integer	квадрат x	$f=x^2$	<code>f:=sqr(x);</code>
Sqrt(x):real	квадратный корень из x	$f=x-5$	<code>f:=sqrt(x-5);</code>
Pi:real	число π (3,14159358...)	$f=2x+\pi$	<code>f:=2*x+pi;</code>
*Cos(x):real	косинус x, x задается в радианах	$f=\cos(x)$	<code>f:=cos(x);</code>
*Sin(x):real	синус x, x задается в радианах	$f=\sin(x)$	<code>f:=sin(x);</code>
*Tan(x):real	тангенс x, x задается в радианах	$f=\operatorname{tg}(x)$	<code>f:=tan(x);</code>
*Exp(x):real	экспонента	$f=e^{x+6}$	<code>f:=exp(x+6);</code>
*Ln(x):real	натуральный логарифм x	$f=\ln x$	<code>f:=ln(x);</code>
*IntPower(x,y): integer	целочисленное возведение в степень x^y	$f=x^y$	<code>f:=(x,y);</code>
*Power(x,y):real	возведение в степень x^y , x и y вещественные		
Random(x):integer;	псевдослучайное число в интервале от 0 до x-1		<code>f:=random(x);</code>

Составной оператор

Составной оператор представляет собой группу из произвольного числа любых операторов, отделённых друг от друга точкой с запятой, и ограниченную операторными скобками **begin** и **end**.

Формат:

Begin

<оператор 1>;

<оператор 2>;

...;

<оператор n>;

End;

Операторы ввода-вывода данных

Оператор ввода

Для ввода данных используется оператор ввода `read` или `readln`.

Формат:

Read(<список ввода>);

Или

Readln(<список ввода>);

Операторы вывода

Оператор **writeln** или **write** служит для вывода на экран каких-либо данных (текста, результатов выполнения программы, значений переменных и т.п.).

Формат:

Write(<список вывода>);

или

Writeln(<список вывода>);

Условный оператор

Условный оператор обеспечивает выполнение или невыполнение некоторых операторов в зависимости от соблюдения определённых условий. Условие представляет собой выражение логического типа.

Имеет два формата: строчный и блочный.

Строчный формат:

If <условие> **then** <оператор1> [**else** <оператор2>];

Блочный формат:

if <Условие> then	
begin	
<Оператор1>;	} 1-ая группа составного оператора
<Оператор2>;	
...;	
<ОператорN>;	
end	
else	
begin	
<Оператор1>;	} 2-ая группа составного оператора
<Оператор2>;	
...;	
<ОператорN>;	
end;	

Оператор выбора

Оператор выбора позволяет сделать выбор из произвольного числа имеющихся вариантов. Если вариантов всего два, то можно использовать условный оператор **if**, если же вариантов гораздо больше, то используется оператор выбора **Case**.

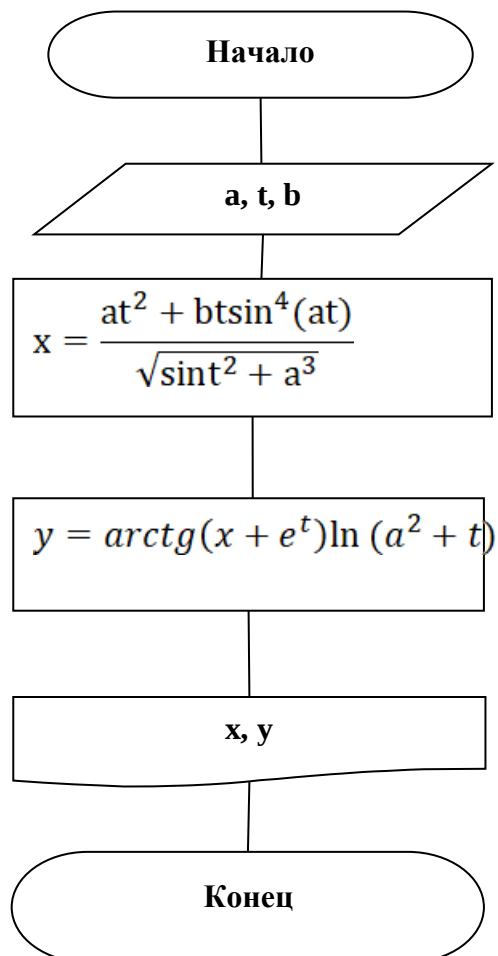
Формат:

```

Case <Выражение-селектор> of
  <Список1> : <Оператор1>;
  <Список2> : <Оператор2>;
  ...
  <СписокN> : <ОператорN>;
[else <Оператор>;]
end;

```

Задача 1. Вычислить: $x = \frac{at^2 + b\sin^4(at)}{\sqrt{\sin^2 t + a^3}}$, $y = \arctg(x + e^t) \ln(a^2 + t)$, если значения переменных a , t и b вводится с клавиатуры.



Программа на Pascal ABC:

```

Program primer_1;
Var a, t, b, x, y: real;
Begin
  Writeln (' введите переменные a,t,b');
  Readln (a, t, b);
  X:=a*t*t+b*t*sqr(sqr(sin(a*t)))/sqrt(sin(t*t)+a*a*a);
  Y:=arctan(x+exp(t))*ln(a*a+t);

```

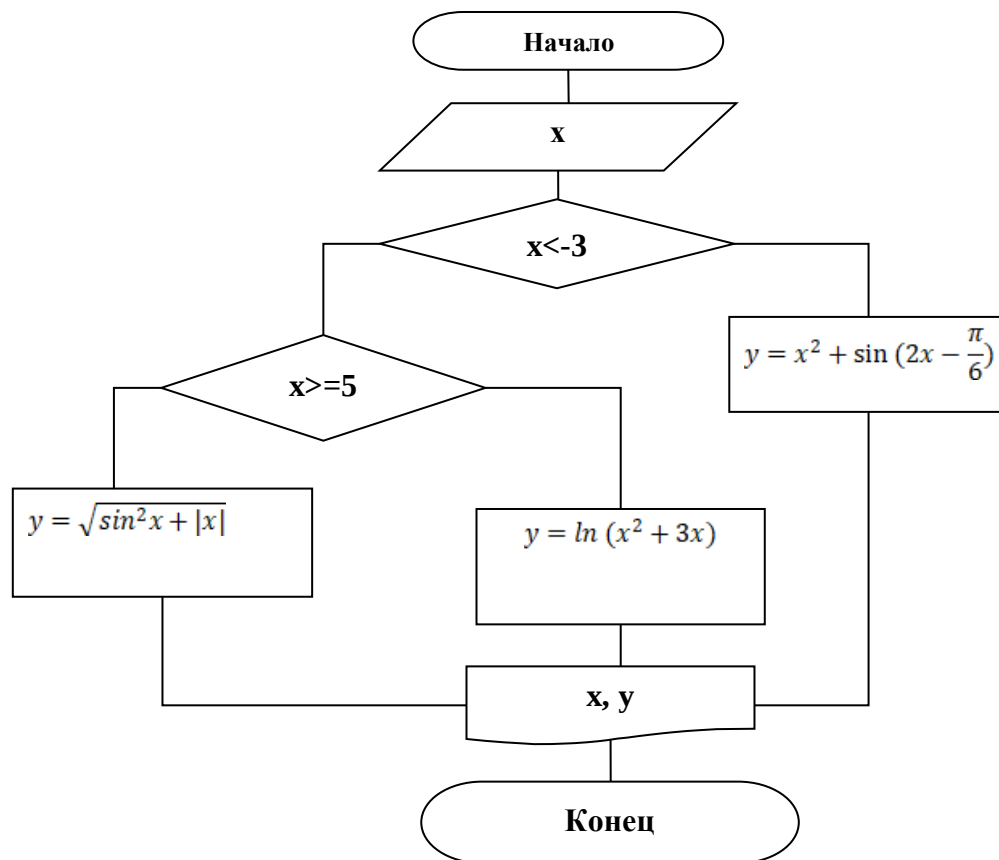
```

Writeln('x=',x:7:3);
Writeln('y=',y:7:3);
End.

```

Задача 2.

Дано x . Вычислить $y = \begin{cases} x^2 + \sin\left(2x - \frac{\pi}{6}\right), & \text{если } x < -3 \\ \sqrt{\sin^2 x + |x|}, & \text{если } -3 \leq x < 5 \\ \ln(x^2 + 3x), & \text{если } x \geq 5 \end{cases}$



Программа на Pascal ABC:

```

Program primer_2;
Var x, y: real;
Begin
    Writeln (' введите переменную x');
    Readln (x);
    If x<-3 then y:=x*x+sin(2*x+pi/6)
    Else
        If x>=5 then Y:=ln(x*x+3*x) else y:=sqrt(sqr(sin(x))+abs(x));

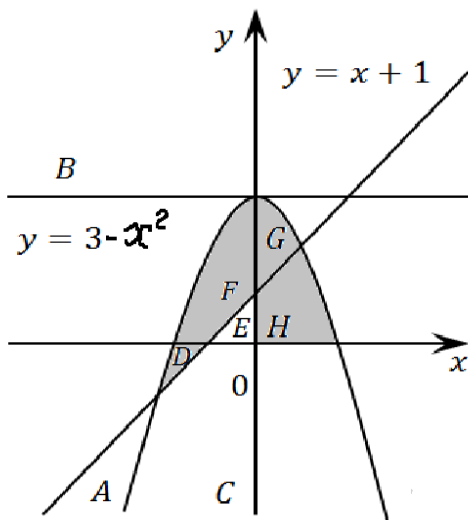
```

```

Writeln('x=',x:7:3, 'y=',y:7:3);
End.

```

Задача 3. Написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы).

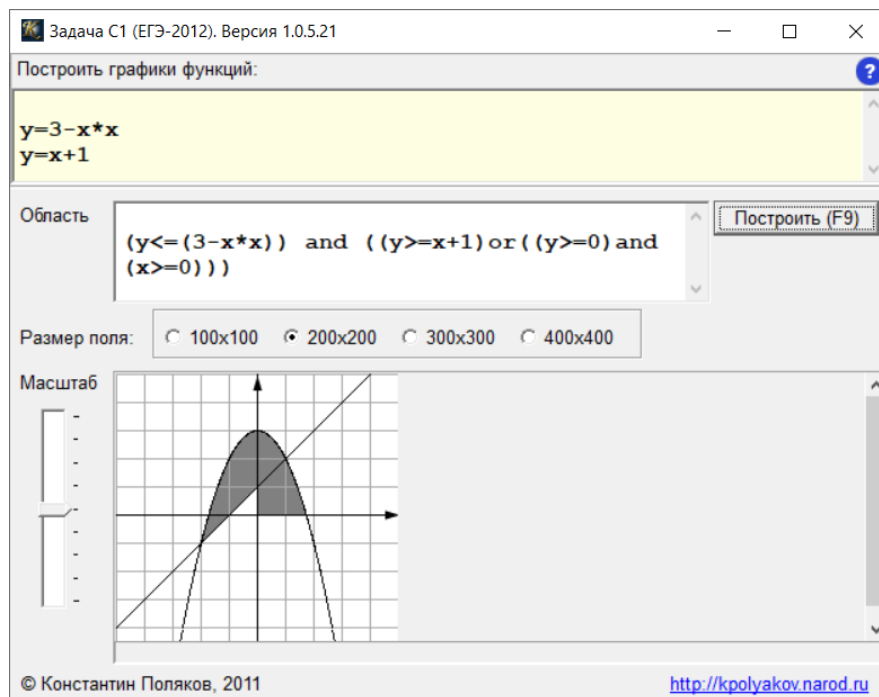


Программа на Pascal ABC:

```

Program Zadacha;
Var x,y: real;
begin
  writeln('Введите координаты точки');
  readln(x,y);
  if (y<=(3-x*x)) and ((y>=x+1) or ((y>=0) and (x>=0))) then
    write('принадлежит заштрихованной области')
  else
    write('не принадлежит заштрихованной области');
end.

```



Задача 4. Программа, которая по введённому номеру дня недели выводит на экран монитора его название на русском языке.

Program Day_week;

Var day:byte;

Begin

 Write ('введите номер дня недели:');

 Readln (day);

Case day **of** {вычисление значения селектора и выбор}

 1: writeln ('понедельник');

 2: writeln ('вторник');

 3: writeln (' среда');

 4: writeln ('четверг');

 5: writeln ('пятница');

 6: writeln ('суббота');

Else

 Writeln('воскресенье');

End;

End.

Варианты заданий для самостоятельной работы:

Задание 1. Напишите программу для расчёта по двум формулам. Предварительно подготовьте тестовые примеры для второй формулы с

помощью калькулятора (результаты вычисления по обеим формулам должны совпадать). Отсутствующие в языке функции выразите через имеющиеся¹¹.

1.	$z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha;$	$z_2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right)$
2.	$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha};$	$z_2 = 2 \sin \alpha.$
3.	$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha};$	$z_2 = \operatorname{tg} 3\alpha.$
4.	$z_1 = \cos^2\left(\frac{3}{8}\pi - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8}\pi + \frac{\alpha}{4}\right);$	$z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}.$
5.	$z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha;$	$z_2 = \cos^2 \alpha + \cos^4 \alpha.$

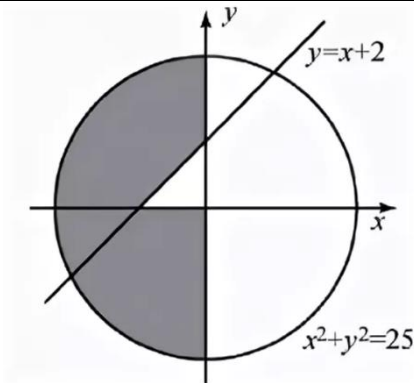
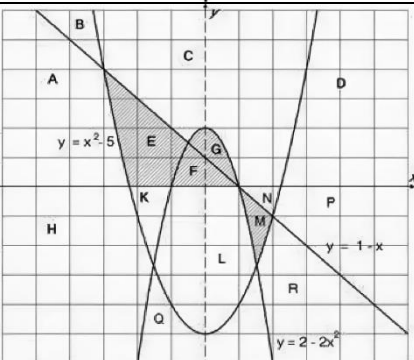
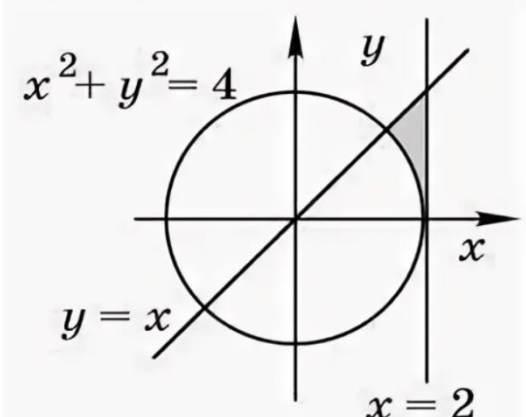
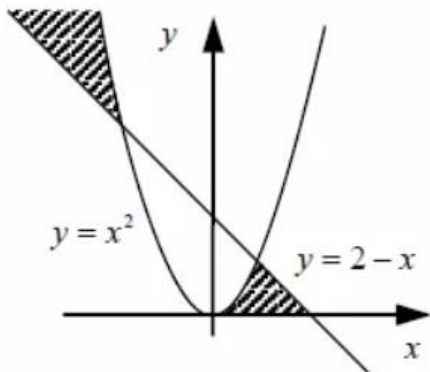
Задание 2. Дано действительное число x . Вычислите $y(x)$, если:

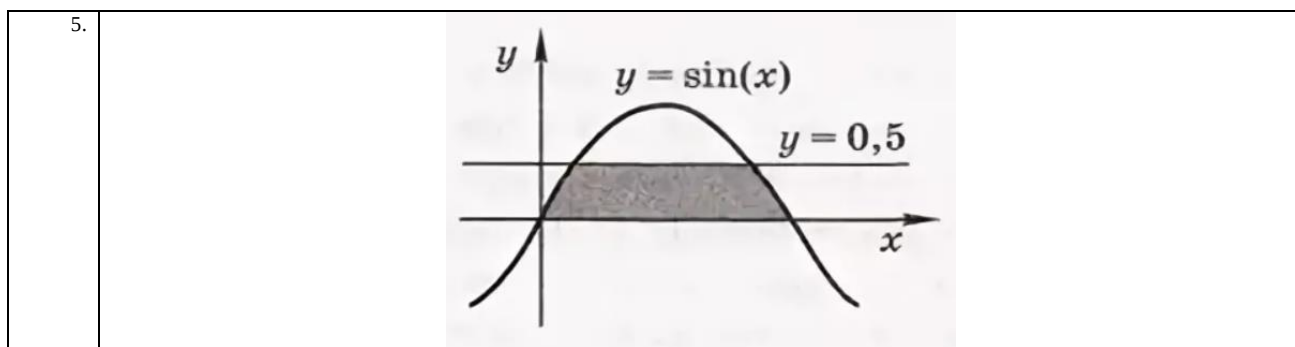
1.	$y = \begin{cases} \sqrt[3]{x^2 + 3} + \sqrt{\log_2 x }, & \text{если } x < 0 \\ \sin^2\left(x + \frac{\pi}{4}\right) - \operatorname{tg} \frac{x}{3}, & \text{если } 0 \leq x \leq 5 \\ \frac{x^3 + x^2 + 3}{x + 5}, & \text{если } x > 5 \end{cases}$
----	--

¹¹ Павловская Т.А. Паскаль. Программирование на языке высокого уровня: Учебник для вузов. – СПб.: Питер, 2008. – 393 с.: ил.

2.	$y = \begin{cases} \sqrt[3]{x^2 + 3} + \sqrt{\log_2 x }, & \text{если } x < 0 \\ \sin^2(x + \frac{\pi}{4}) - \operatorname{tg} \frac{x}{3}, & \text{если } 0 \leq x \leq 5 \\ \frac{x^3 + x^2 + 3}{x + 5}, & \text{если } x > 5 \end{cases}$
3.	$f(x) = \begin{cases} x^2 - a, & x < 0, \\ 4x + \sqrt{x}, & 0 \leq x < a, \\ \ln \left \frac{x}{a} \right , & a \leq x. \end{cases}$
4.	$y = \begin{cases} \sqrt[3]{x^2 + 3} + \sqrt{\log_2 x }, & \text{если } x < 0 \\ \sin^2(x + \frac{\pi}{4}) - \operatorname{tg} \frac{x}{3}, & \text{если } 0 \leq x \leq 5 \\ \frac{x^3 + x^2 + 3}{x + 5}, & \text{если } x > 5 \end{cases}$
5.	$y = \begin{cases} \sqrt[3]{x^2 + 3} + \sqrt{\log_2 x }, & \text{если } x < 0 \\ \sin^2(x + \frac{\pi}{4}) - \operatorname{tg} \frac{x}{3}, & \text{если } 0 \leq x \leq 5 \\ \frac{x^3 + x^2 + 3}{x + 5}, & \text{если } x > 5 \end{cases}$

Задание 3. Написать программу, которая определяет, попадает ли точка с заданными координатами в область, закрашенную на рисунке серым цветом. Результат работы программы вывести в виде текстового сообщения.

1.	 A Cartesian coordinate system showing a circle centered at the origin with the equation $x^2 + y^2 = 25$. A straight line with the equation $y = x + 2$ passes through the circle. The region inside the circle and to the left of the line is shaded gray.
2.	 A Cartesian coordinate system with a grid. Three curves are plotted: $y = x^2 - 5$, $y = 1 - x$, and $y = 2 - 2x^2$. Various regions are labeled with letters A through Q. Some regions are shaded with different patterns: a gray diagonal line pattern for regions like E and F, and a black diagonal line pattern for regions like G and H.
3.	 A Cartesian coordinate system showing a circle centered at the origin with the equation $x^2 + y^2 = 4$. A straight line with the equation $y = x$ passes through the circle. A vertical line is drawn at $x = 2$. The region inside the circle and to the right of the line $y = x$ is shaded gray.
4.	 A Cartesian coordinate system showing a parabola $y = x^2$ and a straight line $y = 2 - x$. The region between the parabola and the line in the first quadrant is shaded with a black diagonal line pattern.



Требования к оформлению отчёта по практической работе

Отчёт должен содержать следующую информацию:

1. Название практической работы.
2. Содержание задания вашего варианта.
3. Описание структуры данных: входные и выходные данные, их тип, диапазон допустимых в данной задаче значений.
4. Блок-схема алгоритма решения задачи.
5. Код программы на языке программирования.
6. Контрольный пример.

Порядок защиты практической работы

Порядок выполнения практических работ осуществляется в соответствии с индивидуальным заданием, согласно номеру варианта, который определяет преподаватель. Защита практической работы осуществляется в устной форме по контрольным вопросам.

