

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

ИНФОРМАТИКА

Методические указания

к практическим работам для студентов 1 курса специальности

26.03.02 «Кораблестроение, океанотехника и системотехника объектов морской
инфраструктуры»

Мурманск

2019

Составитель - Бучкова Зоя Алексеевна, старший преподаватель кафедры автоматике и вычислительной техники Мурманского государственного технического университета

ОГЛАВЛЕНИЕ

1. <i>1 семестр.</i> Практическая работа №1	5
2. Практическая работа №2	11
3. Практическая работа №3	15
4. Практическая работа №4	20
5. Практическая работа №5	39
6. Практическая работа №6	83
7. <i>2 семестр.</i> Практическая работа №1	86
8. Практическая работа №2	87
9. Практическая работа №3	103
10. Практическая работа №4	134
11. Литература	151

Введение

Методические указания предназначены для проведения практических и лабораторных работ по Информатике.

Цель курса: выработать практические навыки работы с компьютером и его основным программным обеспечением, которое широко используется на практике.

Практическая работа №1. Системы счисления.

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.

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 (M – 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:

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

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	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1B	1C	1D
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.

а) $1000001_{(2)}$.

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

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

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:

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

$$\begin{array}{r}
 101101110100 \quad | \underline{1110011} \\
 - \underline{1110011} \qquad \quad 110011 \\
 \hline
 1000100 \\
 - \underline{1110011} \\
 \hline
 10101100 \\
 - \underline{1110011} \\
 \hline
 1110011 \\
 - \underline{1110011} \\
 \hline
 0
 \end{array}$$

Практическая работа №2. Кодирование данных в ЭВМ.

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 11111111_2 . В двухбайтовом формате от $00000000\ 00000000_2$ до $11111111\ 11111111_2$.

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 72_{10} в однобайтовом формате: 01001000_2

Число 72_{10} в двухбайтовом формате: 000000001001000_2 .

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: $X_{10} = 18, X_2 = 10010, [X_2]_{\text{пр}} = [X_2]_{\text{обр}} = [X_2]_{\text{доп}} = 0|10011, X_{10} = -11, X_2 = -1011, [X_2]_{\text{доп}} = [X_2]_{\text{обр}} + 1 = 1|0100 + 1 = 1|0101.$

Модифицированные обратный и дополнительный коды

Переполнение разрядной сетки может привести к переносу единицы в знаковый разряд, что приведёт к неправильному результату. Положительное число, получившееся в результате арифметической операции, может восприниматься как отрицательное, так как в знаковом разряде появится «1», и наоборот¹.

$$\text{Например: } X = 0,1011110$$

$$Y = 0,1101100$$

$$X + Y = 1,1001010$$

¹ 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).

X и Y- коды положительных чисел, но в процессе сложения в знаковом разряде появилась «1», что означает код отрицательного числа. Чтобы распознать переполнение разрядной сетки, вводятся модифицированные коды.

Модифицированный обратный код характеризуется тем, что под знак числа отводится не один, а два разряда. Форма записи чисел в модифицированном обратном коде выглядит следующим образом:

- для положительного числа

$$X = X_n X_{n-1} \dots X_2 X_1 X_0 \dots \Rightarrow X_{\text{обр}}^{\text{мод}} = 00, X_n X_n X_{n-1} \dots X_2 X_1 X_0$$

- для отрицательного числа

$$X = X_n X_{n-1} \dots X_2 X_1 X_0 \dots \Rightarrow X_{\text{обр}}^{\text{мод}} = 11, \bar{X}_n \bar{X}_{n-1} \dots \bar{X}_2 \bar{X}_1 \bar{X}_0$$

В модифицированных обратном и дополнительном кодах под знак числа отводится не один, а два разряда: «00» соответствует знаку «+», «11»-знаку «-». Любая другая комбинация («01» или «10»), получившаяся в знаковых разрядах, является признаком переполнения разрядной сетки. Сложение чисел в модифицированных кодах ничем не отличается от сложения в обычных обратном и дополнительном кодах.

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 устройства².

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

² 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.

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.

$$\begin{array}{r}
 +10 \\
 -3 \\
 \hline
 7
 \end{array}
 \qquad
 \begin{array}{r}
 +0\ 0001010 \\
 \underline{1\ 1111100} \text{ обратный код числа } -3 \\
 0\ 0000110 \\
 \xrightarrow{+1} \\
 0\ 0000111
 \end{array}$$

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

4) X и Y отрицательные.

$$\begin{array}{r}
 -3 \\
 +7 \\
 \hline
 -10
 \end{array}
 \qquad
 \begin{array}{r}
 +1\ 1111100 \text{ обратный код числа } -3 \\
 \underline{1\ 1111000} \text{ обратный код числа } -7 \\
 1\ 1110100 \\
 \xrightarrow{+1} \\
 1\ 1110101 \text{ обратный код числа } -10
 \end{array}$$

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, $2^{n-1} = 2^7 = 128$).

$$\begin{array}{r}
 +65 \\
 97 \\
 \hline
 162
 \end{array}
 \qquad
 \begin{array}{r}
 +0\ 1000001 \\
 \underline{0\ 1100001} \\
 0\ 0100010 \qquad \text{Переполнение}
 \end{array}$$

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.

б) 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

Практическая работа №3. Алгебра логики.

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, ∇ - логическое исключаящее “ИЛИ” (исключающая дизъюнкция);

\rightarrow - логическое следование (импликация);

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

$|$ - функция Шеффера;

\downarrow - стрелка Пирса, или функция Вебба °;

$\overrightarrow{\quad}$ и $\overleftarrow{\quad}$ - 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.

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

Отрицание – унарная (т.е. для одного операнда) логическая операция. Ей соответствует языковая конструкция, использующая частицу НЕ. Это правило можно записать в виде следующей таблицы:

A	\bar{A}
0	1
1	0

Конъюнкцией (логическим умножением) двух высказываний **A** и **B** является новое высказывание **C**, которое истинно только тогда, когда истинны оба высказывания, записывается $C=A \square B$ или $C=A \& B$.

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкцией (логическим сложением) двух высказываний **A** и **B** является новое высказывание **C**, которое истинно, если истинно хотя бы одно высказывание. Записывается $C=A \square B$.

A	B	A \square B
0	0	0
0	1	1
1	0	1
1	1	1

Импликацией двух высказываний **A** (называется *посылкой*) и **B** (называется *заключением*) является новое высказывание **C**, которое ложно только тогда, когда посылка истинна, а заключение ложно, записывается $C=A \rightarrow B$.

A	B	A \rightarrow B
----------	----------	-------------------------------------

0	0	1
0	1	1
1	0	0
1	1	1

Импликация имеет следующие свойства:

$$A \rightarrow B \neq B \rightarrow A$$

$$A \rightarrow A = 1$$

$$0 \rightarrow A = 1$$

$$1 \rightarrow A = A$$

$$A \rightarrow 1 = 1$$

$$A \rightarrow 0 = \bar{A}$$

Эквиваленцией двух высказываний **A** и **B** является новое высказывание **C**, которое истинно только тогда, когда оба высказывания имеют одинаковые значения истинности, записывается $C = A \leftrightarrow B$.