Практическая работа №10. Циклические алгоритмы.

Цель: выработка навыков составления алгоритмов циклической структуры.

Циклической (или циклом) называют алгоритмическую конструкцию, в которой некая, идущая подряд группа действий (шагов) алгоритма может

выполняться несколько раз, в зависимости от входных данных или условия задачи¹². Группа повторяющихся действий на каждом шагу цикла называется телом цикла. Любая циклическая конструкция содержит в себе элементы ветвящейся алгоритмической конструкции.

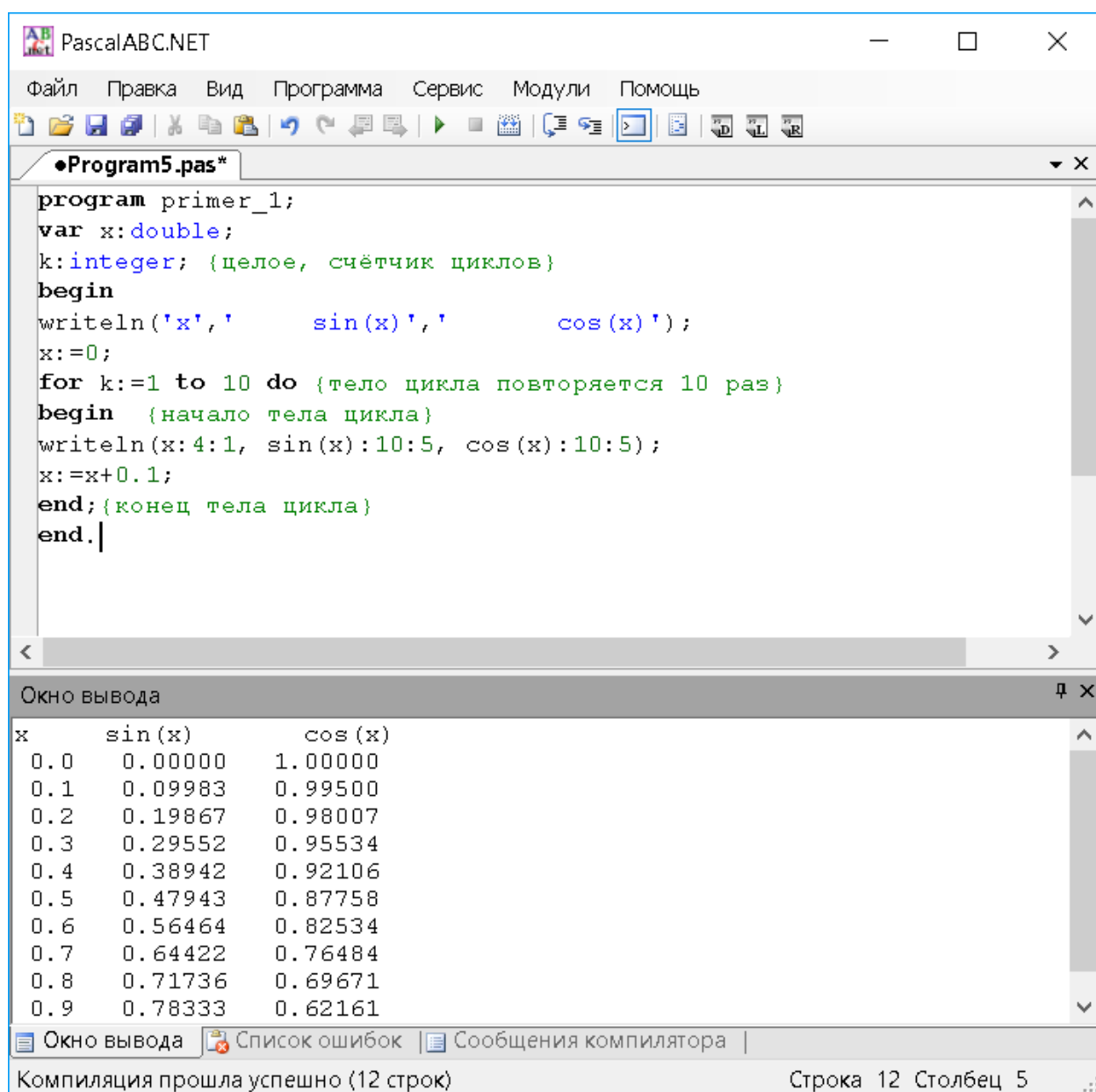
Арифметический цикл

В арифметическом цикле число его шагов (повторений) однозначно определяется правилом изменения параметра, которое задаётся с помощью начального (N) и конечного (K) значений параметра и шагом (h) его изменения.

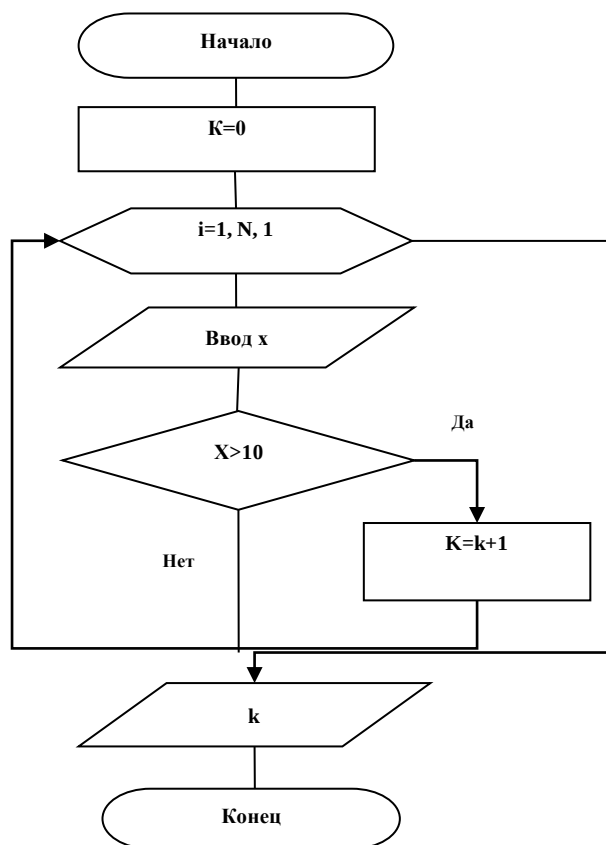
Правило изменения параметра i:	
i=N, K, h	означает
1-й шаг цикла	i=N
2-й шаг цикла	i=N+ h
3-й шаг цикла и т.д.	i=N+2h
Последний шаг цикла	i=K

Задача 1. Вычислить значения функций и вывести на экран результаты в виде таблицы.

¹² Соболев, Б. В. Информатика: учебник / Б.В. Соболев [и др.]. – 3-е, дополн. и перераб. – Ростов н / Д: Феникс, 2007. – 446 [1] с. – (Высшее образование).



Задача 2. Задано 20 чисел. Сколько среди них чисел больших 10?



Псевдокод:

1. $K=0$ {счётчик чисел больших 10}.
2. Повторить 20 раз (для $i=1, 20, 1$).
 - 2.1. Ввод числа x .
 - 2.2. Если $x > 10$, то $K=K+1$.
3. Вывод K .
4. Конец.

Задача 3. Вычислить сумму $S(n) = 1 + 1/2 + 1/3 + \dots + 1/n$.

Подсчёт суммы осуществляется следующим образом:

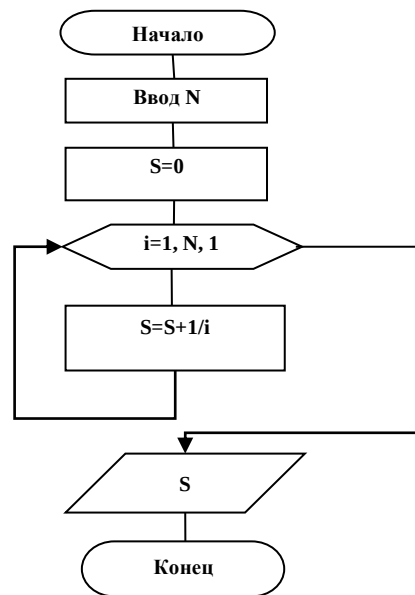
- 1) $S=1$.
- 2) $S=S+1/2$.
- 3) $S=S+1/3$.

Запишем i -й шаг

- i) $S=S+1/i$.

Алгоритм задачи на псевдокоде:

1. Ввод N.
2. $S=0$.
3. Для $i=1, N, 1$ повторять:
 - 1.1. $S=S+1/i$.
2. Вывод S.
3. Конец.



PascalABC.NET

Файл Правка Вид Программа Сервис Модули Помощь

•Program5.pas*

```
program primer_2;
var s:real;
i,n:integer;
begin
writeln('введите n');
readln (n);
s:=0;
for i:=1 to n do begin
s:=s+1/i;
end;|
writeln('s=',s:7:3);
end.
```

Окно вывода

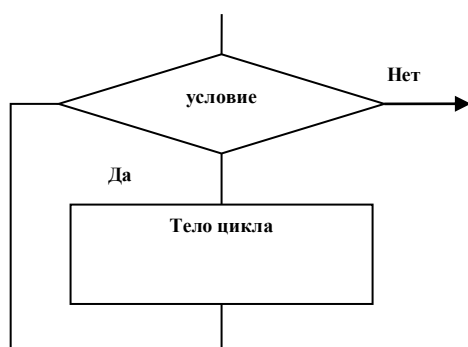
введите n
3
s= 1.833

Окно вывода | Список ошибок | Сообщения компилятора |

Компиляция прошла успешно (12 строк) Строка 10 Столбец 5

Цикл с предусловием

Количество шагов цикла заранее не определено и зависит от входных данных задачи. В данной циклической конструкции сначала повторяется значение условного выражения (условие) перед выполнением очередного шага цикла. Если значение условного выражения истинно, выполняется тело цикла. После чего управление вновь передаётся проверке условия и т.д. Эти действия повторяются до тех пор, пока условное выражение не примет значение ЛОЖЬ. При первом же несоблюдении условия цикл завершается.



Особенностью цикла с предусловием является то, что если изначально условное выражение ложно, то тело цикла не выполнится ни разу.

Задача 4. Вычислить сумму $S(n) = 1 + 1/2 + 1/3 + \dots + 1/n$.

```

program primer_2;
var s:real;
i,n:integer;
begin
  writeln('введите n');
  readln (n);
  s:=0;
  i:=1;
  while i<=n do begin
    s:=s+1/i;
    i:=i+1;
    writeln('s=',s:7:3);
  end;
end.
  
```

Окно вывода

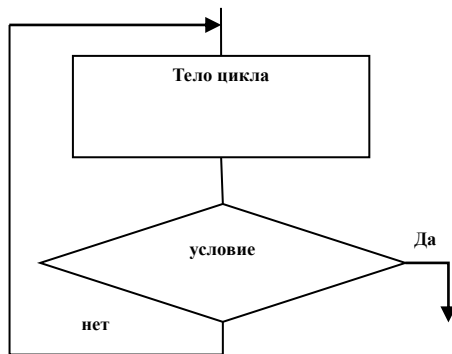
```

введите n
3
s=  1.000
s=  1.500
s=  1.833
  
```

Компиляция прошла успешно (14 строк) Строка 2 Столбец 13

Цикл с постусловием

Как и в цикле с предусловием, в цикле с постусловием заранее не определено число повторений тела цикла, оно зависит от входных данных задачи. В отличие от цикла с предусловием, тело цикла с постусловием всегда будет выполнено хотя бы один раз, после чего проверяется условие. В этой конструкции тело цикла будет выполняться до тех пор, пока значение условного выражения ложно. Как только оно становится истинным, выполнение команды прекращается.



Задача 5. Вычислить сумму $S(n) = 1 + 1/2 + 1/3 + \dots + 1/n$.

Program5.pas

```

program primer_2;
var s:real;
i,n:integer;
begin
writeln('введите n');
readln (n);
s:=0;
i:=1;
repeat
s:=s+1/i;
i:=i+1;
until i>n;
writeln('s=',s:7:3);
end.
  
```

Окно вывода

```

введите n
3
s=  1.833
  
```

Компиляция прошла успешно (14 строк) Строка 14 Столбец 5

Варианты заданий для самостоятельной работы:

Задание 1. Составить программу, используя оператор цикла для вычисления значений функции при изменении аргумента. Вывод на экран вводимых исходных данных и результатов вычислений оформите в виде таблицы с указанием в шапке таблицы имён аргумента и функции¹³.

1.	$y = \begin{cases} \frac{1}{\sin x + 2}, & \text{если } x \leq 0 \\ \lg x + e^x, & \text{если } 0 < x \leq 2 \\ 2x^2, & \text{если } x > 2, \end{cases}$
2.	$y = \begin{cases} e^x + \frac{1}{x+1}, & \text{если } 0 \leq x < 3 \\ \sin x + \sqrt{x}, & \text{если } x = 3 \\ \cos x + b , & \text{если } x \geq 3, \end{cases}$
3.	$Y = \begin{cases} \pi x^2 - 9x^2, & \text{при } x < 1,4 \\ ax^3 + 17\sqrt{x}, & \text{при } x = 1,4 \\ \ln(x + 11\sqrt{ x+a }), & \text{при } x > 1,4, \end{cases}$ где x изменяется в интервале $[0,7 ; 2]$ с шагом 0,1; $a = 1,65$.
4.	$y = \begin{cases} a \lg x + \sqrt[3]{\sin(x)} & \text{при } x > 1 \\ 2a \cos x + e^x & \text{при } x \leq 1, \end{cases}$ где x изменяется в интервале $[0,1 + 1,7]$ с шагом 0,2, $a = 0,9$.
5.	$y = \begin{cases} \sin x \cdot \lg x & \text{при } x > 3,5 \\ \cos^2 x + e^x & \text{при } x \leq 3,5, \end{cases}$ где x изменяется в интервале $[2 + 5]$ с шагом 0,25.

Числовой ряд

¹³ Лабораторный практикум по информатике: Учебное пособие для вузов/ В.С. Микшина, Г.А. Еремеева, Н.Б. Назина и др.; Под ред. В.А. Острейковского. – М.: Высш. шк., 2003. – 376 с.: ил.

Цель: научиться выполнять приближённые вычисления на примере числовых рядов.

Пусть имеется бесконечная последовательность чисел $u_1, u_2, u_3, \dots, u_n$. Числовым рядом называется следующее выражение¹⁴:

$$u_1 + u_2 + u_3 + \dots + u_n = \sum_{i=1}^{\infty} u_n.$$

Числа $u_1, u_2, u_3, \dots$ называются членами ряда, u_n -общим членом ряда.

Примером ряда является сумма членов бесконечно убывающей геометрической прогрессии, т.е. геометрической прогрессии, знаменатель которой меньше единицы $|q| < 1$.

Частичная сумма ряда – сумма n его членов:

$$S_1 = u_1,$$

$$S_1 = u_1,$$

$$S_2 = u_1 + u_2,$$

...

$$S_n = u_1 + u_2 + \dots u_n.$$

Числовой ряд называется *сходящимся*, если существует конечный предел последовательности его частичных сумм: $S = \lim S_n$; S называется *суммой ряда*.

В противном случае (если конечный предел не существует) ряд называется *расходящимся*.

Сумма членов бесконечно убывающей геометрической прогрессии – пример сходящегося числового ряда: $b_{n+1} = b_n q$, общий член ряда равен $b_n = b_1 q^{n-1}$, частичная сумма $S_n = \frac{b_1(1-q^n)}{1-q}$, сумма ряда:

$$\lim_{n \rightarrow \infty} S_n = \sum_{i=1}^{\infty} b_i = \frac{b_1}{1-q}$$

¹⁴ Колдаев, В.Д. Численные методы и программирование: учебное пособие / под ред. проф. Л. Г. Гагариной. – М. : ИД «Форум»: ИНФРА-М, 2008. – 336 с.: ил. – (Профессиональное образование).

Для того, чтобы числовой ряд сходиллся (имел предел конечных сумм), необходимо, чтобы общий член ряда стремился к нулю. В данном случае

$u_n = (-1)^{n+1}$ не стремится к нулю и, следовательно, ряд не сходится.

Это условие является необходимым, но не достаточным. Например, гармонический ряд $\sum_{n=1}^n \frac{1}{n}$ расходится, несмотря на то, что его общий член стремится к нулю. Для решения вопроса о сходимости служат признаки сходимости числовых рядов.

Если ряд сходится, то разность между суммой S и частичной суммой S_n его называется n -м *остатком ряда*. Остаток ряда представляет собой погрешность, которая получится, если в качестве приближённого значения суммы ряда S взять S_n – сумму первых n членов этого ряда.

Таким образом, взяв достаточно большое число членов сходящегося ряда, можно вычислить сумму этого ряда с любой степенью точности.

Для решения вопроса о том, сходится ряд или нет, служат признаки.

Признак Даламбера: Пусть ряд $\sum u_n$ имеет положительные члены. Предположим, что последовательность отношений u_{n+1}/u_n сходится к некоторому числу p при $n \rightarrow \infty$. Тогда, если $p < 1$, то данный ряд сходится, если $p > 1$, то ряд расходится, если $p = 1$, то признак ответа не даёт – возможна как сходимость, так и расходимость ряда.

Признак Коши: Пусть ряд $\sum u_n$ имеет положительные члены. Предположим, что последовательность $\sqrt[n]{u_n}$ сходится к некоторому числу p при $n \rightarrow \infty$. Тогда, если $p < 1$ то данный ряд сходится, если $p > 1$, то ряд расходится, если $p = 1$, то признак ответа не даёт – возможна как сходимость, так и расходимость ряда.

Интегральный признак: Если $f(x)$ при $x > 1$ является непрерывной, положительной и убывающей функцией, то ряд $\sum_{n=1}^{\infty} u_n$, где $u_n = f(n)$, сходится или расходится в зависимости от того, сходится или расходится интеграл $\int_1^{\infty} f(x)dx$.

Признак сходимости знакопеременных рядов: Пусть в ряду u_n есть и положительные и отрицательные члены. Если сходится ряд, составленный из модулей $\sum |u_n|$, то ряд $\sum u_n$ также сходится. Обратное утверждение неверно.

Признак (Лейбница) для знакочередующихся рядов (u_n являются поочерёдно положительными и отрицательными). Если члены ряда убывают по модулю, т.е. $|u_n| > |u_{n+1}|$, а также $|u_n| \rightarrow 0$, то такой ряд сходится.

Упражнение

Исследуйте сходимость ряда

$$1 + (2/3)^2 + (3/5)^3 + (4/7)^2 + \dots = \sum_{n=1}^{\infty} \left(\frac{n}{2n+1} \right)^n.$$

План исследования ряда:

1. Проверка сходимости с помощью одного из признаков.
2. Нахождение суммы ряда, если он сходится.

Сходимость ряда будет проявляться в том, что с увеличением n будет повторяться всё большее количество одних и тех же (правильных) цифр суммы. Чем меньше требуется число n для вычисления суммы ряда с заданной точностью, тем быстрее сходится ряд. Если ряд расходится, то это будет выражаться в том, что значения частичных сумм при увеличении n будут увеличиваться и всё более и более разниться друг от друга.

Визуальное определение того, что ряд сходится, для медленно сходящихся рядов может потребовать большого количества машинного времени. Более того, если в программе не предусмотрен выход из программы по количеству слагаемых в частичной сумме, то для расходящегося ряда программа, предназначенная для вычисления суммы сходящегося ряда, может заикнуться (компьютер повиснет).

Пример расходящегося ряда – гармонический ряд. Обобщённый гармонический ряд: $\sum_{n=1}^{\infty} \frac{1}{n^\alpha}$ сходится при $\alpha > 1$ и расходится при $0 \leq \alpha \leq 1$.

Задача 1. Вычислить сумму обобщённого гармонического ряда, когда он сходится. Кроме того, можно увидеть, как ведут себя частичные суммы при увеличении n для расходящегося ряда и для медленно сходящихся рядов.

```

program summa_ryada;
const epsmin=1e-8; xmax=1000000; alfa=3.0;
type
func=function(x:real):real;
var s,epsilon:real;
{общий член гармонического ряда}
function f1(x:real):real;
begin
  f1:=1/(exp(alfa*ln(x)));
end;
{вычисление суммы}
function lims(f:func;epsilon:real):real;
var x,s:real;
begin
  s:=0; x:=1;
  repeat
    s:=s+f1(x);
    x:=x+1;
  until (abs(f1(x))<epsilon) or (x>xmax);
  if x<=xmax then lims:=s
  else
    begin lims:=77777;
      writeln ('число слагаемых слишком велико');
      writeln('ряд расходится (или сходится очень медленно)');
    end;
  end;
  begin
    epsilon:=0.1;
    writeln('Сумма обобщённого гармонического ряда,
alfa=',alfa:4:2);
  repeat
    s:=lims(f1,epsilon);
    if s<>77777 then
      writeln('epsilon=',epsilon:10:8,' s=',s:8:5);
    epsilon:=epsilon*0.1;
  until epsilon<epsmin;
  writeln('вычисления закончены');
end.

```

PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

Сумма обобщённого гармонического ряда

```
program summa_ryada;
const epsmin=1e-8; xmax=1000000; alfa=3.0;
type
func=function(x:real):real;
var s,epsilon:real;
{общий член гармонического ряда}
function f1(x:real):real;
begin
    f1:=1/(exp(alfa*ln(x)));
end;
{вычисление суммы}
function lims(f:func;epsilon:real):real;
var x,s:real;
```

Окно вывода

Сумма обобщённого гармонического ряда, alfa=3.00

epsilon=0.10000000	s= 1.12500
epsilon=0.01000000	s= 1.17766
epsilon=0.00100000	s= 1.19653
epsilon=0.00010000	s= 1.20098
epsilon=0.00001000	s= 1.20183
epsilon=0.00000100	s= 1.20201
epsilon=0.00000010	s= 1.20205
epsilon=0.00000001	s= 1.20205

вычисления закончены

Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (37 строк) | Строка 25 Столбец 29

Для $\alpha = 1,5$

PascalABC.NET

Файл Плавка Вид Программа Сервис Модули Помощь

ма обобщённого гармонического ряда

```
program summa_ryada;
const epsmin=1e-8; xmax=1000000; alfa=1.5;
type
func=function(x:real):real;
var s,epsilon:real;
{общий член гармонического ряда}
function f1(x:real):real;
begin
    f1:=1/(exp(alfa*ln(x)));
end;
{вычисление суммы}
```

Окно вывода

Сумма обобщённого гармонического ряда, alfa=1.50

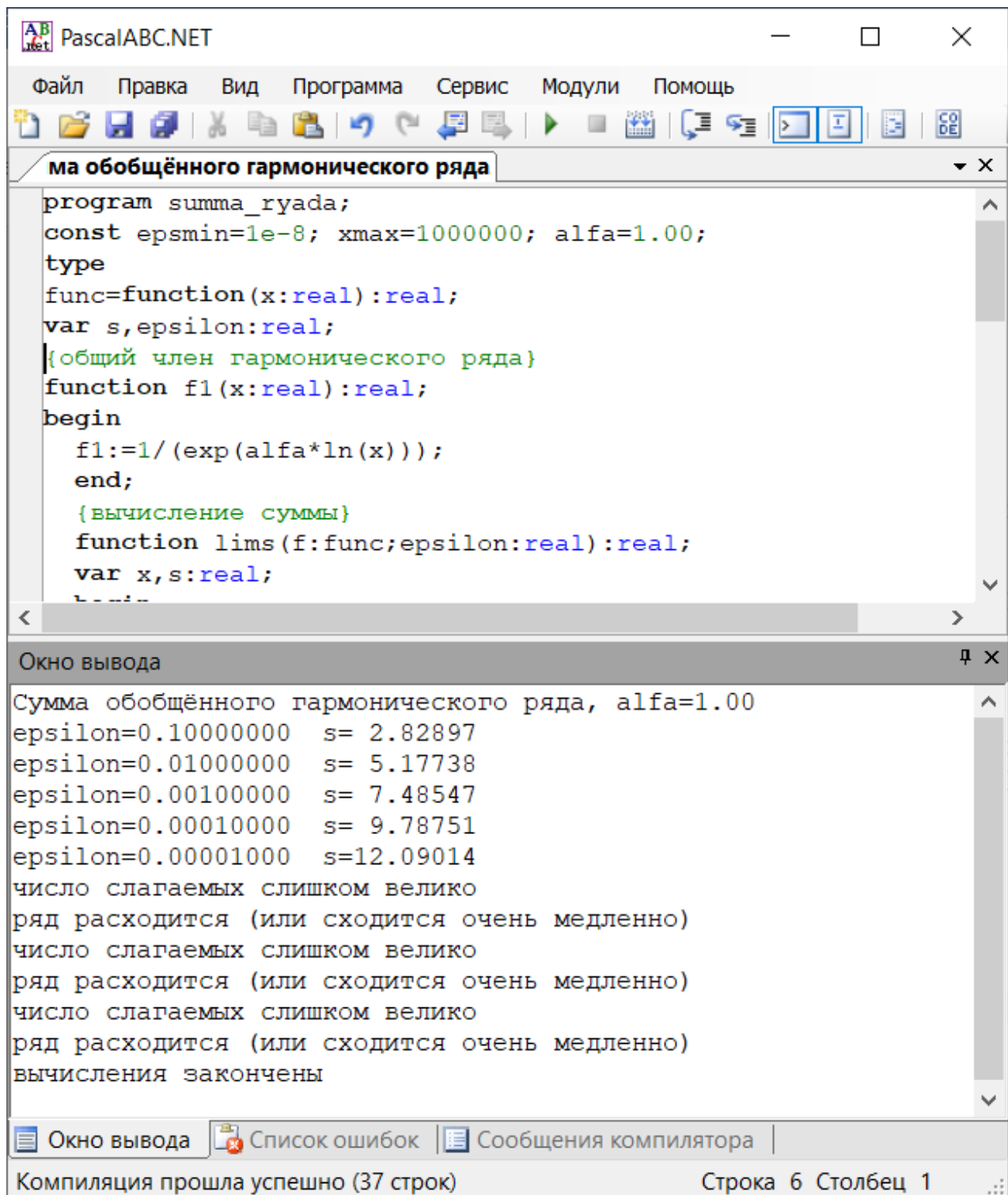
epsilon=0.10000000	s= 1.67100
epsilon=0.01000000	s= 2.18107
epsilon=0.00100000	s= 2.41187
epsilon=0.00010000	s= 2.51958
epsilon=0.00001000	s= 2.56929
epsilon=0.00000100	s= 2.59237
epsilon=0.00000010	s= 2.60309
epsilon=0.00000001	s= 2.60807

вычисления закончены

Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (37 строк) Строка 4 Столбец 28

Для $\alpha = 1$



Задание для самостоятельной работы:

1. Измените программы так, чтобы параметр α мог вводиться.
2. Выполните программу при других значениях этого параметра.
3. Дополните программу определение того, что ряд расходится.

Задача 2. Вычислить сумму ряда $y = \frac{x}{2} - \frac{x^2}{2^2} + \frac{x^3}{2^3} - \frac{x^4}{2^4} + \dots$ с точностью $\varepsilon = 0,001$ при заданном значении¹⁵ $x = 0,235$.

Вычисление суммы ряда будем производить до тех пор, пока очередной член ряда по абсолютной величине не будет меньше $\varepsilon = 0,001$.

Обозначим n -ый член ряда через $U = (-1)^{n+1} \frac{x^n}{2^n}$. Тогда сумма находится следующим образом: $y = y + U$. Определим следующий член ряда через предыдущий: $U_{n+1} = -U_n \frac{x}{2}$, при этом $U_0 = -1$, $U_1 = \frac{x}{2}$.

```

Program summa;
  Const eps = 0.001;
  Var u, x, y : real; n: integer;
Begin
  x:= 0.235; y:= 0; n:= 0; u:= - 1;
repeat n:= n+1;
  u:= - u * x / 2;
  y:= y+u;
until abs (u)< = eps;
  writeln ('y=', y:9:4);
end.

```

Варианты заданий для самостоятельной работы:

¹⁵ Колдаев В.Д. Численные методы и программирование: учебное пособие/ Под ред. Проф. Л.Г. Гагариной. -М.: ИД «ФОРУМ»: ИНФРА-М, 2008.- 336 с.: ил.- (Профессиональное образование).

Задание 1. Вычислить сумму ряда. Сравнить полученное на ЭВМ значение суммы ряда со значением, вычисленным аналитически.

1.	Найти сумму ряда: $S = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \dots + \frac{1}{n^2}$ с заданной точностью $\epsilon = 10^{-3}$.
2.	Найти сумму 35 членов ряда: $S = \frac{2}{3} + \frac{4}{9} + \frac{6}{27} + \frac{8}{81} + \dots$
3.	Найти сумму ряда: $S = \frac{1}{2^2} + \frac{3}{4^2} + \frac{5}{6^2} + \frac{7}{8^2} + \frac{9}{10^2} \dots$ с заданной точностью $\epsilon = 10^{-3}$.
4.	Найти сумму ряда: $S = \frac{1}{1+1!} + \frac{2}{1+2!} + \frac{3}{1+3!} + \dots$ с заданной точностью $\epsilon = 10^{-4}$.
5.	Найти сумму ряда, начиная отсчет с нулевого члена: $S = 1 + \frac{x^2}{2!} - \frac{3x^4}{4!} + \dots + (-1)^n \frac{2n-1}{(2n)!} x^{2n}$ при $x = 0,75$ с заданной точностью $\epsilon = 10^{-3}$.

Требования к оформлению отчёта по практической работе

Отчёт должен содержать следующую информацию:

1. Название практической работы.
2. Содержание задания вашего варианта.
3. Описание структуры данных: входные и выходные данные, их тип, диапазон допустимых в данной задаче значений.
4. Блок-схема алгоритма решения задачи.
5. Код программы на языке программирования.
6. Контрольный пример.

Порядок защиты практической работы

Порядок выполнения практических работ осуществляется в соответствии с индивидуальным заданием, согласно номеру варианта,

который определяет преподаватель. Защита практической работы осуществляется в устной форме по контрольным вопросам.

Подпрограммы

Цель: научиться оформлять процедуры и функции.

Если алгоритм решения задачи содержит фрагменты, которые могут быть использованы не один раз в нескольких местах программы, то такие фрагменты можно выделить в программные единицы (процедуры и функции). Обращение к однажды написанному фрагменту программы с заданием новых входных данных (параметров программной единицы) позволяет существенно сократить общий объём программы.

Структура процедуры повторяет структуру программы. Отличие состоит в том, что у процедуры есть параметры.

Оформление процедур

Procedure name_proc[(list_arg)]; [directives;]

Здесь:

Procedure – служебное слово;

name_proc – имя процедуры;

list_arg – необязательный список параметров;

directives – одна или несколько уточняющих директив.

Пример 1. Написать программу перестановки значений переменных a, b, c в порядке возрастания, т.е. так, чтобы $a < b < c$.

The screenshot shows the PascalABC.NET IDE with a menu bar (Файл, Правка, Вид, Программа, Сервис, Модули, Помощь) and a toolbar. Two tabs are open: 'Program5.pas*' and 'Program6.pas*'. The active tab 'Program6.pas*' contains the following Pascal code:

```
Var a, b, c: integer;
Procedure swap (var x, y: integer);
Var t: integer;
Begin
t:=x;
x:=y;
y:=t;
end;
begin
writeln ('введите три числа');
readln (a, b, c);
if a>b then swap (a, b);
if b>c then swap (b, c);
if a>c then swap (a, c);
writeln (a:5, b:5, c:5);
end.
```

Below the code editor is the 'Окно вывода' (Output Window) which displays the program's execution:

```
введите три числа
3
9
7
      3      7      9
```

At the bottom of the IDE, a status bar indicates 'Компиляция прошла успешно (17 строк)' (Compilation successful (17 lines)) and the current cursor position 'Строка 11 Столбец 29' (Line 11, Column 29).

Пример 2. Перемножение квадратных матриц.

Const n=10;

Type

matN = array [1..N,1..N] of double;

var

a, b, c : matN;

procedure mat_mult (A,B: matN; var C:mat_N);

var i, j, k: byte; s: double;

begin

for i:=1 to N do

for j:= 1 to N do

begin

s:=0;

for k:=1 to N do

```

s:=s+A[i,k]*B[k,j];
c[i,j]:=s;
end;
end;

```

Оформление функций

Функция представляет собой частный вид процедуры, результатом работы которой является *единственное значение*. Его принято называть *значением, которое возвращает функция*. Такой результат позволяет использовать функцию в качестве операнда любой формулы соответствующего типа:

```
Y:=f1(x,z)*sin(f2(u))+f3(v);
```

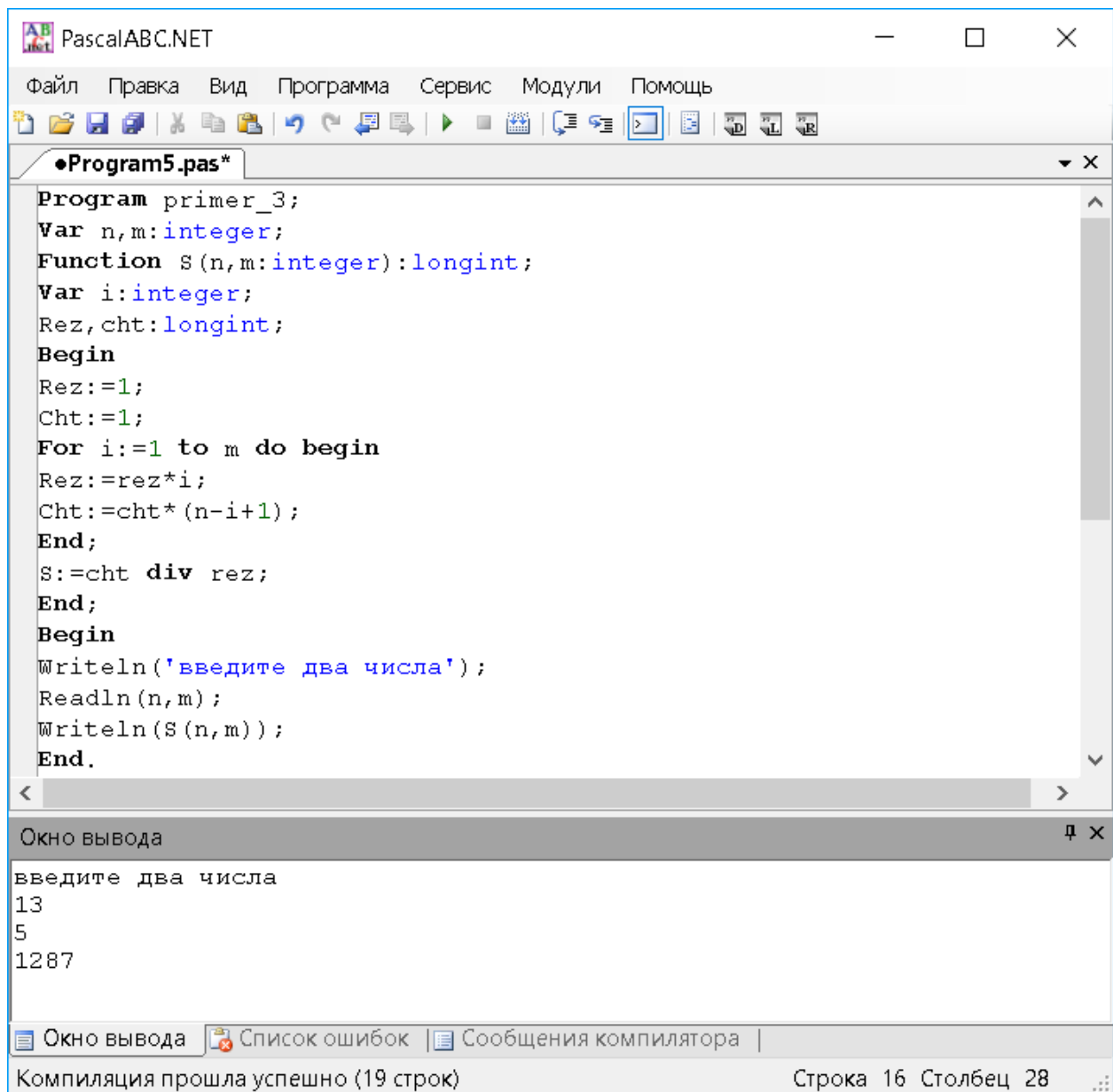
Объявление функции начинается с заголовка, который в общем случае имеет вид:

```
Function name_fun [(list_arg)]:tip; [directives];
```

tip – тип возвращаемого значения.

Пример 3. Составить программу подсчёта числа сочетаний $C(n, m)$.
Написание основной программы сводится к программированию формулы:

$$C(n, m) = \frac{n!}{m! (n - m)!}$$



Пример 4. Вычисление скалярного произведения.

Const N=10;

Type

vecN = array [1..N] of double;

var

a, b : vecN;

c:double;

function vec_mult (A, B:vecN): double;

var i:byte;

```

s: double;
begin
s:=0;
for i:=1 to N do
s:= s+A[i]*B[i];
vec_mul:=s;
end;

```

Практическая работа №11. Одномерные массивы. Двумерные массивы.

Цель: выработка навыков обработки элементов одномерных массивов.

Тип данных, позволяющий хранить вместе под одним именем несколько переменных, называется структурированным. Каждый язык программирования имеет свои структурированные типы¹⁶.

Массивом называется упорядоченная совокупность однотипных величин, имеющих общее имя, элементы которой адресуются (различаются) порядковыми номерами (индексами).

Одномерный массив предполагает наличие у каждого элемента только одного индекса. Примерами одномерных массивов служат арифметическая a_i и геометрическая b_i последовательности, определяющие конечные ряды чисел. Количество элементов массива называют размерностью.

Формат:

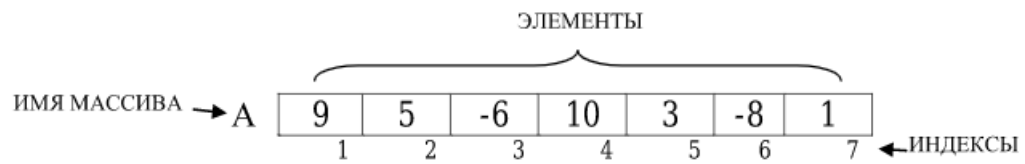
<Имя>: **Array** [Тип индексов] **of** <Тип элементов>;

Пример:

Type MyArray=**Array**[1..20] **of** integer;

Каждый элемент массива однозначно определяется именем массива и индексом (номером этого элемента в массиве)

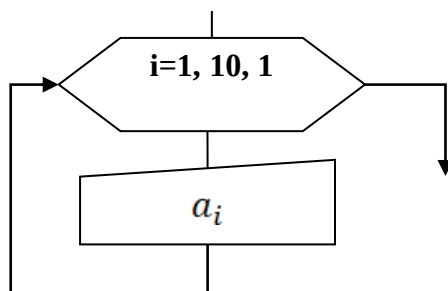
¹⁶ Соболев, Б. В. Информатика: учебник / Б.В. Соболев [и др.]. – 3-е, дополн. и перераб. – Ростов н / Д: Феникс, 2007. – 446 [1] с. – (Высшее образование).



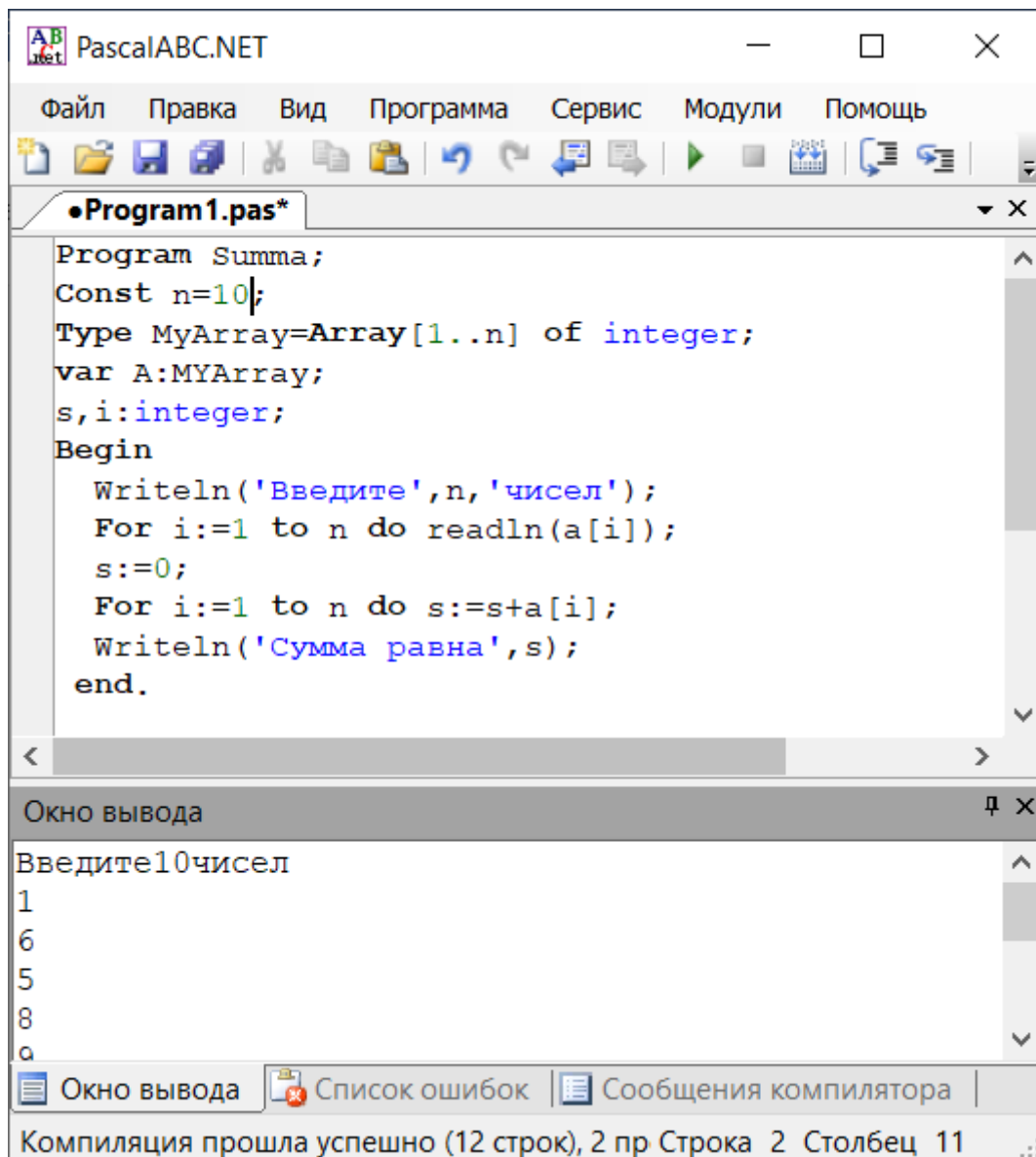
Индексы элементов массива должны принадлежать порядковому типу.