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Эквиваленция имеет следующие свойства:

$$A \leftrightarrow B = B \leftrightarrow A$$

$$A \leftrightarrow B = \bar{B} \leftrightarrow \bar{A}$$

$$A \leftrightarrow 1 = A$$

$$A \leftrightarrow 0 = \bar{A}$$

Чтобы избежать большого количества скобок в булевских функциях, принято следующее соглашение о старшинстве операций.

Первыми выполняются операции в скобках, затем операции в следующем порядке: отрицание, конъюнкция и дизъюнкция слева направо, импликация, эквиваленция.

Теоремы алгебры логики:

1. $\bar{\bar{A}} \equiv A$ - 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. $A \square \bar{A} = 1$ закон исключения третьего
11. $A \& \bar{A} = 0$ закон противоречия
12. $(A \& B) \vee A \equiv A$ - закон поглощения
13. $(A \vee B) \& A \equiv A$ - закон поглощения
14. $\overline{A \& B} \equiv \bar{A} \vee \bar{B}$ - закон де Моргана
15. $\overline{A \vee B} \equiv \bar{A} \& \bar{B}$ - закон де Моргана
16. $A \& 1 \equiv A$ - закон единицы для конъюнкции
17. $A \& 0 \equiv 0$ - закон нуля для конъюнкции
18. $A \vee 1 \equiv 1$ - закон единицы для дизъюнкции
19. $A \vee 0 \equiv A$ - закон нуля для дизъюнкции
20. $A \rightarrow B = \bar{A} \square B$
21. $A \leftrightarrow B = (A \rightarrow B) \& (B \rightarrow A) = (\bar{A} \square B) \& (A \square \bar{B}) = (A \& B) \square (\bar{A} \& \bar{B})$
22. $A \& (\bar{A} \square B) = A \& B$
23. $A \square (\bar{A} \& B) = A \square B$

Существует несколько стандартных форм, к которым приводятся логические выражения с помощью эквивалентных преобразований (формулы 1-23).

Первая из них – *дизъюнктивная нормальная форма (ДНФ)*, имеет вид $A_1 \square A_2 \square \dots \square A_n$, где каждое из составляющих высказываний есть конъюнкция простых высказываний и их отрицаний, например:

$$B = (\bar{A}_1 \& A_2 \& A_3) \square (A_4 \& A_5).$$

Вторая из них – *конъюнктивная нормальная форма (КНФ)*, имеет вид $A_1 \& A_2 \& \dots \& A_n$, где каждое из составляющих высказываний есть дизъюнкция простых высказываний и их отрицаний, например:

$$B = (\bar{A}_1 \square A_2 \square A_3) \& (A_4 \square A_5) \& A_6.$$

Пример 1. Построить таблицу истинности для функции

$$F(x_1, x_2, x_3) = (x_1 \rightarrow x_2) \rightarrow \bar{x}_3$$

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

Пример 2. Доказать тождество

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

$$(A + B) \cdot (\overline{A} + B) = B$$

Доказательство первого тождества проводится с использованием первого закона дистрибутивности:

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

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

Пример 3. Доказать закон свёртки логического выражения:

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

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

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

Пример 4. Используя основные равносильности алгебры логики, а также равносильности $A \rightarrow B \equiv \overline{A} \vee B$, упростить формулу $(X \rightarrow Y) \rightarrow Y$.

Решение:

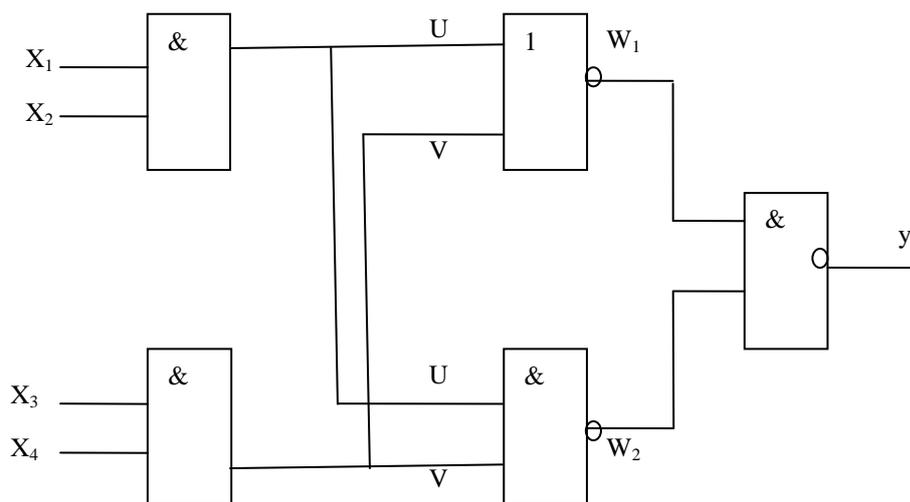
$$\begin{aligned} (X \rightarrow Y) \rightarrow Y &\equiv (\overline{X} \vee Y) \rightarrow Y \equiv \overline{\overline{X} \vee Y} \vee Y \equiv (\overline{X} \& \overline{Y}) \vee Y \equiv (X \& \overline{Y}) \vee Y \equiv \\ &\equiv (X \vee Y) \& (\overline{Y} \vee Y) \equiv (X \vee Y) \& 1 \equiv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

Пример 5. Для заданной комбинационной схемы постройте аналитическое выражение, упростите его с помощью равносильных преобразований и, если возможно, нарисуйте упрощённую схему. Здесь $U = x_1 \& x_2$, $V = x_3 \& x_4$

$$W_1 = \overline{U \vee V} = \overline{(x_1 \& x_2) \vee (x_3 \& x_4)},$$

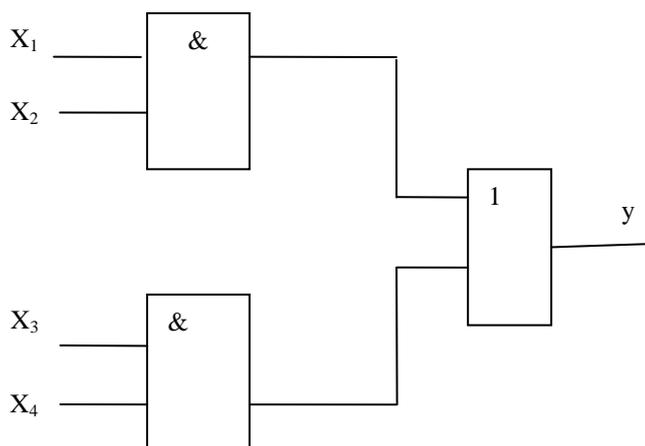
$$W_2 = \overline{U \& V} = \overline{(x_1 \& x_2) \& (x_3 \& x_4)},$$

$$y = \overline{W_1 \& W_2} = \overline{(x_1 \& x_2) \vee (x_3 \& x_4) \& (x_1 \& x_2) \& (x_3 \& x_4)}.$$



Преобразуем последнее выражение по закону де Моргана. Получаем $y = ((x_1 \& x_2) \vee (x_3 \& x_4)) \vee (x_1 \& x_2) \& (x_3 \& x_4)$.

Используя законы ассоциативности и правила приоритета логических операций, получаем $y = x_1 x_2 \vee x_3 x_4 \vee x_1 x_2 x_3 x_4$. Далее используем правило поглощения $A \vee AB \equiv A$, в результате получим упрощённую формулу, равносильную данной $y = x_1 x_2 \vee x_3 x_4$. Ей соответствует упрощённая комбинационная схема:



Практическая работа №4. Работа в прикладных программах. MS Word.

Цель: изучение основных операций процессора *Microsoft Word*. Нарработка 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.

Практическая работа с использованием редактора Microsoft Word

Основные операции:

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.
6. Структурирование и организация документа. Сноски. Названия, закладки, перекрёстные ссылки. Указатели, оглавления, списки иллюстраций.

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.

Задание 1. Создайте документ и наберите необходимый текст:

Системой счисления называется совокупность приемов наименования и записи чисел. В любой системе счисления для представления чисел выбираются некоторые символы (их называют цифрами), а остальные числа получаются в результате каких-либо операций над цифрами данной системы счисления.

Система называется позиционной, если значение каждой цифры (ее вес) изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число.

Число единиц какого-либо разряда, объединяемых в единицу более старшего разряда, называют основанием позиционной системы счисления. Если количество таких цифр равно P , то система счисления называется P -ичной. Основание системы счисления совпадает с количеством цифр, используемых для записи чисел в этой системе счисления.

Запись произвольного числа x в P -ичной позиционной системе счисления основывается на представлении этого числа в виде многочлена

$$x = a_n P^n + a_{n-1} P^{n-1} + \dots + a_1 P^1 + a_0 P^0 + a_{-1} P^{-1} + \dots + a_{-m} P^{-m}$$

Арифметические действия над числами в любой позиционной системе счисления производятся по тем же правилам, что и десятичной системе, так как все они основываются на правилах выполнения действий над соответствующими многочленами. При этом нужно только пользоваться теми таблицами сложения и умножения, которые соответствуют данному основанию P системы счисления.

1. Установите следующие параметры страницы: сверху и снизу 1,5 см, слева – 3 см, справа 2 см.
3. Выделите различными способами быстрого выделения фрагменты текста (символов, абзацев, строк, весь документ целиком).
4. Соедините два первых абзаца в один.
5. Переместите первый абзац в конец, используя метод "Drag-and-Drop".
6. Используя Буфер обмена верните первый абзац на прежнее место.
7. Вставьте в начало текста заголовок (объект WordArt).
8. Выровнять текст по ширине и установите полуторный межстрочный интервал.
6. Каждый абзац оформите с красной строки.
7. Сохраните созданный документ.

Задание №2. Автоматизация работы с текстом.

1. Откройте документ, созданный в первом задании.
2. Проверьте орфографию с помощью меню Рецензирование.
3. Создайте элементы автозамены и автотекста.
4. С помощью таблицы символов наберите следующее выражение: $\Sigma(\mu+\xi) \cdot r/s$.

Задание №3. Элементы издательской работы.

1. Создайте на всех чётных страницах колонтитул с названием данной работы.
2. Представьте данный текст в виде трёх колонок с разделителями и выровняйте их по ширине. Расстояние между колонками 0,7 см.
3. Создайте небольшой рисунок в графическом редакторе Paint и вставьте его в свой текстовый документ.

Задание №4. Использование списков и таблиц.

1. Создайте многоуровневый список в соответствии с образцом:

К техническим мероприятиям с использованием активных средств относятся:

- ❖ Пространственное зашумление:
 - Пространственное электромагнитное зашумление с использованием генераторов шума;
 - Создание акустических и вибрационных помех с использованием генераторов акустического шума;
 - Подавление диктофонов в режиме записи;
- ❖ Линейное зашумление:
 - Линейное зашумление линий электропитания;

Номер задания	Задание
1	Набрать любой текст, состоящий не менее чем из 15 строк, разделённый на три абзаца
2	Перенести первый абзац в конец текста
3	Поменять местами первое и последнее предложение в тексте
4	Выровнять абзацы по центру
5	Выровнять абзацы по левому и правому полям и создать для них красные строки
6	Выровнять последний абзац по ширине и создать для него красную строку
7	Выровнять последний абзац текста по центру
8	Добавить перед первым абзацем строку-заголовок
9	Добавить в конец текста строку с текущей датой
10	Добавить в конец текста строку со своими именем и фамилией
11	Выровнять строку -заголовок по правому полю
12	Установить для текста тип шрифта Times New Roman
13	Установить для текста размер шрифта 11
14	Установить для последней строки жирный шрифт
15	Установить для последней строки подчёркивание
16	Распечатать полученный документ
17	И т.д.

Документ (документированная информация)- зафиксированная на материальном носителе информация с реквизитами, позволяющими её идентифицировать³.

Бланк - это стандартный лист бумаги с воспроизведённой на нём постоянной информацией документа и местом, отведённым для переменной информации.

³ Журавлёва И.В., Журавлёва М.В. Оформляем документы на персональном компьютере: грамотно и красиво. – М.: ИНФРА-М, 2013. – 187 с. – (Просто, кратко, быстро).

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.

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**):

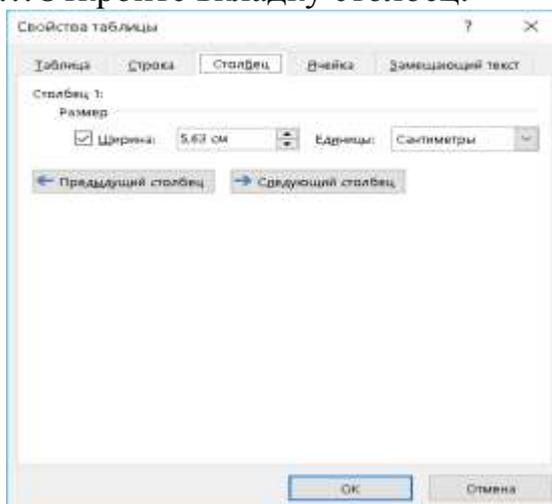
- левое поле - 3,0 см;
- правое поле – 1,5 см;
- верхнее поле – 2,0-2,5 см;
- нижнее поле – 2,0-2,5 см.

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**:

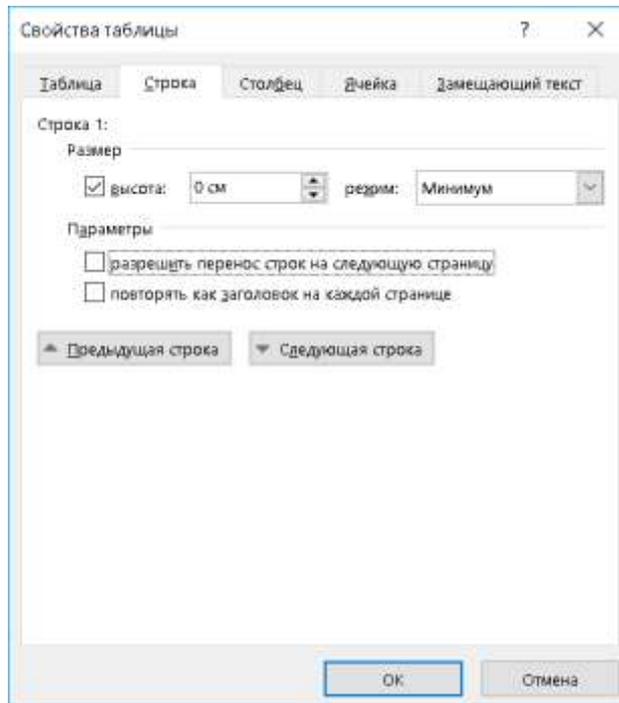


2. Исправьте цифру в поле **Ширина** на **7,3** см и нажмите кнопку **Следующий столбец**. Для второго столбца установите ширину **1,4** см, для третьего столбца – **7,8** см.