Наиболее часто типом индекса является целочисленный тип.

Ввод элементов одномерного массива:



Задача 1. Найти сумму элементов массива.



Задача 2. Используя задачу 1, найдём сумму элементов массива, кратных заданному числу.

Предыдущее решение изменится незначительным образом. Добавляется описание ещё одной переменной для хранения значения числа, на кратность которому проверяются значения элементов массива. Появляются операторы:

Writeln ('Введите число');

Readln (k);

и изменяется оператор из тела цикла:

if a[i] mod k=0 then s:=s+a[i];

Задача 3. Измените программу так, чтобы определялось количество положительных и отрицательных элементов в данном массиве¹⁷.

Суть основного изменения программы, заключается во введении двух переменных (счётчиков pos, neg) для хранения значений количества положительных и отрицательных элементов в массиве.

Найдите место в программе для следующих строк программного кода:

```
Pos, neg:integer;
```

```
Pos:=0;
```

```
Neg:=0;
```

```
If a[i]>0 then pos:=pos+1 else if a[i]<0 then neg:=neg+1;
```

```
Writeln(pos:4,neg:4);
```

Данная программа не решает задачу подсчёта количества нулевых элементов в массиве. Модифицируйте программу для решения этой задачи.

Задача 4. Измените программу так, чтобы номера чётных элементов массива А записывались в массив В.

Введение массива В и работа с ним требуют введения переменной j для обращения к элементам В. Суть решения заключается в просмотре элементов массива А, выявлении чётных элементов и записи их номеров по текущему значению переменной j в массив В.

Мы записываем не значения элементов, а их номера.

Результат работы программы:

¹⁷ Окулов С.М. Основы программирования.- М.: БИНОМ. Лаборатория знаний, 2015. -336 с.: ил. – (Развитие интеллекта школьников).

The screenshot shows the PascalABC.NET IDE. The main window displays a Pascal program named 'Program Summa'. The program defines a constant n=10, a type MyArray as an array of integers from 1 to n, and two variables A and B of type MyArray. It prompts the user to enter 10 numbers, reads them into array A, and then iterates through A to find even numbers, storing their indices in array B. Finally, it prints the indices stored in B. The output window shows the input sequence: 1 3 5 7 2 3 4 3 6 4, followed by the output sequence: 5 7 9 10. The status bar at the bottom indicates a successful compilation with 3 warnings.

```
Program Summa;  
Const n=10;  
Type MyArray=Array[1..n] of integer;  
var A,B:MYArray;  
i,j:integer;  
Begin  
  Writeln('Введите',n,'чисел');  
  For i:=1 to n do read(a[i]);  
  j:=0;  
  For i:=1 to n do if a[i] mod 2=0 then begin  
    j:=j+1;  
    b[j]:=i;  
  end;  
  for i:=1 to j do write (B[i]:3);  
  writeln;  
end.
```

Введите10чисел
1 3 5 7 2 3 4 3 6 4
5 7 9 10

Компиляция прошла успешно (16 строк), 3 предупрежд Строка 8 Столбец 24

Задача 5. Определить, есть ли в массиве элемент с заданными свойствами, например, есть ли отрицательный элемент. Уточним условие – ищем номер последнего отрицательного элемента.

Решение:

i:=n;

While (i>=1) and (a[i]>0) do i:=i-1;

If i<1 then <отрицательных элементов в массиве нет>

Else <последний отрицательный элемент имеет номер i>;

Задача 6. Найдём значения и номера всех отрицательных элементов.

The screenshot shows the PascalABC.NET IDE. The main window displays a Pascal program named 'Program1.pas*'. The code defines an array 'MyArray' of integers and reads 10 elements. It then prints the negative elements and their indices. The output window shows the input sequence '2 4 -3 2 -5 1 -9 3 1 5' and the resulting output for negative elements: '-3 3', '-5 5', and '-9 7'. The status bar at the bottom indicates 'Компиляция прошла успешно (11 строк), 2 предупреждений' and 'Строка 9 Столбец 77'.

```
Program Summa;  
Const n=10;  
Type MyArray=Array[1..n] of integer;  
var A:MYArray;  
i:integer;  
Begin  
  Writeln('Введите элементы массива, в том числе и отрицательных!');  
  For i:=1 to n do read(a[i]);  
  writeln ('Вывод отрицательных элементов массива и их номеров (индексов)');  
  For i:=1 to n do if a[i]<0 then writeln(a[i]:5,i:5);  
end.
```

Окно вывода

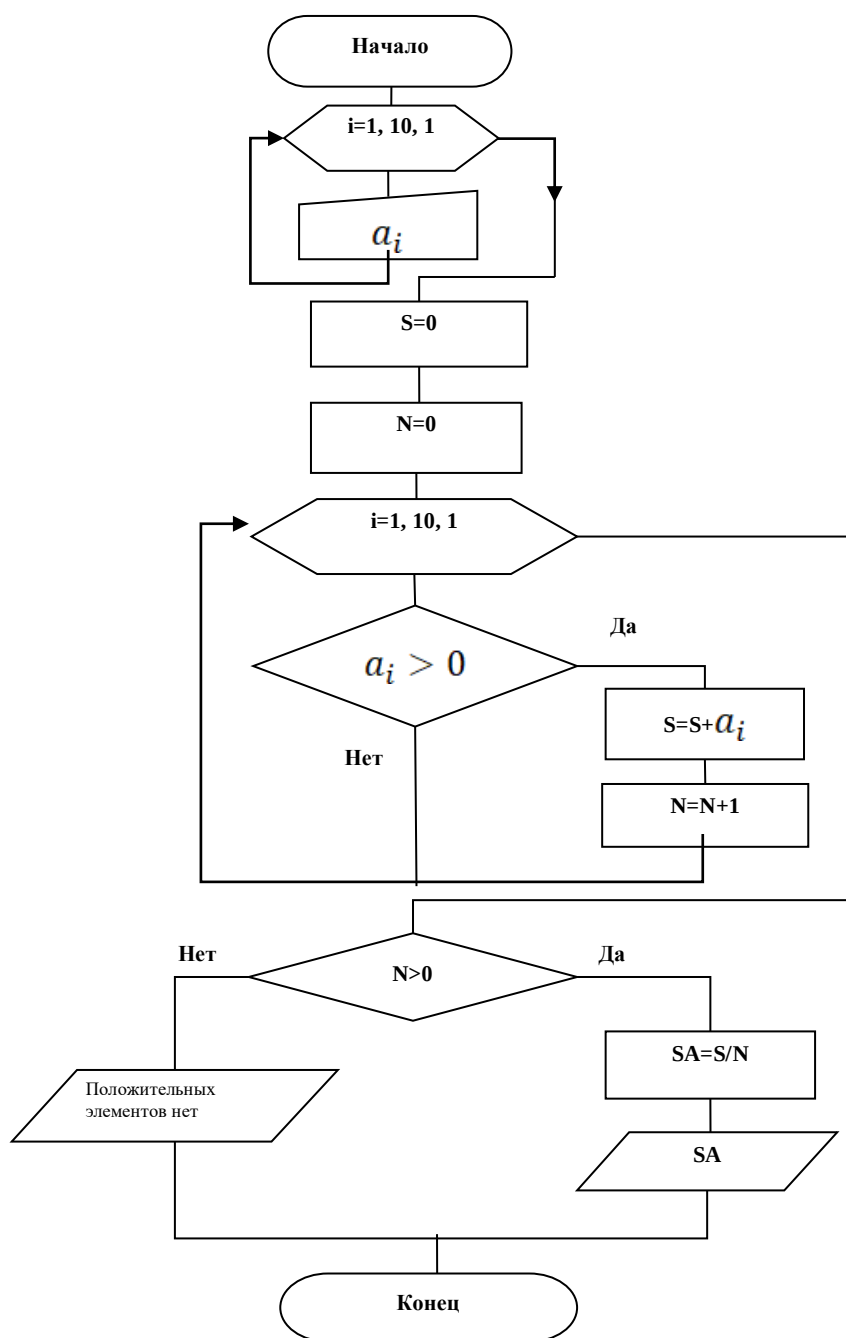
Введите элементы массива, в том числе и отрицательных!
2 4 -3 2 -5 1 -9 3 1 5
Вывод отрицательных элементов массива и их номеров (индексов)
-3 3
-5 5
-9 7

Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (11 строк), 2 предупреждений

Строка 9 Столбец 77

Задача 7. Вычислить среднее арифметическое положительных элементов массива, используя данную блок-схему.



Функция *Random* (генератор случайных чисел)

Описание: `Random[(range:Word)]`. В квадратных скобках указывается необязательный параметр конструкции языка Паскаль.

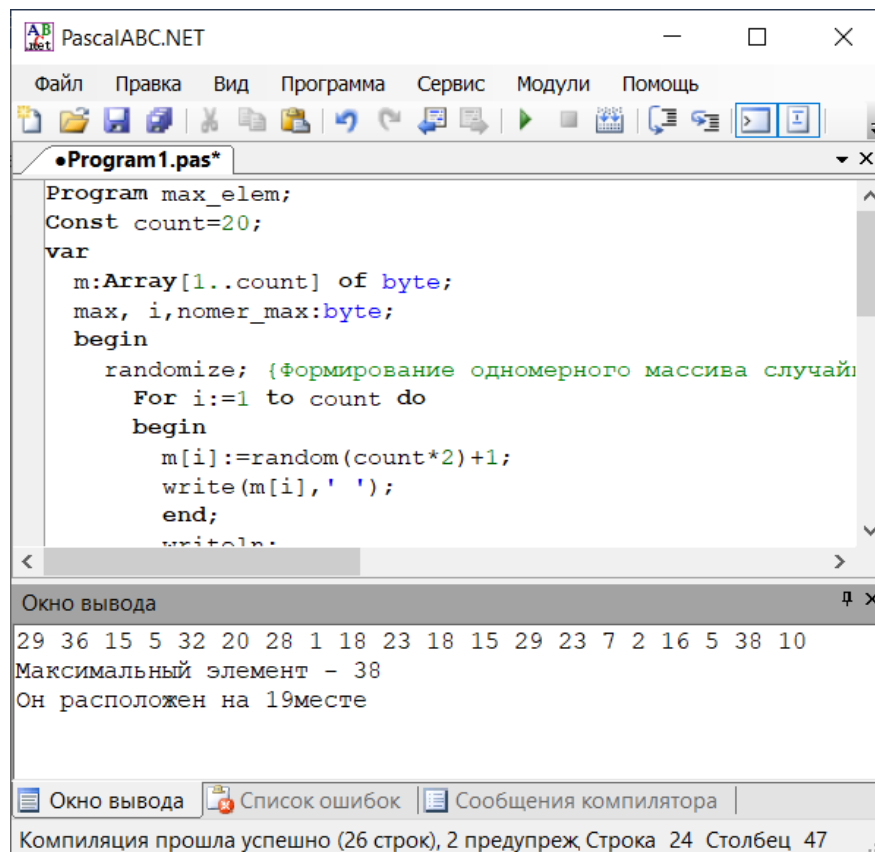
Тип результата: `Real` или `Word` в зависимости от наличия параметра.

Если параметр не задан, то результатом является число типа `Real` в диапазоне $0 \leq x < 1$. При наличии параметра возвращается число типа `Word` в диапазоне $0 \leq x < \text{range}$. Обратите внимание на то, что верхняя граница диапазона не достигается, - неравенство строгое.

Задача 8. Составить программу, которая формирует одномерный массив случайных чисел, выполняет поиск максимального элемента массива, а затем выводит на экран его значение и порядковый номер в массиве.

```
Program max_elem;
Const count=20;
var
  m:Array[1..count] of byte;
  max, i,nomer_max:byte;
begin
  randomize; {Формирование одномерного массива случайной функцией
Random(count*2) и вывод массива на экран}
  For i:=1 to count do
  begin
    m[i]:=random(count*2)+1;
    write(m[i], ' ');
  end;
  writeln;
  max:=m[1]; {Считать первый элемент максимальным}
  nomer_max:=1; {Запомнить номер максимального элемента}
  for i:=2 to count do { Проверить все элементы, начиная со второго}
  begin
    if m[i]>max then
    begin
      max:=m[i];
      nomer_max:=i;
    end;
  end;
  writeln('Максимальный элемент - ',max);
  writeln('Он расположен на ', nomer_max,'месте');
end.
```

Результат работы программы:



Сортировка массивов

Сортировка- один из наиболее распространённых процессов современной обработки данных. **Сортировкой** называется распределение элементов множества по группам в соответствии с определёнными правилами.

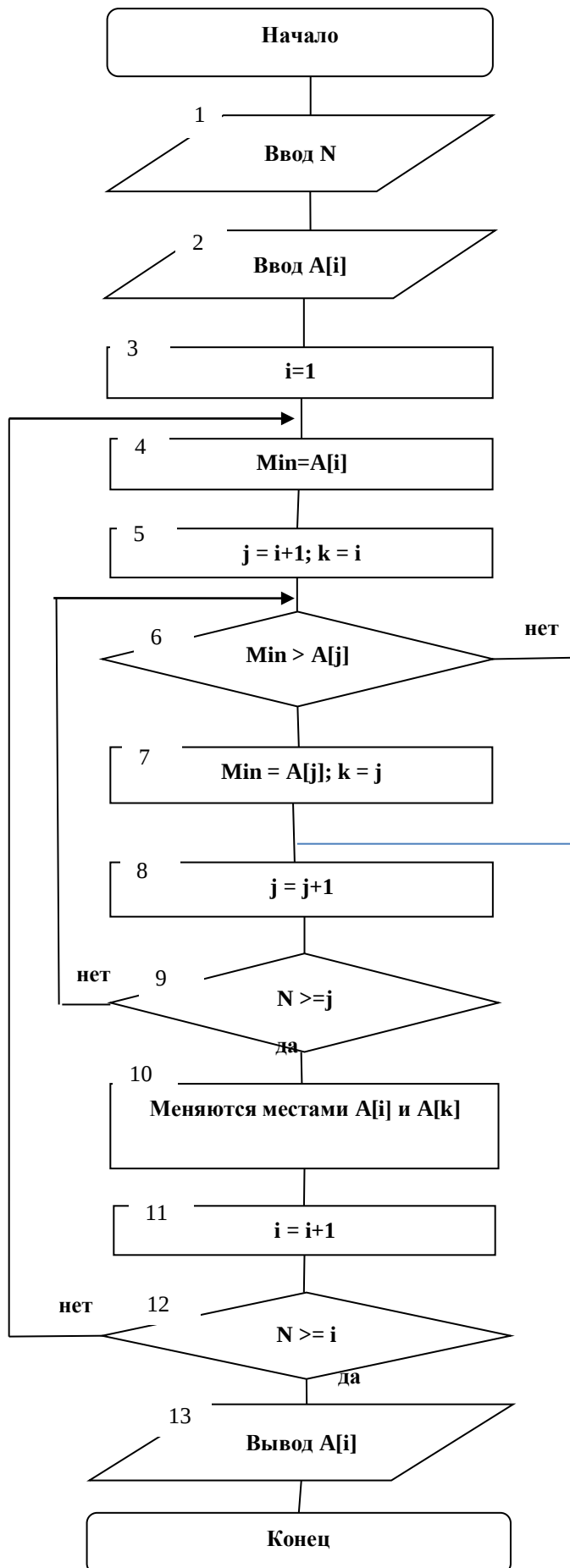
Пример 2. Выполнить сортировку по возрастанию N целых чисел.

Метод сортировки посредством простого выбора предполагает циклический просмотр элементов массива, начиная с i -го ($i=1, 2, \dots, N-1$), поиск минимального элемента и перестановку найденного минимального элемента с i -м. За $N-1$ проход по массиву (элемент с номером N останется на своём месте) числа будут отсортированы.

<table><tr><td>2</td><td>11</td><td>3</td><td>1</td><td>4</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	2	11	3	1	4	1	2	3	4	5	1-й проход	i=1 1-й элемент меняется местами с 4-м
2	11	3	1	4								
1	2	3	4	5								
<table><tr><td>1</td><td>11</td><td>3</td><td>2</td><td>4</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	11	3	2	4	1	2	3	4	5	2-й проход	i=2 2-й элемент меняется местами с 4-м
1	11	3	2	4								
1	2	3	4	5								

<table><tr><td>1</td><td>2</td><td>3</td><td>11</td><td>4</td></tr></table> 1 2 3 4 5	1	2	3	11	4	3-й проход i=3 3-й элемент не меняет места
1	2	3	11	4		
<table><tr><td>1</td><td>2</td><td>3</td><td>11</td><td>4</td></tr></table> 1 2 3 4 5	1	2	3	11	4	i=4 4-й элемент меняется с 5-м
1	2	3	11	4		
<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>11</td></tr></table>	1	2	3	4	11	Отсортированный массив
1	2	3	4	11		

Цикл обработки массива представляет собой совокупность двух «вложенных циклов»: первый цикл (с переменной цикла i) обеспечивает «внешний» проход по элементам массива, начиная с 1-го и кончая элементом с номером $N-1$, который должен заканчиваться на каждом шаге перестановкой элементов; второй цикл (с переменной цикла j) реализует поиск минимального элемента среди элементов с номерами от $i+1$ до N (т.е. j меняется местами от $i+1$ до N).



The screenshot shows the PascalABC.NET IDE with a program named 'Program6.pas'. The code implements a selection sort algorithm. It prompts the user to enter the number of elements in the array (5), then enters each element (2, 6, 8, 9, 4). The program then outputs the sorted array elements (2, 4, 6, 8, 9).

```

Program sort_vibor;
Var A: array [1..20] of integer; i, j, min, I_min, N : integer;
Begin
  Writeln ('Введите количество элементов массива');
  Readln (N);
  For i:= 1 to N do begin // Ввод элементов массива
    Write ('Введите элемент массива №', i, ':');
    Readln (A [i] );
  end;
  For i := 1 to N do // Внешний цикл прохода по элементам
    Begin I_min := I; // Начальное задание индекса минимального элемента
      For j :=i+1 to N do // Внутренний цикл поиска минимального элемента
        // в пределах от i+1 до N
        If A[j] < A[I_min] then I_min := j;
      Min := A[I_min]; // Сохранение минимального значения
      // Перестановка элементов: I -го и минимального
      A[I_min]:= A[i];
      A[i]:= min;
    End;
  For i:= 1 to N do // Вывод отсортированного массива
    Writeln ('Элемент массива №', I, '=', A[i]);
End.

```

The output window shows the following text:

```

Введите количество элементов массива
5
Введите элемент массива №1:2
Введите элемент массива №2:6
Введите элемент массива №3:8
Введите элемент массива №4:9
Введите элемент массива №5:4
Элемент массива №1=2
Элемент массива №2=4
Элемент массива №3=6
Элемент массива №4=8
Элемент массива №5=9

```

The status bar at the bottom indicates: "Компиляция прошла успешно (22 строк) Строка 6 Столбец 24".