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

2. 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 2,0см;
- 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:

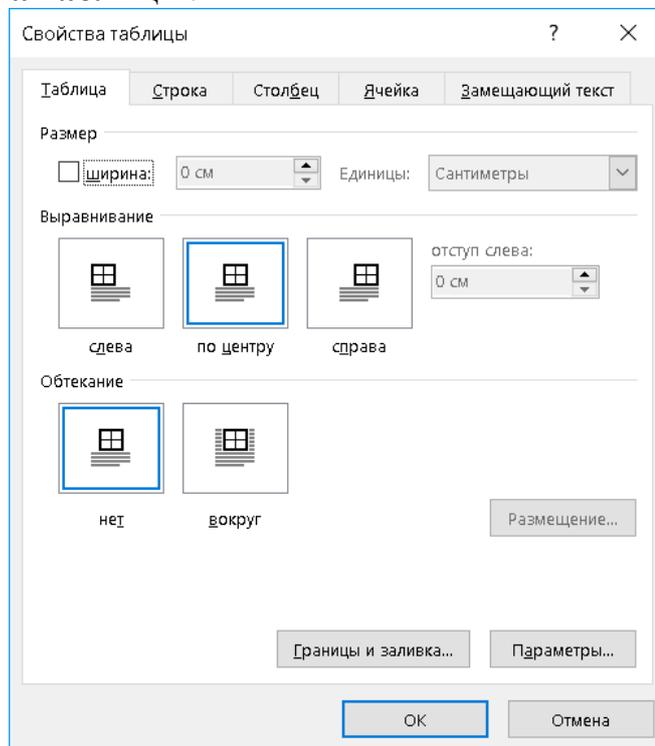
1		
2		
3	4	5
6		
7	8	9
10	11	12
13		

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:

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;
- 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 потребуются 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 **В** напечатайте знак «№». В ячейку **Д** введите «На №», в ячейке **Е** напечатайте предлог «от».

7. В ячейках **В** и **Е** 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.

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.ММ.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**.

- Здесь можно выбрать разные функции. **Печать** – распечатать письма. Или **Изменить часть писем**. Можно сохранить письма. Всё, документ готов. Закройте 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.

Макросы

Макрос- набор команд для автоматизации работы⁷. Хранятся макросы в шаблонах или документах. По умолчанию сохраняются в шаблоне Обычный для использования во всех документах. Если макрос применяется только в определённых документах, то его можно скопировать с помощью Организатора в

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).

шаблон, используемый в этих документах. Можно также закрепить макрос только за определённым документом.

1. Создать новый документ.
2. Ввести любых три заголовка, например, так:
 1. Шаблоны документов.
 - 1.1. Создание шаблонов.
 - 1.2. Применение шаблонов.
3. Установить мышью курсор за первым заголовком, нажать кнопку Непечатаемые символы и выполнить команду **Вид/Макрос/Запись макроса**.
4. В диалоговом окне Запись макроса ввести имя макроса ПолеОглавления и щёлкнуть кнопку **Клавишам**.
5. В диалоговом окне **Настройка клавиатуры** установить мышью курсор в поле Новое сочетание клавиш и нажать клавиши CTRL+ALT+1.
6. Выполнить команду **Вставка/Поле**.
7. В диалоговом окне **Поле** в группе **Категории** выбрать **Указатели**.
8. В группе **Поля** выбрать **ТС**, нажать кнопку **Коды поля**, в поле **Коды поля** после **ТС** ввести в двойных кавычках пробел и нажать **ОК**.
9. Выполнить команду **Вид\Макрос\Остановить запись**.
10. Установить курсор за вторым заголовком и выполнить команду **Вид\Макрос\Макросы**.
11. В диалоговом окне **Макрос** выделить макрос ПолеОглавления и нажать кнопку **Выполнить**.
12. За первым и вторым заголовками в фигурных скобках вместо пробелов набрать соответствующие заголовки.
13. Отжать кнопку Непечатаемые символы и произвести сборку оглавления.

Создание стилей, работа со сносками. Создание оглавления.

1. Скопируйте **Приложение 2** в вашу папку на диске X/:Avto.
2. Откройте Приложение 2 из вашей папки.

3. Создайте свой стиль: Главная — Стили открытие окна стилей



4. Укажите для стиля следующие параметры:

Имя стиля: Ф.И.О. студента

Основан на стиле: красная строка

Форматирование: шрифт - TimesNewRoman, размер - 14, интервал - полупетельный, выравнивание - по ширине, цвет текста - лиловый.

Использовать только в этом документе.

5. Примените созданный стиль к *первому абзацу*.

6. Измените форматирование в вашем стиле, присвойте имя «Ф.И.О. студента_изменен», примените измененный стиль к остальному тексту. Поместите измененный стиль на панель быстрого доступа.

7. Установите в тексте следующие сноски (вкладка **Сноски - Вставить сноску** или **Вставить концевую сноску**): к словам «информация» и «полнота» обычные сноски, к заголовку - концевую сноску:

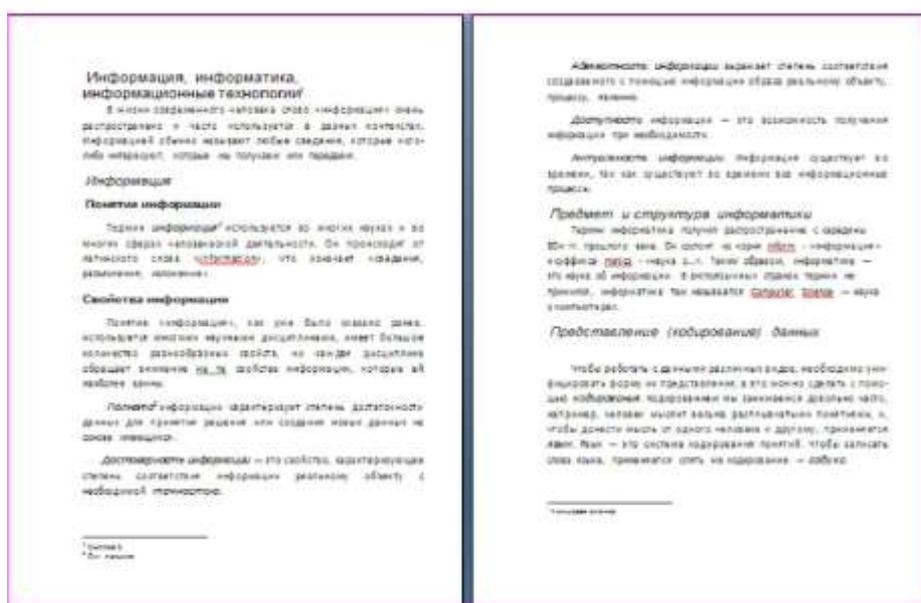


Рис. 2. Сноски

8. Продемонстрируйте работу преподавателю. (Созданный стиль удалите).

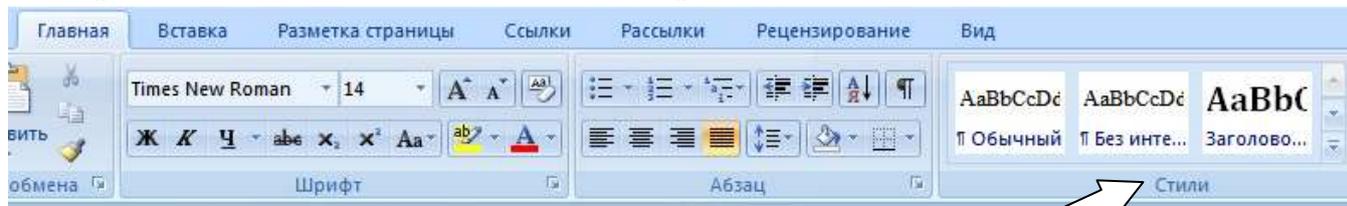
2. Вставка оглавления в MS Word

Перед началом работы скопируйте в свою папку документ Оглавление.doc.

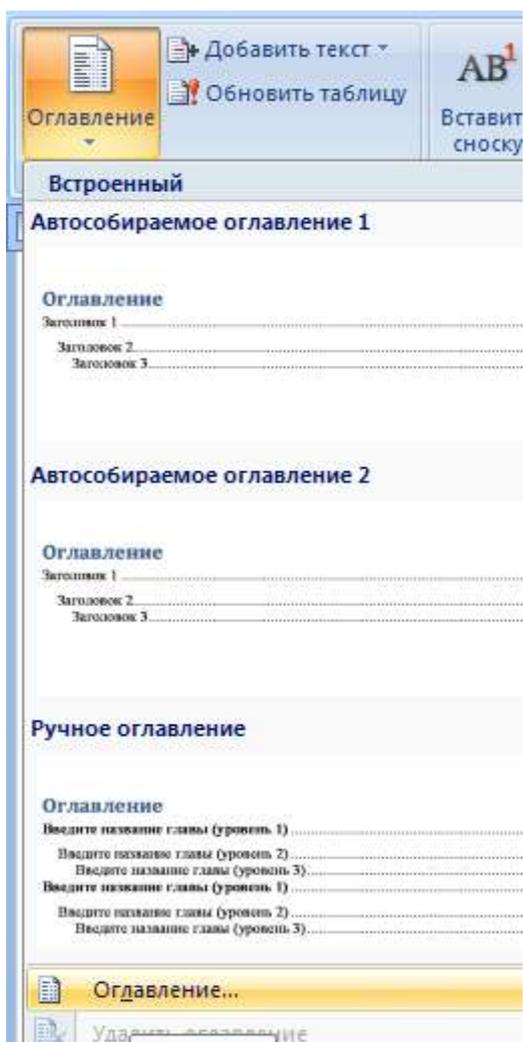
2.1 Для рефератов:

При создании оглавления реферата к заголовкам документа применяется стиль Заголовок 1.

Выбор стиля осуществляется с помощью раздела «стили» ленты «Главная»



1. Откройте документ **Оглавление.doc** из вашей папки.
2. Применим к следующим заголовкам: **Введение, глава 1, глава 2, заключение** – стиль **Заголовок 1**.
3. Для вставки Оглавления необходимо:
 - 3.1. Установить курсор в то место в документе, где будет вставляться оглавление, т.е. в нашем случае в начале документа, перед словом «Введение»
 - 3.2. Выполните команду **Ссылки/Оглавление**, кнопка находится к крайнем левом положении на ленте «ссылка». При нажатии на кнопку появится несколько возможных вариантов вывода оглавления, выберите Оглавление (см. рисунок ниже). В появившемся окне Оглавления оставьте текущие настройки без изменений и нажмите **ОК**.



У вас получится оглавление, указанное на рисунке:

ВВЕДЕНИЕ	1
ГЛАВА 1. БАНКОВСКИЕ ВЕКСЕЛЯ	2
ГЛАВА 2. ВЕКСЕЛЬ В СИСТЕМЕ СБЕРЕГАТЕЛЬНОГО БАНКА	14
ЗАКЛЮЧЕНИЕ	25

Практическая работа №5. Табличные процессоры. MS Excel.

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

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

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).

Ошибки в формулах

- #ДЕЛ/0 Деление на ноль;
- #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: $Y = e^{3x+1}$, $Y = \ln(1 + 3x)$, $Y = \sqrt{5 \sin x}$.

14. Открыть Книгу1 и выполнить ввод аргумента $x = 0,8$ в ячейку A1.

15. Найти нужную функцию (EXP, LN, КОРЕНЬ).

16. В диалоговом окне Аргументы функции ввести аргумент, например, $3 * x + 1$, где x - ссылка на ячейку A1.

17. 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 упражнения⁸:

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 $O_{\text{ср}}$.

⁸ Соболев, Б. В. Практикум по информатике : / Б. В. Соболев [и др.]; под ред. Б.В. Соболя. – Ростов н / Д: Феникс, 2009. – 509 [1] с. – (Высшее образование).

- Максимальное отклонение O_{\max} .
- Сумму произведений измерений на их отклонения от среднего значения $\sum_{i=1}^n Y_i O_i$.
- Сумму квадратов отклонения от среднего значения $\sum_{i=1}^n O_i^2$.

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:

$$A * X = B,$$

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:

$$X = A^{-1} * B.$$

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].

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

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

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

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

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

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

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

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

Нахождение корней уравнения методом Подбора параметра

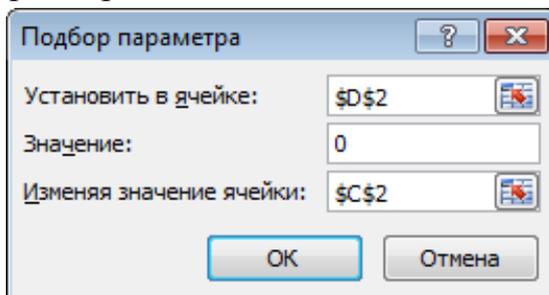
Цель: научиться применять метод подбора параметров для нахождения корней уравнений в MS Excel.

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

Microsoft Excel

Файл Правка Вид Вставка Формат
Сервис Данные Окно Справка

D2 fx =C2^3-2*C2^2+3*C2-5

бор_параметра

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			

Решение систем линейных уравнений методом Гаусса

Цель: научиться применять метод Гаусса при решении систем линейных уравнений.