Варианты заданий для самостоятельной работы:

Дан массив целых чисел. Найдите:

1.	Сумму элементов массива, больших данного числа а(а вводить с клавиатуры).
2.	Сумму элементов массива, принадлежащих промежутку от а до b (а и b вводить с клавиатуры).
3.	Номера всех элементов массива с максимальным значением.

4.	Все элементы кратные 3 или 5.
5.	Сумму всех чётных элементов массива, стоящих на чётных местах, т.е. имеющих чётные номера.
6.	Сумму элементов, имеющих нечётные значения.
7.	Сумму элементов, имеющих нечётные индексы.
8.	Индексы тех элементов, значения которых больше заданного числа а.
9.	Количество элементов массива, значения которых больше заданного числа а и кратны 5.
10.	Индексы тех элементов массива, которые кратны 3 и 5.

Контрольные вопросы:

1. Что такое массив?
2. Как определить местоположение элемента в массиве?
3. Что такое индекс? Каким требованиям он должен удовлетворять?
4. Каким образом задаётся описание массива, что в нём указывается?
5. Что называется сортировкой массива? Какие методы сортировки вы знаете? Опишите их существенные отличия.

Требования к оформлению отчёта по практической работе

Отчёт должен содержать следующую информацию:

1. Название практической работы.
2. Содержание задания вашего варианта.
3. Описание структуры данных: входные и выходные данные, их тип, диапазон допустимых в данной задаче значений.
4. Код программы на языке программирования.
5. Контрольный пример.

Порядок защиты практической работы

Порядок выполнения практических работ осуществляется в соответствии с индивидуальным заданием, согласно номеру варианта, который определяет преподаватель. Защита практической работы осуществляется в устной форме по контрольным вопросам.

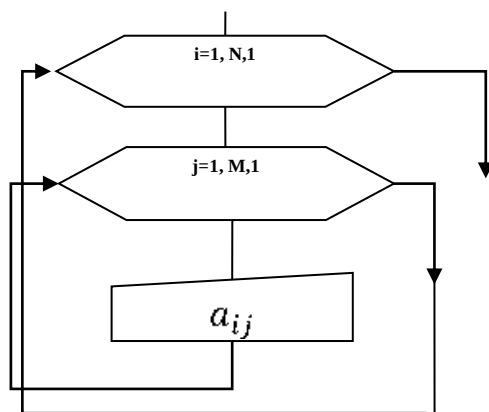
Двумерные массивы

Рассмотрим алгоритм обработки двумерного массива. Аналогом двумерного массива в математике является матрица размерности (M×N) [доп. 4]:

$$A = \{a_{i,j}, i = \overline{1..M}, j = \overline{1..N}\} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{pmatrix}$$

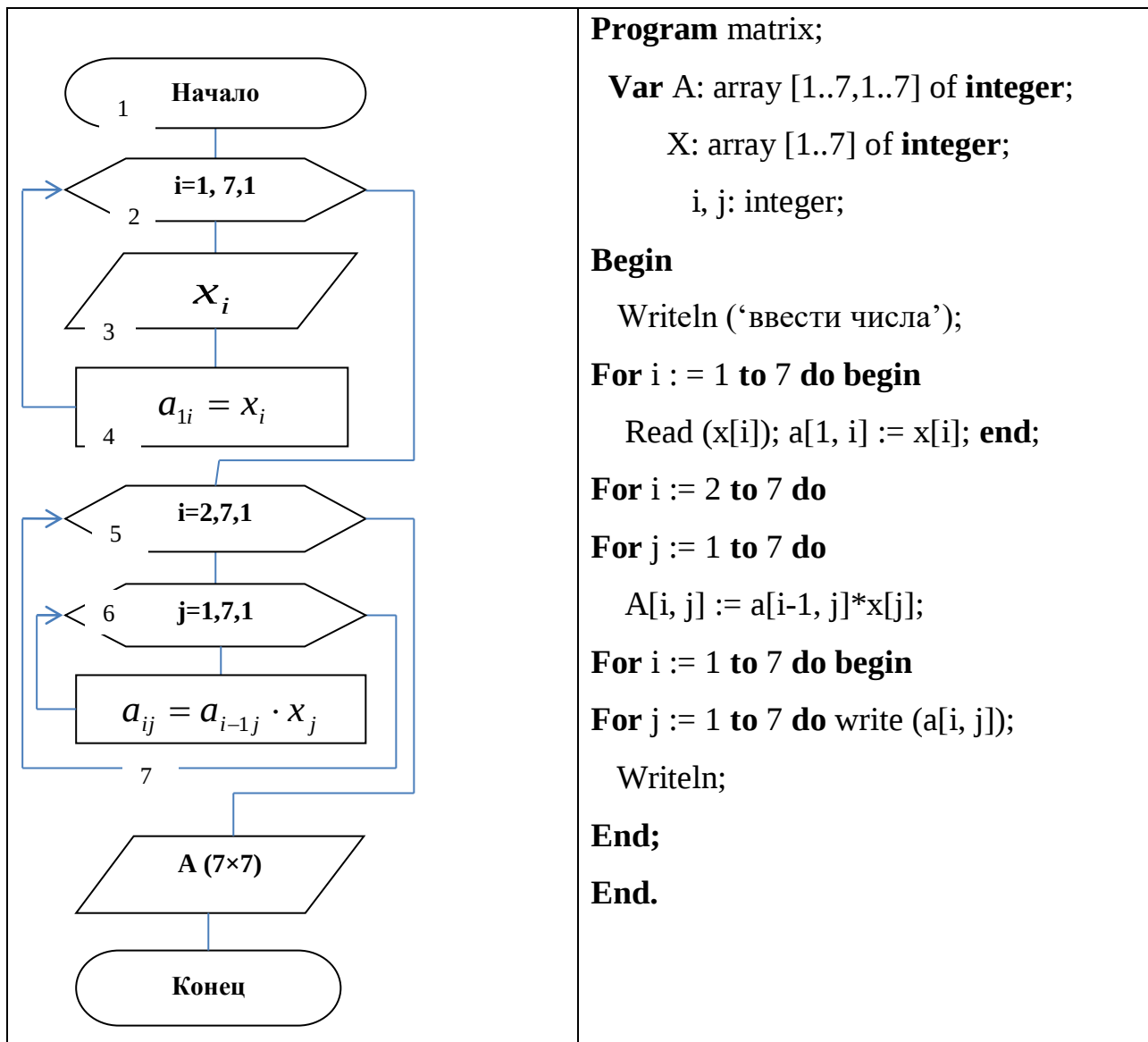
Для матрицы размерностью N×N (квадратной матрицы) определены понятия главной и побочной диагоналей. Элементы главной диагонали - $\{a_{ii}, i = \overline{1..N}\}$, а элементы побочной диагонали - $\{a_{i(N-i+1)}, i = \overline{1..N}\}$.

Ввод элементов двумерного массива осуществляется построчно, в свою очередь, ввод каждой строки производится поэлементно, тем самым определяется циклическая конструкция, реализующая вложение циклов. Внешний цикл определяет номер вводимой строки i, внутренний – номер элемента по столбцу j.



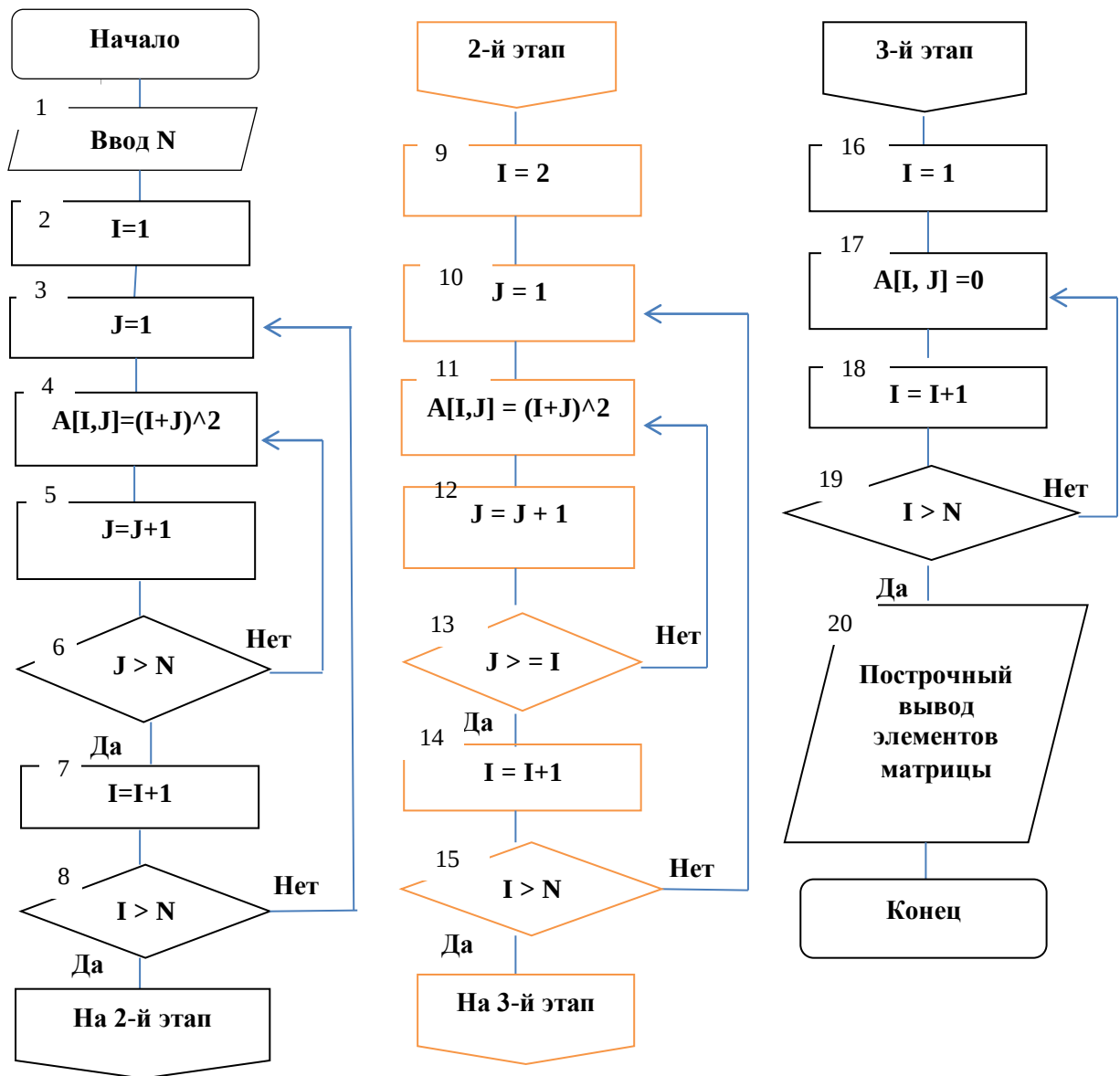
Пример 3. Даны действительные числа $x_1, x_2, \dots, x_6$. Получить квадратную

матрицу седьмого порядка [доп. 13]:

$$\begin{pmatrix} x_1 & x_2 & \dots & x_7 \\ x_1^2 & x_2^2 & \dots & x_6^2 \\ \dots & \dots & \dots & \dots \\ x_1^7 & x_2^7 & \dots & x_7^7 \end{pmatrix}$$


Пример 4. Пусть необходимо построить матрицу, элемент которой a_{ij} вычисляется как $(i + j)^2$, а затем зеркально отобразить её относительно главной диагонали. Элементам главной диагонали при этом присвоить

значение 0. На первом этапе построим вложенный цикл расчёта элементов и заполнения массива. На втором этапе проведём «зеркальное отражение» элементов, находящихся под главной диагональю. На третьем этапе, обнулим элементы главной диагонали.



Program matrix;

Var A: array [1..100, 1..100] of integer;

i, j, N : integer;

Begin

```

Write ('Введите размерность матрицы N');
Readln (N);
For i := 1 to n do {вложенные циклы формирования элементов матрицы}
  For j := 1 to N do
    A[i, j] := (i + j)*(i + j);
    For i := 2 to N do {вложенные циклы алгоритма зеркального отображения}
      For j := 1 to i-1 do
        A[j, i] := A[i, j];
        For i := 1 to N do {цикл обработки элементов главной диагонали}
          A[i, i] := 0;
          For i := 1 to N do {построчный вывод элементов матрицы}
Begin for j := 1 to N do
  Write (A[I, j] : 6); {отводится 6 позиций на вывод элемента}
Writeln;
End;
End.

```

Объектно-ориентированное программирование (ООП). Язык высокого уровня Lazarus.

Цель: ознакомиться с основными возможностями и порядком разработки программ в среде программирования *Lazarus*.

Методические указания

Чтобы создать программу в *Lazarus* необходимо¹⁸:

- 1) Создать форму программы (диалоговое окно), нанести компоненты, написать процедуры обработки событий;
- 2) В форме практически любого приложения есть базовые компоненты:
 Label – поле вывода текста;
 Edit – поле ввода текста, редактирования;
 Button – кнопка;
 CheckBox – независимая кнопка выбора;
 RadioButton – зависимая кнопка выбора;

¹⁸ Культин, Н. Б. Delphi в задачах и примерах. – СПб. : БХВ – Петербург, 2004. – 288 с.: ил.

ListBox – список выбора;

ComboBox – комбинированный список выбора;

- 3) Вид компонента, его поведение и размер определяются значениями свойств компонента;
- 4) Основную работу в программе выполняют процедуры обработки событий;
- 5) Для преобразования текста, например, находящегося в поле ввода/редактирования, в целое число нужно использовать функцию StrToInt, а в дробное – функцию StrToFloat. Для преобразования целого, например, значения переменной, в строку нужно использовать функцию IntToStr, а для преобразования дробного – функцию FloatToStr или FloatToStrF [доп. 8].

Пример 1. Вычислить сумму 10 элементов ряда: $1+1/2+1/3+\dots$ (значение i – го элемента ряда связано с его номером формулой $1/i$). После вычисления очередного элемента ряда в поле метки формы, предназначенное для вывода результата, вводится номер элемента ряда и его значение.

Диалоговое окно программы содержит два компонента: поле метки (label1) и командную кнопку (Button1). Вычисление суммы ряда и вывод результата выполняет процедура обработки события OnClick, текст которой приведён ниже [доп. 8].

```
Procedure TForm1.Button1Click(Sender: TObject);
Var i:integer; {номер элемента ряда}
    Elem:real; {значение элемента ряда}
    Summ:real; {сумма элементов ряда}
Begin
    Summ:=0;
    Label1.caption:=' ';
    For i:=1 to 10 do
        Begin
            Elem:=1/i;
            Label1.caption:=label1.caption+
                IntToStr (i)+' '+FloatToStr (elem)+'13;
            Summ:=Summ+elem;
        End;
    Label1.Caption:=label1.Caption+
        'сумма ряда:'+FloatToStr(summ);
end;
```

Если в инструкции for вместо слова To записать downto, то после очередного выполнения инструкций цикла, значение счётчика будет не увеличиваться, а уменьшаться.

Пример 2. Написать программу, которая вычисляет сопротивление электрической цепи, состоящей из двух сопротивлений. Сопротивления могут быть соединены последовательно или параллельно. Если величина сопротивления цепи превышает 1000 Ом, то результат должен быть выведен в килоомах [доп. 8].

```
// щелчок на кнопке Вычислить
procedure TForm1.Button1Click(Sender: TObject);
var
  r1,r2: real; // величины сопротивлений
  r: real;     // сопротивление цепи
begin
  // получить исходные данные
  r1 := StrToFloat(Edit1.Text);
  r2 := StrToFloat(Edit2.Text);

  if (r1 = 0) and (r2 = 0) then
  begin
    ShowMessage('Надо задать величину хотя бы одного сопротивления');
    exit;
  end;

  // переключатели RadioButton1 и RadioButton2
  // зависимые, поэтому о типе соединения можно
  // судить по состоянию одного из этих
  // переключателей
  if RadioButton1.Checked
  then // выбран переключатель Последовательно
    r:= r1+r2
  else // выбран переключатель Параллельно
    r:= (r1*r2)/(r1+r2);

  Label4.Caption := 'Сопротивление цепи: ';
  if r < 1000 then
    Label4.Caption := Label4.Caption +
      FloatToStrF(r,ffGeneral,3,2) + ' Ом'
  else
  begin
    r:=r/1000;
    Label4.Caption := Form1.Label4.Caption +
      FloatToStrF(r,ffGeneral,3,2) + ' кОм';
  end
end;
```

```

// щелчок на переключателе Последовательно
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
    // пользователь изменил тип соединения
    Label4.Caption := "
end;

// щелчок на переключателе Параллельно
procedure TForm1.RadioButton2Click (Sender: TObject);
begin
    // пользователь изменил тип соединения
    Label4. Caption: = "
end;
end.

```

Пример 3. Напишите программу, которая на поверхности формы вычерчивает график функции, например $2\sin(x)e^{x/5}$.

```

// функция, график которой надо построить
Function f (x: real): real;
begin
    f:= 2*Sin (x)*exp (x/5);
end;

//строит график функции
Procedure Groffunc;
var
    x1, x2: real;    //границы измерения аргумента функции
    y1, y2: real;    //границы изменения значения функции
    x: real;         //аргумент функции
    y:real;          //значения точки в точке x
    dx: real;        //приращение аргумента
    l, b: integer;   //левый нижний угол области вывода графика
    w, h:integer;    //ширина и высота области вывода графика
    mx,my:real;     //масштаб по осям x и z
    x0,y0:integer //точка- начало координат

begin
    // область вывода графика
    L:=10; // координата левого верхнего угла
    B:=Form1.ClientHeight-20; // координата левого верхнего угла
    H:= Form1.ClientHeight-40;// высота

```

```

W:= Form1.Width-40;// ширина
X1:=0; // нижняя граница диапазона аргумента
X2:=25;// верхняя граница диапазона аргумента
Dx:=0.01;//шаг аргумента
// найдём максимальные и минимальные значения функции на отрезке [x1,x2]
Y1:=f(x1);// минимум
Y2:=f(x1);// максимум
X:=x1;
Repeat
Y:=f(x);
If y<y1 then y1:=y;
If y>y2 then y2:=y;
X:=x+dx;
Until (x>=x2);
// вычислим масштаб
My:=h/abs(y2-y1); // масштаб по оси x
Mx:=w/abs(x2-x1); // масштаб по оси y
// оси
X0:=1;
Y0:=b-abs(round(y1*my));
With form1.Canvas do
Begin
// оси
moveTo (1,b); LineTo (1,b-h);
moveTo (x0,y0); LineTo (x0+w,y0);
TextOut(1+5,b-h, FloatToStrF(y2,ffGeneral,6,3));
TextOut(1+5,b, FloatToStrF(y1,ffGeneral,6,3));
// построение графика
X:=x1;
Repeat
Y:=f(x);
Pixels[x0+Round(x*mx),y0-Round(y*my)]:=clRed;
X:=x+dx;
Until (x>=x2);
End;
End;
Procedure TForm1.FormPaint(Sender:TObject);
Begin
GrOfFunc;
End;
// изменился размер окна программы
Procedure TForm1.FormResize(Sender:TObject);
Begin

```

```
// очистить форму
Form1.Canvas.FillRect(Rect(0,0,ClientWidth,ClientHeight));
// построить график
GrOfFunc;
End;
```

Практическая работа №12. Компьютерные технологии вычислений. Системы Matlab, Scilab.

Цель: познакомиться с основными приёмами работы в системе MatLab.

Цель: познакомиться с основными приёмами работы в системе MatLab и Scilab.

Matlab – одна из старейших, тщательно проработанных и проверенных временем систем автоматизации математических и научно-технических расчётов, построенная на расширенном представлении и применении матричных операций. Применение матриц как основных объектов системы способствует резкому уменьшению числа циклов, которые очень распространены при выполнении матричных вычислений на обычных языках программирования высокого уровня, и облегчению реализации параллельных вычислений¹⁹.

Matlab реализует три важные концепции программирования:

- 1) Процедурное модульное программирование, основанное на создании модулей – процедур и функций;
- 2) Объектно-ориентированное программирование, особенно ценное в реализации графических средств системы;
- 3) Визуально-ориентированное программирование, направленное на создание средств графического интерфейса пользователя GUI (Graphics User Interface).

Методические указания

Тригонометрические функции и обратные к ним:

sin, cos, tan, cot – синус, косинус, тангенс, котангенс;
sec, csc – секанс, косеканс ($\sec(x)=1/\cos(x)$, $\csc(x)=1/\sin(x)$);
asin, acos, atan, acot – арксинус, арккосинус, арктангенс, арккотангенс;
asec, acsc – арксеканс, арккосеканс.