Теоретическое введение

Метод Гаусса – метод последовательного исключения неизвестных. Пусть задана система:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \Lambda + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \Lambda + a_{2n}x_n = b_2 \\ a_{m1}x_1 + a_{m2}x_2 + \Lambda + a_{mn}x_n = b_m \end{cases}$$

Поставим ей в соответствие расширенную матрицу:

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \Lambda & a_{1n} & b_1 \\ a_{21} & a_{22} & \Lambda & a_{2n} & b_2 \\ \Lambda & \Lambda & \Lambda & & \Lambda \\ a_{m1} & a_{m2} & \Lambda & a_{mn} & b_m \end{array} \right)$$

Преобразуем матрицу к треугольному виду:

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \Lambda & a_{1n} & b_1 \\ 0 & a_{22} & \Lambda & a_{2n} & b_2 \\ \Lambda & \Lambda & \Lambda & & \Lambda \\ 0 & 0 & \Lambda & a_{nn} & b_n \end{array} \right)$$

Задание

Решить по методу Гаусса систему уравнений:

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

Порядок выполнения работы:

1. Начиная с ячейки A1 запишем все коэффициенты при неизвестных и свободные члены;
2. Выполним равносильные преобразования, которые будут приводить к последовательности систем, решение которых совпадает с решением исходной системы. Разделим каждое уравнение на коэффициент при x_1 .
В ячейке A5: =A1/\$A\$1;
3. Скопируем эту формулу до ячейки D5;
4. Аналогичные действия выполним в ячейках A6 и A7;
5. В ячейке A9 из второго уравнения вычитаем первое уравнение: =A6-A5 и скопируем до ячейки D9;
6. В ячейке A10 из третьего уравнения вычитаем первое: =A7-A5;
7. Во втором и третьем уравнении делаем коэффициент при x_2 равным 1.
Для этого всё второе уравнение делим на коэффициент при x_2 , а всё третье уравнение делим на коэффициент при x_3 . В ячейке B12: =B9/\$B\$9 и копируем до ячейки D12;
8. В ячейке B13: =B10/\$B\$10 и копируем до ячейки D13;
9. Из третьего уравнения исключим x_2 , вычитая из третьего уравнения второе. В ячейке B15: =B13-B12 и копируем до ячейки D15;
10. В третьем уравнении мы исключили x_1 и x_2 . Из третьего уравнения находим x_3 . В ячейке D17: =D15/C15;

11. Из второго уравнения находим x_2 . В ячейке D18: =D12-C12*D17;

12. Из первого уравнения находим x_1 . В ячейке D19: =D5-C5*D17-B5*D18.

The left screenshot shows the initial data and constraint equations in Excel. The right screenshot shows the solution results.

	A	B	C	D	
1	1	2	4	4	
2	3	-4	1	-1	
3	2	3	6	4	
4					
5	=A1/\$A\$1	=B1/\$A\$1	=C1/\$A\$1	=D1/\$A\$1	1
6	=A2/\$A\$2	=B2/\$A\$2	=C2/\$A\$2	=D2/\$A\$2	
7	=A3/\$A\$3	=B3/\$A\$3	=C3/\$A\$3	=D3/\$A\$3	
8					
9	=A6-A5	=B6-B5	=C6-C5	=D6-D5	
10	=A7-A5	=B7-B5	=C7-C5	=D7-D5	
11					
12		=B9/\$B\$9	=C9/\$B\$9	=D9/\$B\$9	2
13		=B10/\$B\$10	=C10/\$B\$10	=D10/\$B\$10	
14					
15		=B13-B12	=C13-C12	=D13-D12	3
16					
17			x3=	=D15/C15	
18			x2=	=D12-C12*D17	
19			x1=	=D5-C5*D17-B5*D18	

	A	B	C	D	E
1	1	2	4	4	
2	3	-4	1	-1	
3	2	3	6	4	
4					
5	1	2	4	4	1
6	1	-1,33	0,33	-0,333333	
7	1	1,5	3	2	
8					
9	0	-3,33	-3,67	-4,333333	
10	0	-0,5	-1	-2	
11					
12		1	1,1	1,3	2
13		1	2	4	
14					
15		0	0,9	2,7	3
16					
17			x3=	3	
18			x2=	-2	
19			x1=	-4	

Построение оптимизационных моделей

Цель: изучить методы построения оптимизационных моделей.

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 мебель⁹. Известен расход трудовых, сырьевых и финансовых ресурсов, требуемый для выпуска единицы каждого 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:

$$F = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \rightarrow \max$$

⁹ Королёв А.Л. Компьютерное моделирование. Лабораторный практикум - М.: БИНОМ. Лаборатория знаний, 2012.-296 с.: ил.-(Педагогическое образование).

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. Естественно 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.

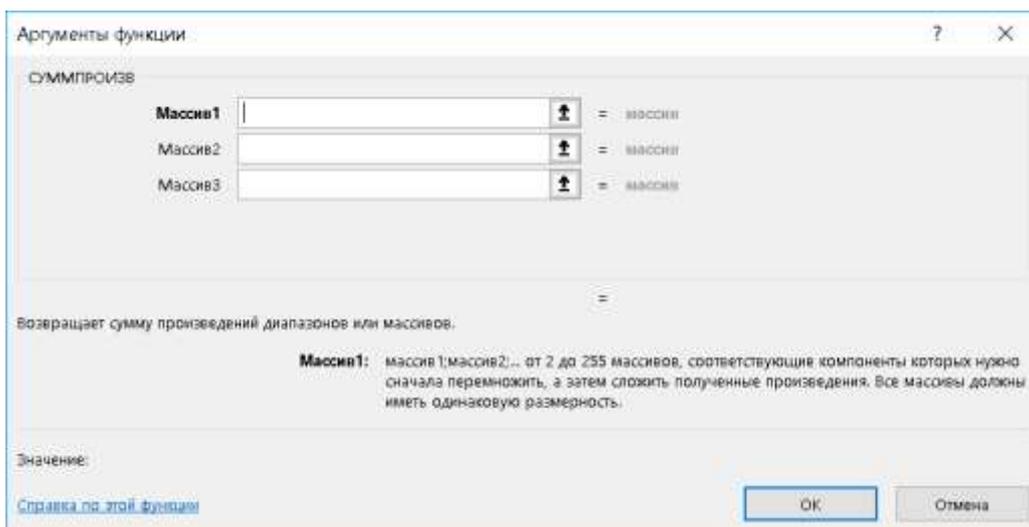
Порядок выполнения работы

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	80	95	125	150	0	
7	Вид ресурса	Потребное количество ресурса для производства единицы продукции				Использован ие 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 открытом 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**.

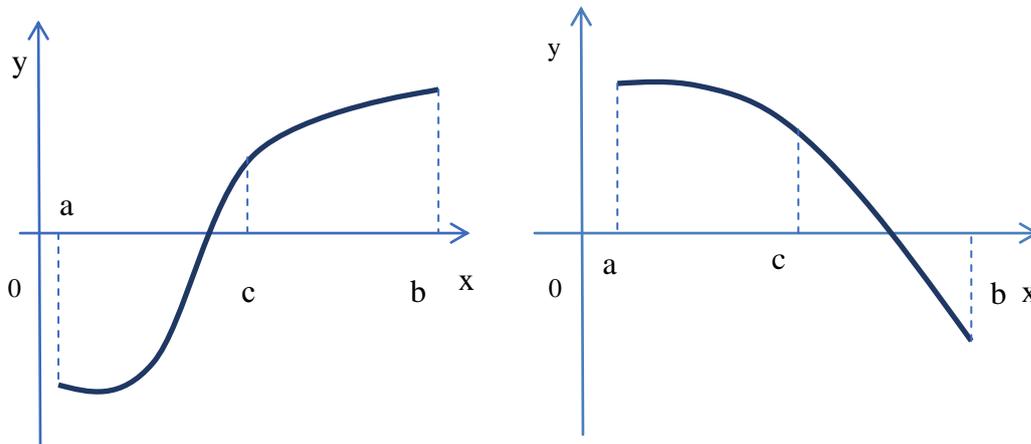
При решении задач линейной оптимизации необходимо по команде **Параметры** в окне **Поиск решения** выбрать вид модели **Линейная модель**. В этом случае для решения задачи используется симплекс-метод.

Уточнение корня уравнения методом половинного деления

Пусть функция $F(x) = 0$ имеет на отрезке $[a, b]$ единственный корень, причём функция $F(x)$ непрерывна на этом отрезке. Разделим отрезок $[a, b]$ пополам точкой $c = \frac{(a+b)}{2}$. Если $F(c) \neq 0$, то возможны два случая: $F(x)$ меняет знак либо на отрезке $[a, c]$, либо на отрезке $[c, b]$. Выбирая в каждом случае тот из отрезков, на котором функция меняет знак, и, продолжая путь половинного деления дальше, можно дойти до сколь угодно малого отрезка, содержащего корень уравнения.

Если на каком-то этапе процесса получен отрезок $[a, b]$, содержащий корень, то, приняв приближённо $x = \frac{(a+b)}{2}$, получим ошибку, не превышающую

значения $\Delta x = \frac{(b-a)}{2}$ (погрешность метода).



Функция $F(x)$ меняет знак на отрезке $[a; c]$; функция $F(x)$ меняет знак на отрезке $[c;b]$.

Задание

- 1) Отделить корень уравнения с помощью табличного процессора MSExcel.
- 2) Локализовать наименьший положительный корень уравнения $\sin 2x - \ln x = 0$ и уточнить его значение заданным методом с точностью $\varepsilon=0,0001$;

Порядок выполнения работы

Отделим корни уравнения:

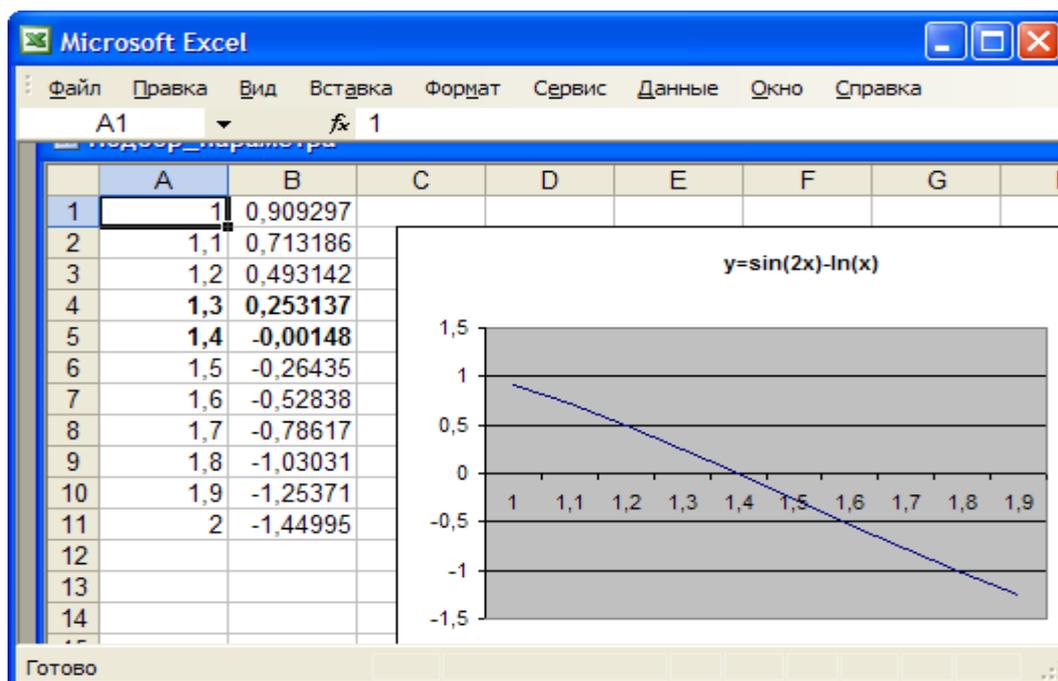


Рис.2. Отделение корня уравнения

Уточним корни уравнения. Для этого необходимо выполнить следующие действия (рис.3):

1. В ячейку B1 введите заданную точность $\varepsilon=0,0001$;
2. В ячейку A3 введите начальное значение отрезка 1,3;
3. В ячейку B3 введите конечное значение отрезка 1,5;
4. В ячейку C3 введите середину отрезка: $=(A3+B3)/2$;
5. В ячейку D3 введите: $=(\sin(2*A3)-\ln(A3))*(\sin(2*C3)-\ln(C3))$;
6. В ячейку E3 введите: $=\sin(2*A3)-\ln(A3)$;
7. В ячейку F3 введите:
 $=\text{ЕСЛИ}(B3-A3<\$B\$1; \text{”корень найден и равен”}\&\text{ТЕКСТ}(C3; \text{”0,0000”});$
8. В ячейку A4 введите: $=\text{ЕСЛИ}(D3<=0; A3; C3)$;
9. В ячейку B4 введите: $=\text{ЕСЛИ}(D3<=0; C3; B3)$;
10. В ячейку C4 введите: $=(A4+B4)/2$;
11. В ячейку D4 введите: $=(\sin(2*A4)-\ln(A4))*(\sin(2*C4)-\ln(C4))$;
12. В ячейку E4 введите: $=\sin(2*A4)-\ln(A4)$;
13. В ячейку F4 введите:
 $=\text{ЕСЛИ}(B4-A4<\$B\$1; \text{”корень найден и равен”}\&\text{ТЕКСТ}(C4; \text{”0,0000”});$
14. Выделите ячейки A4:F4 и протяните до ячейки A14.

Результат

	A	B	C	D	E	F	G	H
1	точность	0,0001						
2	a	b	c	проверка знака	значение функции в средней точке			
3	1,3	1,5	1,4	-0,000375677	-0,00148409	ЛОЖЬ		
4	1,3	1,4	1,35	0,032218098	0,127275288	ЛОЖЬ		
5	1,35	1,4	1,375	0,008044722	0,063207261	ЛОЖЬ		
6	1,375	1,4	1,3875	0,001955191	0,03093301	ЛОЖЬ		
7	1,3875	1,4	1,39375	0,000455999	0,014741496	ЛОЖЬ		
8	1,39375	1,4	1,396875	9,77783E-05	0,00663286	ЛОЖЬ		
9	1,396875	1,4	1,398438	1,70824E-05	0,002575413	ЛОЖЬ		
10	1,398438	1,4	1,399219	1,40596E-06	0,000545918	ЛОЖЬ		
11	1,399219	1,4	1,399609	-2,56047E-07	-0,00046902	ЛОЖЬ		
12	1,399219	1,399609	1,399414	2,09984E-08	3,84645E-05	ЛОЖЬ		
13	1,399414	1,399609	1,399512	-8,28041E-09	-0,00021527	ЛОЖЬ		
14	1,399414	1,399512	1,399463	-3,40041E-09	-8,8404E-05	конень найден и равен	1,3995	

Уточнение корня уравнения методом деления отрезка пополам

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 В1 следующее логическое выражение, используя функцию ЕСЛИ:

=ЕСЛИ (А1<0,2; 1+ln(1+А1); ЕСЛИ (А1<=0,8; (1+А1^(1/2))/(1+а1); 2*EXP(-2*А1)))

Пример 2. Построить поверхность $S = x^2 - y^2$ при $x \in [-2; 2]$, $y \in [-1; 1]$.