Экспоненциальная функция, логарифмы, степенные функции:

¹⁹ Дьяконов В.П., Пеньков А.А. Matlab и Simulink в электроэнергетике. Справочник. – М.: Горячая линия – Телеком, 2009. – 816 с., ил.

Exp – экспоненциальная функция;
 Log – натуральный логарифм;
 Log10 – десятичный логарифм;
 Pow2 – возведение числа 2 в степень;
 Sqrt – квадратный корень.

Пример №1. Вычислить значение выражения:

$$e^{-2,5} (\ln 10,3)^{0,3} - \sqrt{\frac{\sin 2,35\pi + \cos 3,78\pi}{\operatorname{tg} 3,3}}$$

Введите в командной строке:

```
>>exp(-2.5)*log(10.3)^0.3-sqrt((sin(2.35*pi)+cos(3.78*pi))/tan(3.3))
ans = -3.2105
```

Пример №2. Вычислить сумму векторов $a = \begin{pmatrix} 1.3 \\ 5.4 \\ 6.9 \end{pmatrix}$ и $b = \begin{pmatrix} 7.1 \\ 3.5 \\ 8.2 \end{pmatrix}$

```
>>a=[1.3;5.4;6.9]
>>b=[7.1;3.5;8.2];
>>c=a+b
```

Пример №3. Вычислить значение cos сразу от всех элементов вектора c:

```
>>d=cos(c)
```

Пример №4. Из нескольких вектор – столбцов составить один:

```
>>v1=[1;2];
>>v2=[3;4;5];
>>v=[v1;v2]
```

Пример №5. Для сцепления вектор – строк:

```
>>v1=[1 2];
>>v2=[3 4 5];
>>v=[v1 v2]
```

Пример №6. Построить графики функций $y(x) = e^{-0.1x} \sin^2 x$ и $g(x) = e^{-0.2x} \sin^2 x$ на отрезке $[-2\pi; 2\pi]$.

```
>>x=[-2*pi:pi/20:2*pi];
>>f=exp(-0.1*x).*sin(x).^2;
>>g=exp(-0.2*x).*sin(x).^2;
>>plot(x,f,x,g)
```

Пример №7. Построить графики в логарифмических масштабах $f(x) = \ln 0.5x$ и $g(x) = \sin \ln(x)$ на отрезке $[0.1, 5]$ по оси x .

```
>>x=[0.1:0.01:10];
```

```
>>f=log(0.5*x);
```

```
>>g=sin(log(x));
```

```
>>semilogx(x,f,x,g)
```

Пример №8. Оформление графиков.

Цвет	Тип маркера	Тип линии
y жёлтый	· точка	- сплошная
m розовый	° кружок	: пунктирная
c голубой	X крестик	-.штрихпунктирная
r красный	+ знак «плюс»	-- штриховая
g зелёный	* звёздочка	
b синий	s квадрат	
k чёрный	d ромб	
w белый	^ треугольник вершиной вверх	

Положение легенды в графическом окне:

-1 – вне графика в правом верхнем углу графического окна;

0 – выбирается лучшее положение в пределах графика так, чтобы как можно меньше перекрывать сами графики;

1 – в верхнем углу графика;

2 – в верхнем левом углу графика;

3 – в нижнем левом углу графика;

4 – в верхнем левом углу графика.

grid on – нанесение сетки;

xlabel, ylabel – подписи к осям;

title – заголовок;

legend – легенда.

Выведем график изменения суточной температуры:

```
>>time=[0 4 7 9 10 11 12 13 13.5 14 14.5 15 16 17 18 20 22];
```

```
>>temp1=[14 15 14 16 18 17 20 22 24 28 25 28 25 20 16 13 13 14 13];
```

```
>>temp2=[12 13 13 14 16 18 20 21 23 25 20 25 20 16 13 11 20 15 13];
```

```
>>plot(time,temp1,'ro-',time,temp2,'go-')
```

```
>>grid on
```

```
>>title ('суточные температуры')
```

```
>>xlabel ('время (час.)')
```

```
>>ylabel ('Температура (C)')
```

```
>>legend ('3 июня', 4 июня')
```

Пример №9. Построить график параметрической функции:

$$x(t) = 0.5 \cdot \sin t, \quad y(t) = 0.5 \cdot \cos t \quad \text{для } t \in [0, 2\pi]$$

```
>>t=[0:0.01:2*pi];
```

```
>>x=0.5*sin(t);
```

```
>>y=0.5*cos(t);
```

```
>>plot(x,y)
```

Пример №10. Построить график функции:

$$y(x) = \begin{cases} 2\pi \cdot \sin x, & -2\pi \leq x \leq -\pi \\ \pi - |x|, & -\pi < x < \pi \\ \pi \cdot \sin^2 x, & \pi \leq x \leq 2\pi \end{cases}$$

```
>>x1=[-2*pi : 0.01 : -pi];
```

```
>>y1=2*pi*sin(x1);
```

```
>>x2=[-pi : 0.01 : pi];
```

```
>>y2=pi-abs(x2);
```

```
>>x3=[pi : 0.01 : 2*pi];
```

```
>>y3=pi*sin(x1).^2;
```

```
>>x=[x1 x2 x3];
```

```
>>y=[y1 y2 y3];
```

```
>>plot(x,y)
```

```
>>plot (x1,y1,x2,y2,x3,y3)
```

Пример №11. Ввести матрицу: $A = \begin{pmatrix} 3 & -1 & 2 \\ 4 & 4 & 1 \end{pmatrix}$

```
>>A=[3 -1 2;4 4 1]
```

Пример №12. Ввести квадратную матрицу: $B = \begin{pmatrix} 4 & 3 & -2 \\ 1 & 3 & 5 \\ -4 & 3 & 1 \end{pmatrix}$

```
>>B=[4 3 -2
```

```
1 3 5
```

```
-4 3 1]
```

Пример №13. Ввести матрицу два на три: $C = \begin{pmatrix} 3 & -4 & 2 \\ 4 & 2 & 1 \end{pmatrix}$

```
>>C=[[3;4] [-4;2] [2;1]]
```

Пример №14. Найти сумму и разность матриц C и A:

```
>>S=A+C
```

```
>>R=C-A
```

```
>>P=C*B
```

```
>>P=A*3
```

Транспонирование матриц:

```
>>B'
```

```
>>B.'
```

Для вещественных чисел эти матрицы приводят к одинаковым результатам.

Сопряжение и транспонирование матриц, содержащие комплексные числа, приведут к созданию разных матриц.

Пример №15. Перемножить матрицу и вектора: $[1 \ 4 \ -3] \begin{pmatrix} 2 & 1 & 3 \\ 1 & 0 & -2 \\ 0 & 5 & 3 \end{pmatrix} \begin{bmatrix} -4 \\ 3 \\ 2 \end{bmatrix}$

```
>>a=[1 4 -3];
```

```
>>B=[2 1 3;1 0 -2;0 5 3];
```

```
>>c=[-4;3;2];
```

```
>>a*B*c
```

Пример №16. Выполнить поэлементное умножение, деление, возведение в степень матриц: $A = \begin{pmatrix} 1 & 3 & -1 \\ 3 & 5 & 2 \end{pmatrix} \quad B = \begin{pmatrix} -3 & 1 & 5 \\ 3 & 2 & -2 \end{pmatrix}$

```
>>C=A.*B
```

```
>>D=A./B (деление элементов первой матрицы на соответствующие  
элементы второй матрицы)
```

```
>>D=A.\B (деление элементов второй матрицы на соответствующие  
элементы первой)
```

```
>>P=A.^2
```

```
>>P1=A.^B
```

Пример №17. Построить параметрически заданные линии:

$$x = e^{-|t-50|/50} \sin t, \quad y = e^{-|t-50|/50} \cos t, \quad z = t, \quad t \in [0,100]$$

```
>>t=[0:0.1:100];
```

```
>>x=exp(abs(t-50)/50).*sin(t);
```

```
>>y=exp(abs(t-50)/50).*cos(t);
```

```
>>z=t;
```

```
>>plot3(x,y,z)
```

```
>>grid on
```

```
>>plot3(x,y,z,'r:')
```

Пример №18. Построить траекторию движения точки в течении 10 секунд,

координаты которой изменяются по закону: $x(t) = \frac{\sin t}{t+1}$ $y(t) = \frac{\cos t}{t+1}$.

```
>>t=[0:0.001:10];  
>>x=sin(t)./(t+1);  
>>y=cos(t)./(t+1);  
>>comet(x,y)
```

Пример №19. Построить траекторию точки, перемещающейся в пространстве, координаты которой изменяются в течении 100 секунд по следующему закону:

$$x = e^{-|t-50|/50} \sin t, \quad y = e^{-|t-50|/50} \cos t, \quad z = t$$

```
>>t=[0:0.1:100];  
>>exp(abs(t-50)/50).*sin(t);  
>>y=exp(abs(t-50)/50).*cos(t);  
>>z=t;  
>>comet3(x,y,z)
```

Пример №20. Решите уравнение $\sin x - x^2 \cos x = 0$ на отрезке $[-5,5]$

Для более точного графика необходимо написать файл – функцию:

```
Function y=myf(x)  
Y=sin(x)-x.^2.*cos(x);  
fplot ('myf', [-5 5])  
grid on
```

Уточните значение корня, расположенного вблизи $x=-5$, при помощи fzero:

```
>>x1=fzero('myf',-5)
```

Zero found in the interval: [-4.7172, -5.2].

X1=-4.7566

Проверьте ответ, вычислив значение функции myf в точке x1:

```
>>myf(x1)
```

ans= 2.6645e-015

Найдите самостоятельно корни x_2 и x_3 , расположенные около -2 и -5 .

Для того, чтобы увидеть больше значащих цифр корня x_1 , следует установить формат long и вывести x_1 ещё раз (точность вычислений не зависит от формата вывода результата).

```
>>format long
```

```
>>x1
```

x1=-4.75655940570290

Использование fzero с двумя аргументами приводит к нахождению корня с точностью eps, где eps – встроенная функция MatLab, задающая точность вычислений, значение которой можно посмотреть так же, как и значения других переменных:

```
>>eps  
ans=2.220446049250313e-016.
```

Итак, корни уравнения были найдены с точностью ± 2 в шестнадцатом знаке после десятичной точки, т.е. практически с максимально возможной точностью.

Проверьте работу fzero, вычислив корень myf, расположенный вблизи нуля, там, где точное значение корня равно нулю.

```
>>x4=fzero('myf',-0.1)  
Zero found in the interval: [0.028, -0.19051].  
X4=-1.2423865059634334e-022
```

Пример №21. Вычислить все корни полинома:

$$p = x^7 + 3.2x^5 - 5.2x^4 + 0.5x^2 + x - 3.$$

Полином в MatLab задаётся вектором его коэффициентов:

```
>>p=[1 0 3.2 -5.2 0 0.5 1 -3];
```

Число элементов вектора, т.е. число коэффициентов полинома, всегда на единицу больше его степени, нулевые коэффициенты должны содержаться в векторе.

Функция polyval предназначена для вычисления значения полинома от некоторого аргумента:

```
>>polyval(p,1)  
ans= -2.5000
```

Нахождение сразу всех корней полиномов осуществляется при помощи функции roots, в качестве аргумента которой указывается вектор с коэффициентами полинома.

```
>>r = roots(p)  
r =  
-0.5668 + 2.0698i  
-0.5668 - 2.0698i  
1.2149  
0.5898 + 0.6435i  
0.5898 - 0.6435i  
-0.6305 + 0.5534i  
-0.6305 - 0.5534i
```

Число корней полинома, как известно, совпадает со степенью полинома.

Пример 22. Графическая иллюстрация поиска корней функций.

Создадим m-файл:

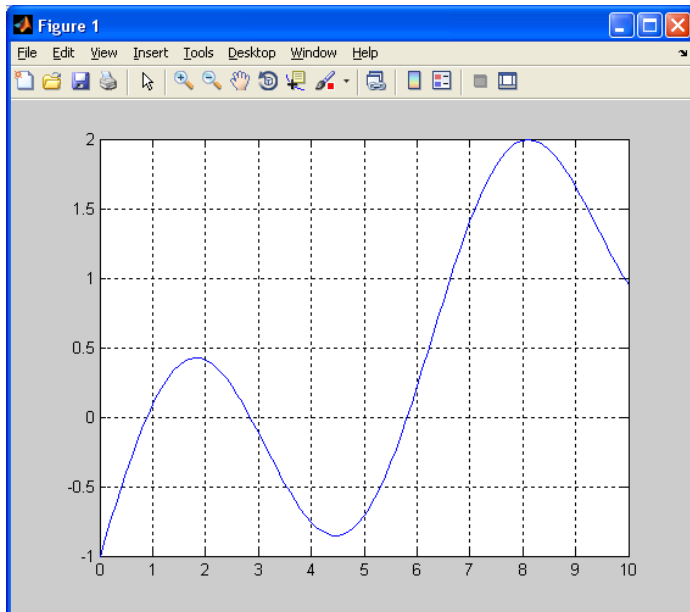
```
function f=fun1(x)
```

```
f=0.25*x+sin(x)-1;
```

Построим график функции:

```
>> x=0:0.1:10;
```

```
>> plot(x,fun1(x));
>> grid on
```

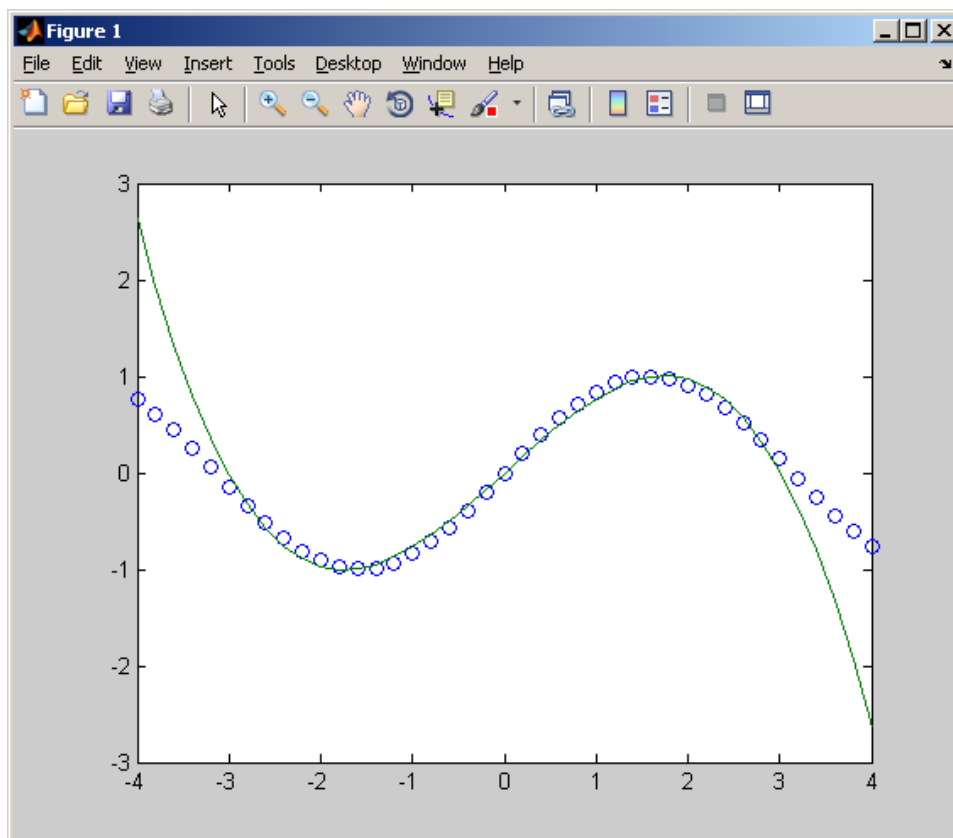


Из графика функции видны интервалы нахождения корней.

```
>> x1=fzero(@fun1,[0.5 1])
x1 = 0.8905
>> x2=fzero(@fun1,[2 3])
x2 = 2.8500
>> x3=fzero(@fun1,[5 6])
x3 = 5.8128
>> x3=fzero(@fun1,5,0.001)
x3 = 5.8128
```

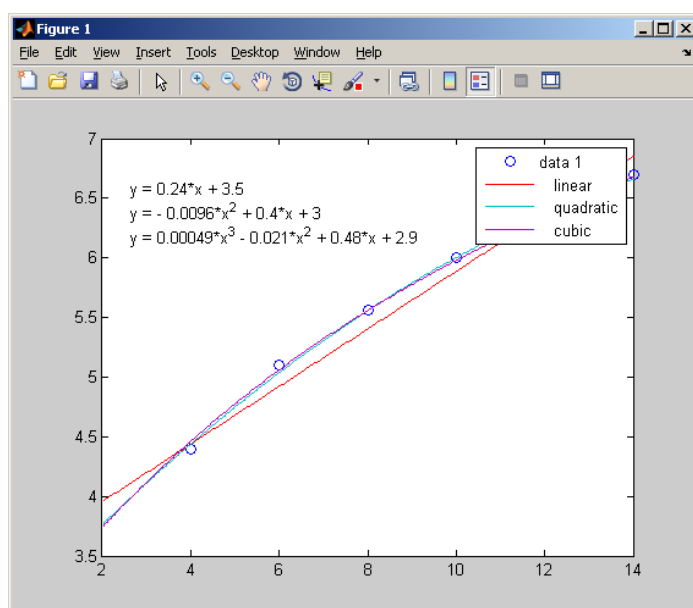
Пример 23. Полиномиальная регрессия.

```
>> x=(-3:0.2:3)';y=sin(x);p=polyfit(x,y,3)
p = -0.0953 0.0000 0.8651 -0.0000
>> x=(-4:0.2:4)';y=sin(x);
>> f=polyval(p,x);
>> plot(x,y,'o',x,f)
```



Пример 24. Полиномиальная регрессия для табличных данных.

```
>> x=[2,4,6,8,10,12,14];
>> y=[3.76,4.4,5.1,5.56,6,6.3,6.7];
>> plot(x,y,'o');
```



Data Statistics - 1

Statistics for: data 1

Check to plot statistics on figure:

	X		Y
min	2		3.76
max	14		6.7
mean	8		5.403
median	8		5.56
mode	2		3.76
std	4.32		1.054
range	12		2.94

Задания для самостоятельной работы:

1. Найти значение выражения: $(A+C)B^3(A-C)^T$, где

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 3 & -4 & 5 \\ 2 & 1 & -3 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -1 & 0 \\ 2 & 3 & 4 \\ 1 & -5 & 6 \end{pmatrix}, \quad C = \begin{pmatrix} 3 & 4 & 5 \\ 1 & -3 & 2 \\ 8 & 6 & -7 \end{pmatrix}$$

2. Решите систему линейных уравнений с тремя неизвестными:

$$а) \begin{cases} 2,34x_1 - 4,21x_2 - 11,61x_3 = 14,41 \\ 8,04x_1 + 5,22x_2 + 0,27x_3 = -6,44 \\ 3,92x_1 - 7,99x_2 + 8,37x_3 = 55,56 \end{cases}$$

$$б) \begin{cases} x_1 + 10x_2 - 5x_3 = 1 \\ 2x_1 + 4x_2 - x_3 = 2 \\ x_1 + 3x_2 + x_3 = 3 \end{cases}$$

Введите матрицу системы в массив A, для вектора правой части используйте массив b. Решите систему при помощи символа \:

>>x=A\b

3. Построить траекторию движения циклоиды:

$$x(t) = t - \sin t, \quad y(t) = 1 - \cos t$$

4. Построить график функции $\cos(x) = 0,1x$ на отрезке $[-10; 10]$ с шагом 0,1.

Технология построения SIMULINK-моделей

Цель: познакомиться с основными приёмами построения SIMULINK-моделей

Пример 1. Построить спираль Карно с применением следующих выражений²⁰:

$$x(t) = \int_0^1 \cos(x^2) dt$$

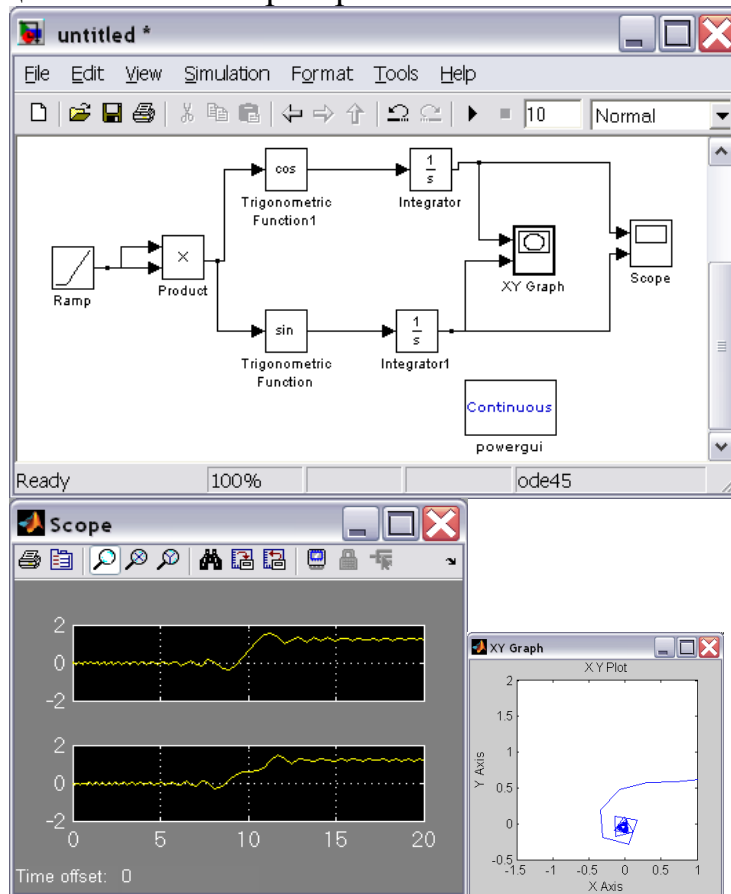
$$y(t) = \int_0^1 \sin(x^2) dt,$$

при изменении t от 0 до tmax.

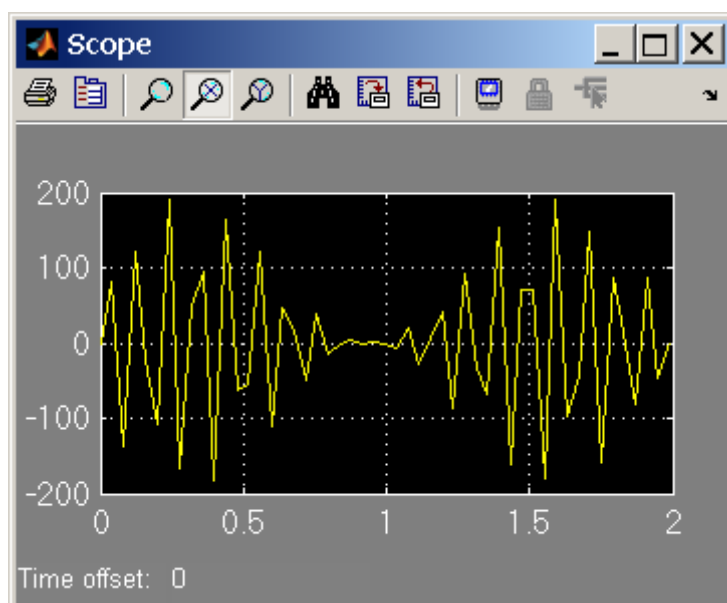
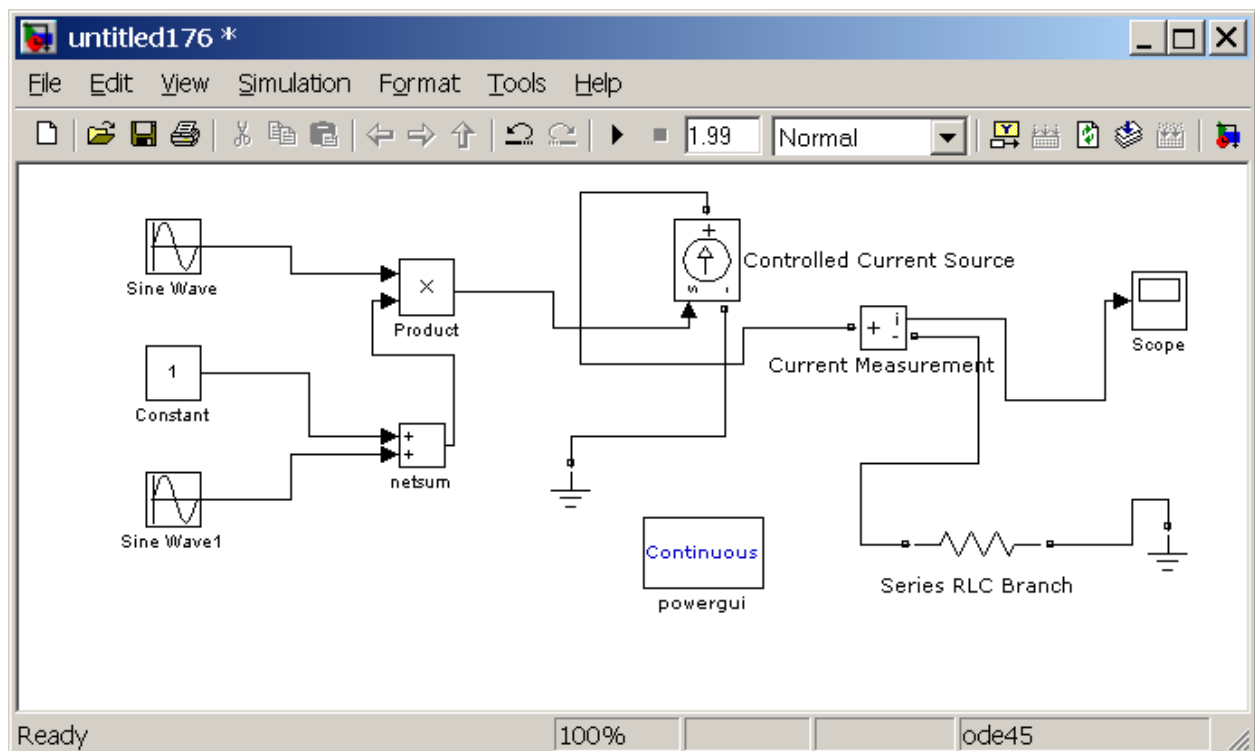
Текущее время формируется генератором нарастающего напряжения, которое начинается с уровня -10 для построения двух ветвей спирали. Затем

²⁰ Дьяконов В.П., Пеньков А.А. Matlab и Simulink в электроэнергетике. Справочник. – М.: Горячая линия – Телеком, 2009. – 816 с., ил.

ЭТОТ СИГНАЛ ВОЗВОДИТСЯ В КВАДРАТ И ЕГО СИНУС И КОСИНУС ИНТЕГРИРУЮТСЯ С ПОМОЩЬЮ БЛОКА ИНТЕГРАТОРОВ.



Пример 2. Построить амплитудно-модулированный сигнал.



Контрольные вопросы:

1. Что такое несущий сигнал? Несущая частота?
2. Что понимают под процессом модуляции?
3. Назовите виды модуляции.

Практическая работа №13. Построение оптимизационных моделей.

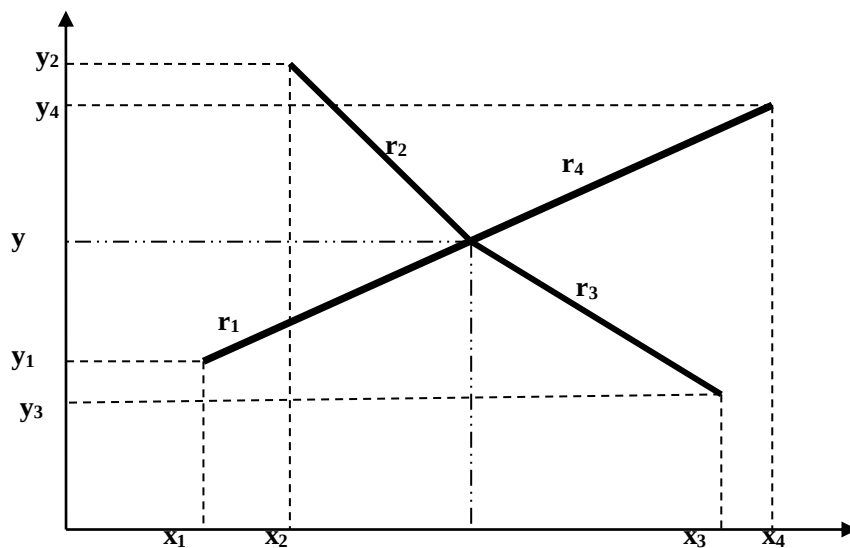
Цель: изучить методы построения оптимизационных моделей.

Задача о строительстве универсама

Имеется 4 жилых дома, расположенных в некотором микрорайоне города. Требуется определить местоположение для строительства универсама, так чтобы общее расстояние от построенного универсама до всех жилых домов было минимальным.

Математическая постановка задачи

Для математической постановки задачи следует ввести некоторые обозначения для географических координат 4-х исходных жилых домов. С этой целью введём некоторую прямоугольную систему координат, в которой исходные дома и универсам будут представлять отдельные точки на плоскости. Координаты исходных домов могут быть записаны как координаты соответствующих точек в виде: x_i, y_i , $i \in \{1, 2, 3, 4\}$. Искомые координаты универсама, который предполагается построить, можно положить равными: (x, y) . Очевидно, они служат переменными рассматриваемой задачи оптимизации, каждая из которых по своему характеру может принимать действительные значения из R^1 .



В некоторой фиксированной прямоугольной системе координат значения переменных x и y могут быть как положительными, так и отрицательными. Тем самым задачу о строительстве универсама можно считать задачей оптимизации без ограничений.

В качестве целевой функции данной задачи будем рассматривать сумму расстояний от искомой точки (x, y) до каждой из заданных точек (x_i, y_i) , ($\forall i \in \{1, 2, 3, 4\}$), рассчитанных по формуле Евклида. Каждое отдельное расстояние в этом случае равно: $r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$, где $i \in \{1, 2, 3, 4\}$.

Тогда общее расстояние будет определяться выражением:
 $r = r_1 + r_2 + r_3 + r_4$.

Таким образом, можно записать математическую модель:

$$f(x, y) = \sum_{i=1}^4 \sqrt{(x - x_i)^2 + (y - y_i)^2} \rightarrow \min,$$

где $\Delta = \{x, y \in R^1\}$.

Поскольку целевая функция данной задачи является нелинейной, задача о строительстве универсама относится к классу задач нелинейного программирования без ограничений.

Решение задачи в MS Excel:

	A	B	C	D
1	Переменные	Оптимальное решение		
2		x=	y=	
3	Исходные координаты домов		Целевая функция	=СУММ(B9:B12)
4		x1=	y1=	
5		x2=	y2=	
6		x3=	y3=	
7		x4=	y4=	
8	Расстояния до домов			
9		r1=	=КОРЕНЬ((\$B\$2-B4)^2+(\$D\$2-D4)^2)	
10		r2=	=КОРЕНЬ((\$B\$2-B5)^2+(\$D\$2-D5)^2)	
11		r3=	=КОРЕНЬ((\$B\$2-B6)^2+(\$D\$2-D6)^2)	
12		r4=	=КОРЕНЬ((\$B\$2-B7)^2+(\$D\$2-D7)^2)	

Для дальнейшего решения задачи следует вызвать мастер поиска решения. После этого следует в поле с именем Установить целевую ячейку ввести абсолютный адрес ячейки \$D\$3 со значением целевой функции, а в поле с именем Изменяя ячейки ввести абсолютный адрес ячеек \$B\$2; \$D\$2.

Поля с ограничениями можно оставить пустыми, поскольку целевая функция является выпуклой на всём множестве допустимых значений. Параметры Поиска решения можно оставить без изменения.

Параметры поиска решения

Оптимизировать целевую функцию:

До: ☐ Максимум ☒ Минимум ☐ Значения:

Изменяя ячейки переменных:

В соответствии с ограничениями:

☒ Сделать переменные без ограничений неотрицательными

Выберите метод решения:

Метод решения

Для гладких нелинейных задач используйте поиск решения нелинейных задач методом ОПГ, для линейных задач - поиск решения линейных задач симплекс-методом, а для негладких задач - эволюционный поиск решения.

Параметры

Все методы | Поиск решения нелинейных задач методом ОПГ

Точность ограничения: 0,000001

☒ Использовать автоматическое масштабирование

☐ Показывать результаты итераций

Решение с целочисленными ограничениями

☐ Игнорировать целочисленные ограничения

Целочисленная оптимальность (%): 5

Пределы решения

Максимальное время (в секундах): 100

Число итераций: 100

Эволюционные и целочисленные ограничения:

Максимальное число подзадач:

Максимальное число допустимых решений:

ОК Отмена

	A	B	C	D	E	F	G	H	I	J
1	Переменные	Оптимальное решение								
2	x=	10,04761486	y=	7,757595						
3	Исходные координаты домов		Целевая функция	36,1252						
4	x1=	1	y1=	2						
5	x2=	3	y2=	10						
6	x3=	25	y3=	3						
7	x4=	12	y4=	9						
8	Расстояния до домов									
9	r1=	10,7242356								
10	r2=	7,395759439								
11	r3=	15,69103332								
12	r4=	2,314169211								
13										
14										

Результатом решения задачи для 4-х жилых домов с координатами (1,2), (3,10), (25,3), (12,9) является оптимальное расположение универсама с координатами $x=10,0476148640742$ и $y=7,75759450334051$, при которых общее расстояние от универсама до жилых домов будет минимальным и равно значению: $f(x,y)=36,12520233$. Допустимая точность решения задаётся в каждом конкретном случае отдельно, исходя из специфики задачи²¹.

Оптимальное использование ресурсов

Теоретическое введение

Пусть предприятие может выпустить 4 вида продукции, для изготовления которой требуется сырьё одного и того же типа. Например, пусть такой продукцией является мебель²². Известен расход трудовых, сырьевых и финансовых ресурсов, требуемый для выпуска единицы каждого

²¹ Леоненков А.В. Решение задач оптимизации в среде MS Excel.- СПб.: БХВ-Петербург, 2005. – 704 с.: ил.

²² Королёв А.Л. Компьютерное моделирование. Лабораторный практикум - М.: БИНОМ. Лаборатория знаний, 2012.-296 с.: ил.-(Педагогическое образование).

вида продукции. Требуется спланировать производство так, чтобы получить максимальную прибыль от реализации продукции.

В качестве целевой функции F выберем прибыль, которую можно вычислить по формуле:

$$F = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \rightarrow \max$$

Здесь x_j - количество единиц j -го вида продукции, c_j - прибыль, которая будет получена при её реализации.

Общее количество продукции, которое может выпускать предприятие, ограничено количеством работников, производительностью труда, мощностью оборудования и т.п. Всё это составляет трудовые ресурсы предприятия. Кроме того, предприятие имеет определённые запасы сырья, из которого могут быть произведены разные виды продукции. Финансовые ресурсы предприятия также ограничены. Производство единицы каждого вида продукции требует различных затрат, например в виде зарплаты рабочим. Все эти факторы являются ограничениями для выпуска продукции.

Будем считать, что затраты всех ресурсов линейно зависят от количества произведённой продукции. Тогда указанные ограничения можно задать неравенств:

$$T_p = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 \leq b_1;$$

$$C_p = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 \leq b_2;$$

$$\Phi_p = a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 \leq b_3;$$

где T_p - расход трудовых ресурсов, C_p - расход сырьевых ресурсов, Φ_p - расход финансовых ресурсов, b_i - имеющийся запас i -го ресурса, $a_{i,j}$ - величина i -го ресурса, необходимая для производства единицы j -го вида продукции.

Кроме того, могут быть заданы явные ограничения на количество единиц каждого вида продукции в виде *граничных условий*:

$$e_j \leq x_j \leq d_j$$

Такие ограничения появляются, например, при выполнении предприятием заказа по контракту. Естественной же нижней границей для каждого вида продукции является нулевое значение, так как предприятие производит продукцию, а не закупает её.

В задаче оптимизации необходимо указать целочисленный тип допустимых значений переменных x_j .

Таким образом, задача оптимального использования ресурсов сводится к задаче линейной оптимизации или, точнее, к задаче целочисленной линейной оптимизации.

Постановка задачи моделирования

Определить оптимальный план производства исходя из заданных ограничений при условии получения максимума прибыли.

Порядок выполнения работы

Для решения задачи оптимизации средствами электронной таблицы нужно создать таблицу:

	A	B	C	D	E	F	G
1		Переменные					
2		x1	x2	x3	x4		
3	Значение	0	0	0	0		
4	Нижняя граница	0	0	0	0		
5	Верхняя граница	4	5	8	8	Прибыль	
6	Прибыль от реализации единицы продукции						
7	Вид ресурса	Потребное количество ресурса для производства единицы продукции				Использован ресурса	Запас ресурса
8		2	3	3	4	0	80
9		5	7	7	6	0	140
10		12	10	12	15	0	250
11							

Все формулы (вычисления прибыли и объёма использованных ресурсов) одинаковы по своей структуре, причём каждая формула представляет собой сумму произведений заданных коэффициентов и переменных x_j . Например, общая прибыль от реализации всех видов продукции вычисляется как сумма произведений объёма каждого вида продукции на величину прибыли от продажи единицы продукции. При вводе этих формул рекомендуется использовать функцию СУММПРОИЗВ.

В ячейки B6:E6 вводятся данные о прибыли при реализации единицы конкретного вида продукции-коэффициенты c_j целевой функции.

В ячейку F6 вводится формула для вычисления прибыли от реализации всего объёма произведённой продукции всех видов.

В ячейки F8:F10 вводятся формулы вычисления трудовых, сырьевых и финансовых затрат.

В ячейки G8:G10 вводятся значения коэффициентов b_j - запасы ресурсов, которыми располагает предприятие.

В ячейки B8:E10 вводятся данные о затратах ресурсов на производство единицы каждого вида продукции-коэффициенты $a_{i,j}$.

В ячейках B4:E5 задаются граничные значения для каждого вида продукции (если они заданы в задаче).

Диапазон ячеек B3:E3 является изменяемым, в эти ячейки будет помещено решение задачи.

В созданной таблице исходные данные выделите красным цветом, ячейки с формулами -синим цветом, а результат решения задачи- зелёным цветом. Далее вызовите надстройку **Поиск решения**.

В открывшемся диалоговом окне **Поиск решения** указывается адрес целевой ячейки (ячейка с формулой для вычисления прибыли), вид оптимизационной задачи (поиск максимума), диапазон изменяемых ячеек, где будет помещено найденное оптимальное решение, и, наконец, задаются ограничения.

Ввод и редактирование ограничений осуществляется с помощью кнопок диалогового окна **Поиск решения: Добавить, Изменить, Удалить**.

вращение эскиза вокруг оси, лежащей в плоскости эскиза; *кинематическая операция* – перемещение эскиза вдоль направляющей; *операции по сечениям* – построение объёмного элемента по нескольким эскизам, которые располагаются в нескольких параллельных плоскостях.

Задача 1

Построить трёхмерную модель цилиндра, диаметр основания которого равен 70 мм, а высота -80 мм. Цилиндр имеет одно продольное и два поперечных сквозных отверстия диаметром по 30 мм.

Порядок выполнения работы

Первым шагом при построении трёхмерной модели является запуск *подсистемы трёхмерного моделирования*, который производится нажатием кнопки **Создать...Деталь** на стартовой странице. В результате откроется окно построения трёхмерной модели.

Вначале необходимо ознакомиться с функциональными кнопками подсистемы трёхмерного моделирования. Обратите внимание на кнопку интерактивной справки, которая позволяет оперативно получить подсказку по любой команде.

Первоначально в *дереве модели* имеются система координат и три системных плоскости. Теперь следует выбрать одну из системных плоскостей (*X_Y, Z_X, Z_Y*) для построения первого эскиза. Выберите плоскость *X_Y*.

Далее надо задать ориентацию выделенной плоскости. Выберите **Нормально к....** Теперь можно создать плоский эскиз. Для этого следует включить *режим редактирования эскиза*. Содержание интерфейса при этом изменяется, становятся доступными *операции построения плоского эскиза*. Необходимо также включить режим отображения сетки; по умолчанию шаг сетки равен 5 мм.

Далее для точного построения эскиза установите *режим привязок* **По сетке**. Приступите к построению первого эскиза. Это будет окружность –

основание цилиндра с диаметром 70 мм. Выберите вариант построения **центр окружности – точка на окружности**. Центр окружности привяжите к началу координат.

Завершите редактирование эскиза и задайте его ориентацию **Изометрия XYZ**. Выполните операцию выдавливания. Расстояние выдавливания -80 мм.

Окончательное выполнение операции производится при нажатии кнопки **Создать**.

Теперь необходимо вырезать в ней продольное отверстие диаметром 30 мм. Выделите плоскость XY. Выберите её ориентацию **Нормально к...** Включите режим редактирования эскиза. Создайте новый эскиз – окружность диаметром 30 мм. Завершите редактирование эскиза и задайте для него ориентацию **Изометрия XYZ**. Выполните операцию **Вырезать выдавливанием**.

Необходимо правильно выбрать направление вырезания, иначе отверстие не будет вырезано. Расстояние вырезания следует задать несколько большим, чем длина детали.

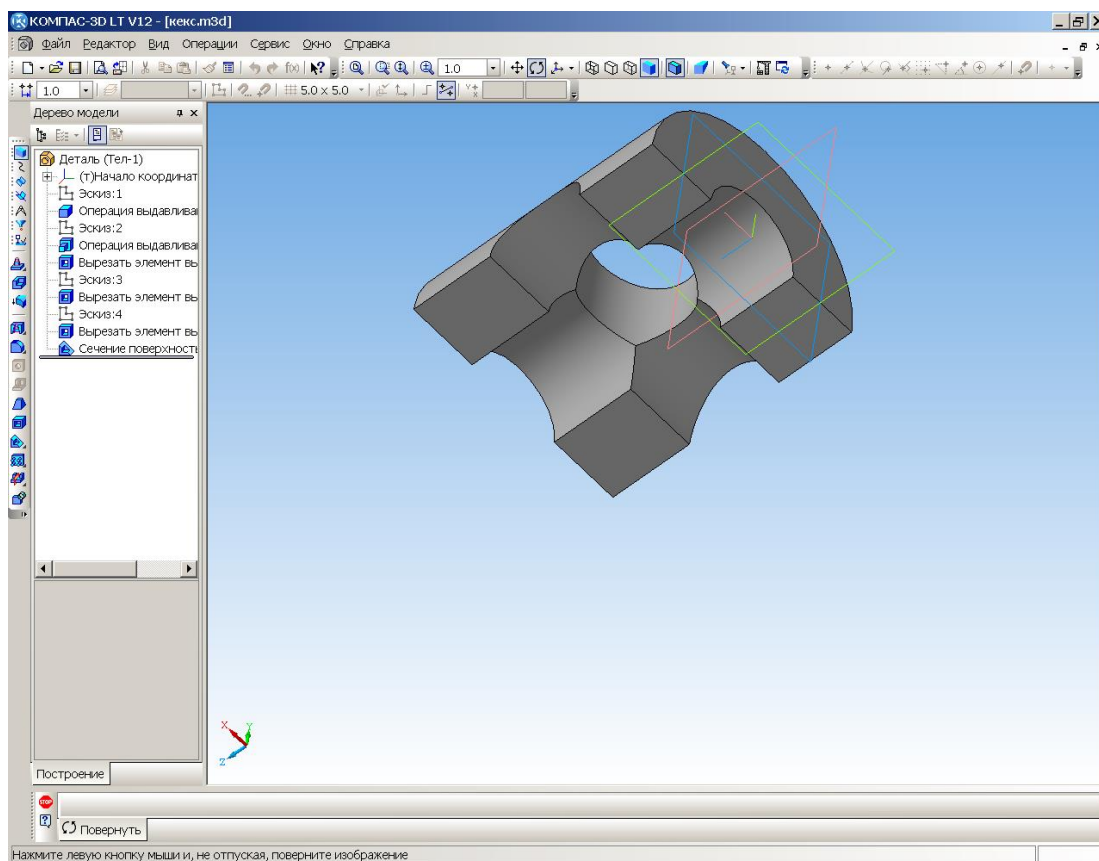
Вид детали на экране может быть задан в нескольких вариантах: *каркас, каркас без видимых линий, невидимые линии тонкие, полутонное или полутонное с каркасом*.

Теперь вырежьте по центру детали поперечные отверстия диаметром 30 мм. Для построения эскиза следует выбрать плоскость ZX. При выполнении операции следует выбрать режим **Два направления** и задать расстояния, достаточные для создания сквозного отверстия.

Самостоятельно создайте второе поперечное отверстие с тем же диаметром. Отображение системных плоскостей и координатных осей можно скрыть, если в дереве детали в контекстном меню каждой плоскости задать соответствующий режим.

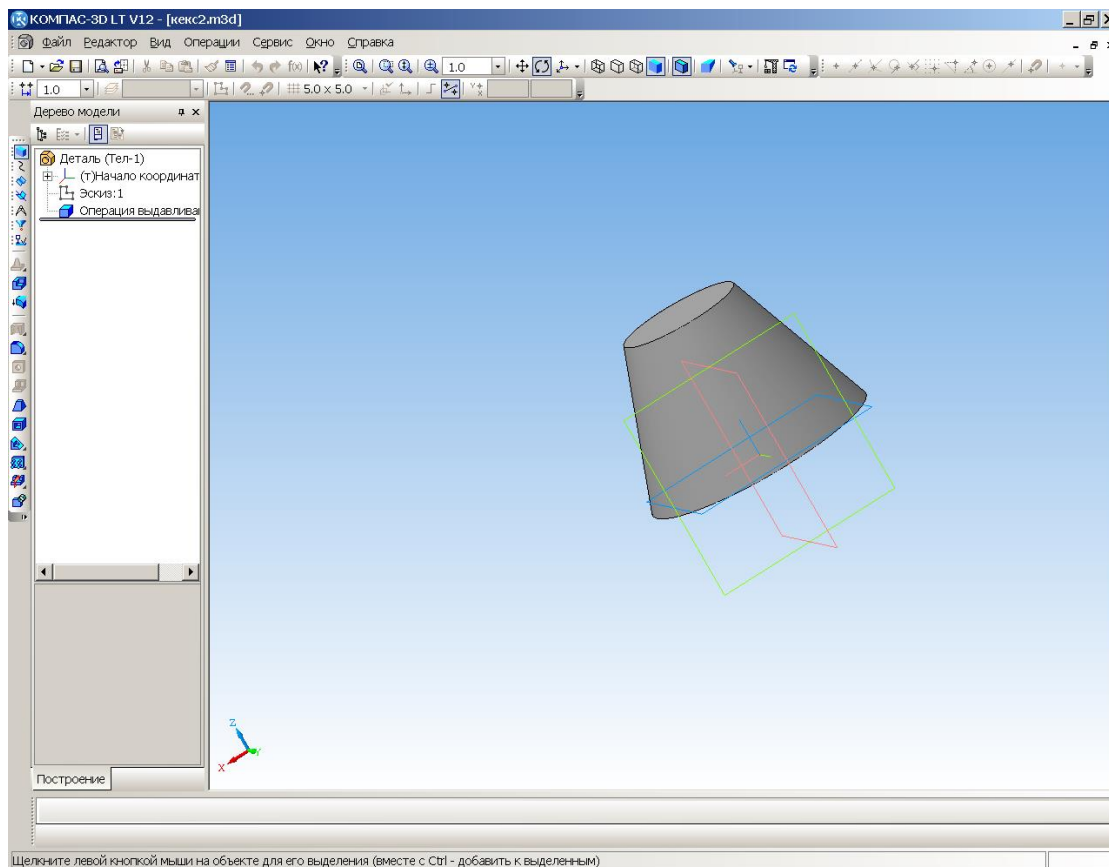
Цвет и оптические свойства детали установлены по умолчанию. Изменить эти параметры можно, выбрав в контекстном меню детали пункт **Свойства**.

В заключении постройте сечение тела плоскостью ZX . Для этого в дереве модели выделите плоскость ZX и выполните операцию **Сечение поверхностью**. В результате деталь примет вид, показанный на рисунке.



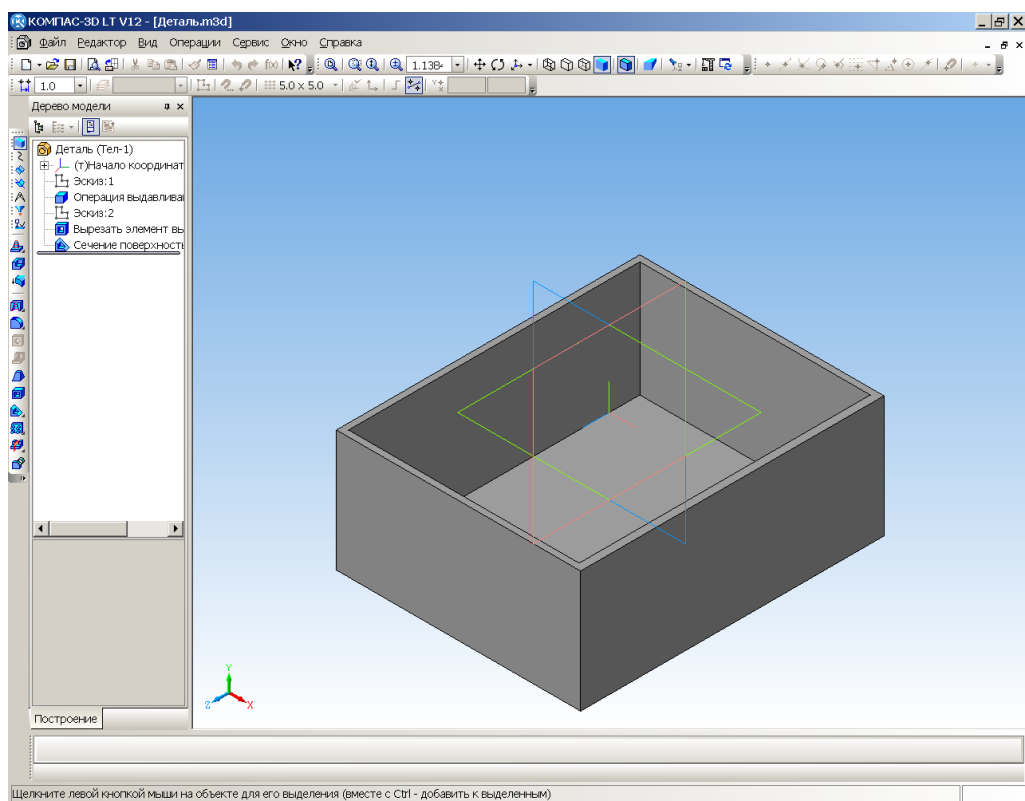
Задача 2

Создать модель усечённого конуса. Диаметр основания конуса – 60 мм. Высота конуса – 40 мм.



Задача 3

Построить модель тонкостенного параллелепипеда. В основании параллелепипеда лежит квадрат со стороной 80 мм, высота параллелепипеда – 100 мм, толщина стенок – 2мм.



Практическая работа №15. Моделирование в КОМПАС 3D. Построение чертежей.

Цель: научиться создавать чертежи.

Методические указания представлены в электронном учебнике:

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. Автоматики и вычисл. техники; сост. Лейко Н.Н., Майорова О.В. – Мурманск: Изд-во МГТУ, 2019 (файл находится на рабочих компьютерах в лабораториях).

Литература

Основная 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. Филимонова, Е. В. Математика и информатика : учебник для вузов / Е. В. Филимонова. - 3-е изд., перераб. и доп. - Москва : Дашков и К, 2010. - 480 с. : ил. - ISBN 978-5-394-00283-0 : 220-00. 22.1 - Ф 53. **Количество -30.**
4. Информатика. Общий курс : учеб. для вузов / А. Н. Гуда [и др.]; под общ. ред. В. И.

- Колесникова. - 2-е изд. - Москва : Наука-Пресс, 2008. - 398, [1] с. : ил. - Библиогр.: с. 391-392. - ISBN 978-5-91131-654-9 : 230-00. 32.97 - *И 74. Количество-1.*
5. Информатика : учебник для вузов / Н. В. Макарова [и др.] ; под ред. Н. В. Макаровой. - 3-е изд., перераб. - Москва : Финансы и статистика, 2007, 2006, 2005, 2004, 2002, 2000. - 768 с. : ил. - ISBN 5-279-02202-0 : 470-00; 380-00; 370-00; 250-00; 320-00; 305-00; 358-40. 32.97 - *И 74. Количество 306.*
6. Каймин, В. А. Информатика : учебник [для вузов] / В. А. Каймин; М-во образования РФ. - 4-е изд. - Москва : Инфра-М, 2004. - 283, [1] с. - (Высшее образование). - ISBN 5-16-001928-6 : 112-38. 32.97 - *К 15. Количество -2.*
7. Информатика : Введение в компьютерные науки : учебник для вузов / Л. Н. Королев, А. И. Миков. - Москва : Высш. шк., 2003. - 342 с. - ISBN 5-06-004272-3 : 153-16. 32.97 - *К 68. Количество -48.*
8. Советов, Б. Я. Информационные технологии : учебник для вузов / Б. Я. Советов, В. В. Цехановский. - Москва : Высш. шк., 2003. - 263 с. : ил. - ISBN 5-06-004275-8 : 145-86. 32.81 - *С 56. Количество -10.*
9. Могилев, А. В. Информатика : учеб. для вузов / А. В. Могилев; А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. - Москва : Академия, 2001, 2000. - 811 с. - ISBN 5-7695-0330-0 : 143-44; 100-94. 32.97 - *М 74. Количество -16.*
10. Острейковский, В. А. Информатика : учебник для вузов / В. А. Острейковский. - Москва : Высш. шк., 2001, 2001, 1999. - 511 с. : ил. - ISBN 5-06-003533-6 : 90-00; 66-67; 69-85; 39-30. 32.97 - *О-76. Количество-73.*
11. Иванова, Г. С. Объектно-ориентированное программирование : учебник для вузов / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев; под ред. Г. С. Ивановой. - Москва : МГТУ им. Н. Э. Баумана, 2001. - 320 с. : ил. - (Информатика в техническом университете). - ISBN 5-7038-1525-8 : 94-00. 32.97 - *И 20. Количество-2.*
12. Информатика [Электронный ресурс] : метод. указания к самостоят. работам для студентов техн. специальностей / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматики и вычисл. техники ; сост. З. А. Масыгина. - Электрон. текстовые дан. (1 файл : 748 Кб). - Мурманск: Изд-во МГТУ, 2015. http://elib.mstu.edu.ru/2015/M_15_15.pdf.
13. Информатика [Электронный ресурс] : метод. указания к расчет.-граф. заданиям для студентов 1 курса техн. специальностей / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматики и вычисл. техники ; сост. З. А. Масыгина. - Электрон. текстовые дан. (1 файл : 1.2 Мб). - Мурманск : Изд-во МГТУ, 2013. - Доступ из локальной сети Мурман. гос. техн. ун-та. - Загл. с экрана. http://elib.mstu.edu.ru/2013/M_13_24.pdf.
14. 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. автоматики 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.
15. Луковкин С.Б. Теоретические основы информатики : учеб. пособие для вузов / С. Б. Луковкин; Федер. агентство по рыболовству, ФГОУ ВПО "Мурман. гос. техн. ун-т". - Мурманск : Изд-во МГТУ, 2010. - 95 с. http://elib.mstu.edu.ru/2009/U_09_14.pdf.
16. 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. автоматики 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://elibr.mstu.edu.ru/2012/U_12_11.pdf.

17. Информатика. ч. 2 [Электронный ресурс] : опор. конспект лекций для студентов 1 курса техн. специальностей / Федер. агентство по рыболовству, Мурман. гос. техн. ун-т, Каф. автоматики 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.
18. Долинер, Л.И. Основы программирования в среде PascalABC.NET : учебное пособие / Л.И. Долинер ; науч. ред. Г.А. Матвеева ; Министерство образования и науки Российской Федерации, Уральский федеральный университет им. первого Президента России Б. Н. Ельцина. - Екатеринбург : Издательство Уральского университета, 2014. - 129 с. : ил. - Библиогр. в кн. - ISBN 978-5-7996-1260-3 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=275988> (29.11.2018).

Дополнительная литература:

1. Компьютерное моделирование. Лабораторный практикум / А.Л. Королёв.-М. : БИНОМ. Лаборатория знаний, 2012. – 296 с.: ил. (Педагогическое образование).
2. Дьяконов В.П., Пеньков А.А. Matlab и Simulink в электроэнергетике. Справочник. – М.: Горячая линия – Телеком, 2009. – 816 с., ил.
3. Калабухова Г.В. Компьютерный практикум по информатике. Офисные технологии. Гриф УМО МО РФ, 2016.
4. Информатика Учебник / Сергеева И.И., Музалевская А.А., Тарасова Н.В.- 2-е изд. перераб. и доп. – (Профессиональное образование)., (Гриф), 2014.
5. Информатика: учебник / Б.В. Соболев [и др.]. – Изд. 3-е, дополн. и перераб. – Ростов н/Д: Феникс, 2007. – 446 [1] с. – (Высшее образование).
6. Практикум по статистике в Excel: учебное пособие. / Б.В. Соболев [и др.] – Ростов н/Д: Феникс, 2010. – 381, [2] : ил. – (Высшее образование).
7. Практикум по информатике: Учебное пособие для вузов (+CD) / Под ред. проф. Н.В. Макаровой. – СПб.: Питер, 2012.- 320 с.: ил.
8. Информатика (для технических направлений): учебное пособие/ Н.И. Иопы. – 2-е изд., ст-ер.- М.: КНОРУС, 2012.- 472 с.- (бакалавриат).
9. Компьютерное моделирование. Лабораторный практикум / А.Л. Королёв.-М. : БИНОМ. Лаборатория знаний, 2012. – 296 с.: ил. (Педагогическое образование).
10. Сулейманов Р.Р. Компьютерное моделирование математических задач. Элективный курс : учебное пособие / Р.Р. Сулейманов. – М. : БИНОМ. Лаборатория знаний, 2012. – 381 с.: ил.
11. Информатика: базовый курс : учеб. для студентов вузов, бакалавров, магистров, обучающихся по направлению «Информатика и вычислительная техника» / О.А. Акулов, Н.В. Медведев. – 5-е изд., испр. и доп.- М.: Омега – Л, 2008. – 574 с.
12. Павловская Т.А. Паскаль. Программирование на языке высокого уровня: Учебник для вузов. – СПб.: Питер, 2008. – 393 с.: ил.
13. Волков Е.А. Численные методы. – М.: Наука, 2004.
14. Могилев А.В., Пак Н.И., Хеннер Е.К., Практикум по информатике. – М.: Академия, 2007.
15. Иопы Н.И. Информатика (для технических направлений) : учебное пособие – 2-е изд., стер. – М.: КНОРУС, 2012. – 472 с. – (Бакалавриат).
16. Баас, Р. Delphi 4 / Р/ Р. Баас, М. Фервай, Х. Гюнтер ; пер. с нем. под ред. А. Шевцова. - Киев : ВHV, 1999. - 459 с.

17. Кривилёв, А. В. Основы компьютерной математики с использованием системы MATLAB : [учеб. пособие] / Кривилёв А. В. - М. : Лекс - Книга, 2005. - 483, [1] с., [4] л. цв. ил.
18. Лабораторный практикум по информатике : учеб. пособие для вузов / В. С. Микшина [и др.] ; под ред. В. А. Острейковского. – М. : Высш. шк., 2003. – 376 с. : ил.
19. Мартынов, Н. Н. / MATLAB 5.x. Вычисления, визуализация, программирование / Н. Н. Мартынов, А. П. Иванов – М. : Кудиц – Образ, 2000. – 332 с.
20. Немнюгин, С. А. Turbo Pascal / С. А. Немнюгин. – СПб. : Питер, 2002. - 496 с. : ил.
21. Фаронов, В. В. Delphi 3. : учебный курс / Фаронов В. В. – М. : Нолидж, 1998. – 400 с. : ил.
22. Delphi 3. Программирование на Object Pascal – СПб. : BHV – Санкт – Петербург, 1998. – 304 с. : ил.
23. Колдаев, В.Д. Численные методы и программирование: учебное пособие / под ред. проф. Л. Г. Гагариной. – М. : ИД «Форум»: ИНФРА-М, 2008. – 336 с.: ил. – (Профессиональное образование).
24. Культин, Н. Б. Delphi в задачах и примерах. – СПб. : БХВ – Петербург, 2004. – 288 с. : ил.
25. Лапчик, М.П. Численные методы: Учеб. пособ. для студ. вузов /М. П. Лапчик, М. И. Рагулина, Е. К. Хеннер; под ред. М. П. Лапчика. – М. : Издательский центр «Академия», 2004. – 384 с.
26. Попов, А. М. Информатика: учеб. пособие для студентов вузов, обучающихся по специальности «Юриспруденция» (030501) / А. М. Попов, В. Н. Сотников, Е. М. Нагаева: под ред. А. М. Попова. –М. :ЮНИТИ-ДАНА, 2008.- 303 с.
27. Символоков, Л. В. Решение бизнес – задач в Microsoft Office – М. : ЗАО «Издательство БИНОМ», 2001 г. – 512 с. : ил.
28. Соболев, Б. В. Информатика : учебник / Б.В. Соболев [и др.]. – 3-е, дополн. и перераб. – Ростов н / Д: Феникс, 2007. – 446 [1] с. – (Высшее образование).
29. Соболев, Б. В. Практикум по информатике : / Б. В. Соболев [и др.]; под ред. Б.В. Соболева. – Ростов н / Д: Феникс, 2009. – 509 [1] с. – (Высшее образование).