1. В ячейку А2 введите значение -2, в ячейку А3 – значение -1,8.
2. Выделите А2:А3 и с помощью маркера автозаполнения протяните на диапазон А4:А22. Т. о, значения аргумента x протабулированы от - 2 до 2 с шагом 0,2.
3. В ячейку В1 введите значение -1, в ячейку С1 введите значение -0,8. Аналогично протяните на диапазон D1:L1.
4. В ячейку В2 введите формулу: =\$А2^2 – В\$1^2.
5. Скопируйте формулу на диапазон В2:L22.
6. По полученному диапазону значений постройте поверхность.

Задачи и упражнения:

Постройте графики функций:

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

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

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

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.

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 **Данные¹⁰**.

	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.

¹⁰ Соболев, Б. В. Практикум по информатике: / Б. В. Соболев [и др.]; под ред. Б.В. Соболя. – Ростов н / Д: Феникс, 2009. – 509 [1] с. – (Высшее образование).

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**.

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**.

Использование расширенных фильтров

Для создания расширенного фильтра выполните следующие действия:

1. Копировать строку заголовков столбцов и поместить её справа от списка, отделив хотя бы одним пустым столбцом. Заголовки в диапазоне в диапазоне условий должны в точности совпадать с заголовками столбцов в списке.
2. Под скопированной строкой заголовков столбцов в строках, начиная со второй, вводить в соответствующие столбцы нужные критерии. Диапазон условий должен содержать по крайней мере две строки. В диапазоне условий нужно использовать заголовки только тех столбцов, которые используются в условиях отбора.
3. Операция **ИЛИ** создаётся с помощью строк, а операция **И** - с помощью одноимённых столбцов. Одноимённый столбец должен быть создан в скопированной строке заголовков столбцов рядом с заголовком основного столбца.
4. Анализ списка с помощью расширенного фильтра выполняется по команде **Данные/Фильтр/Расширенный фильтр**. В диалоговом окне **Расширенный фильтр** вводятся **Исходный диапазон** (определяется автоматически, если выделена любая ячейка списка) и **Диапазон условий** (выделяется с помощью мыши).

Для выполнения этой работы создайте список, состоящий не менее чем из 50 строк, или использовать готовую БД.

год	дата	месяц	объём	поставщик	продукция	район	сбыт	Цена за 1
2019	12.02.2019	февраль	4000	Иванов	напитки	западный	4016	1800
.....								

Найти все события сбыта продукции от 3000 до 5000 ед. в 2005 и 2006 годах:

1. На новый р\л копировать приведённый выше список.
2. Копировать строку заголовка справа от списка.

3. Вставить новый столбец рядом со столбцом Сбыт и копировать заголовок Сбыт в новый столбец.
4. В первой строке ниже заголовка ввести в столбцы Сбыт < 3000 и > 5000, в столбец Год ввести 2005.
5. Во вторую строку ниже заголовка в столбец Год ввести 2006, в столбцы Сбыт ввести ту же информацию.

Вычисляемые критерии. Три правила помогут избежать ошибок при использовании вычисляемого критерия:

1. Заголовок столбца над вычисляемым критерием *не должен совпадать* с заголовком какого-либо столбца в анализируемом списке. Он может быть пустым или содержать любой другой текст.
2. Ссылки на ячейки вне списка должны быть абсолютными.
3. Ссылки на ячейки внутри списка должны быть относительными, например, =H2>=\$V\$10, где V10- результат вычисления по формуле.

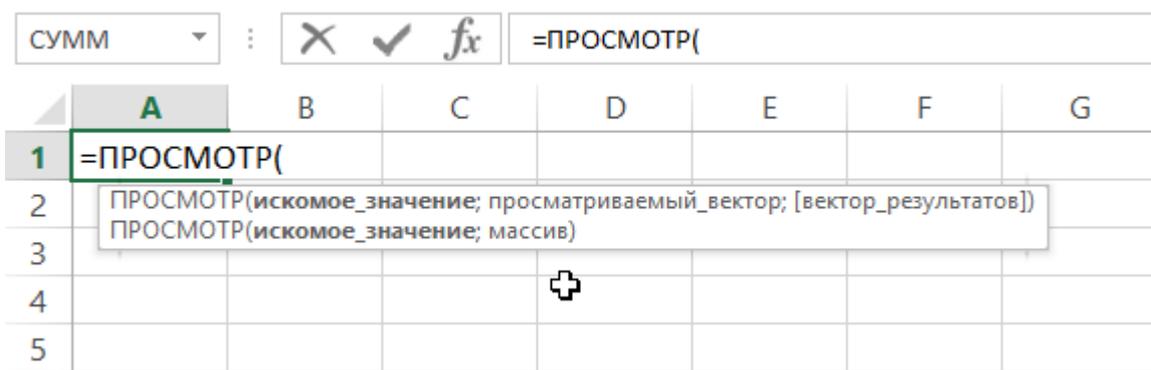
Добавим вычисляемый критерий. В ячейке V2 вычислим средний объём сбыта с помощью функции =СРЗНАЧ(H2:H25), а в ячейки U2 и U3 введём формулу =H2>=\$V\$2. При этом диапазон расширенного фильтра увеличится и станет равным K1:U3.

Использование формы данных производится по алгоритму:

1. Выделить любую ячейку в списке.
2. Выполнить команду **Данные/Форма**, нажать кнопку **Критерии**.
3. Заполнить поля ввода для создания списка критериев. Вводить можно только простые критерии.
4. Нажать кнопку **Далее**.
5. Пролистать выделенные записи, используя кнопки **Далее** и **Назад**.

Использование функции ПРОСМОТР

Начнем с того, что функция ПРОСМОТР имеет две формы записи: векторная и массив. Вводя функцию на рабочий лист, Excel напоминает Вам об этом следующим образом:



Форма массива

Форма массива очень похожа на функции ВПР и ГПР. Основная разница в том, что ГПР ищет значение в первой строке диапазона, ВПР в первом столбце, а функция ПРОСМОТР либо в первом столбце, либо в первой строке, в зависимости от размерности массива. Есть и другие отличия, но они менее существенны.

Данную форму записи мы подробно разбирать не будем, поскольку она давно устарела и оставлена в Excel только для совместимости с ранними версиями программы. Вместо нее рекомендуется использовать функции ВПР или ГПР.

Векторная форма

Функция ПРОСМОТР (в векторной форме) просматривает диапазон, который состоит из одной строки или одного столбца. Находит в нем заданное значение и возвращает результат из соответствующей ячейки второго диапазона, который также состоит из одной строки или столбца.

Пример 1

На рисунке ниже представлена таблица, где указаны номера телефонов и фамилии сотрудников. Наша задача по фамилии сотрудника определить его номер телефона.

	A	B	C	D
1	34-25-30	Анисимов		
2	82-12-63	Егоров		
3	36-15-25	Игнатов		
4	75-14-12	Карманов		
5	32-20-74	Павлюченко		
6	45-16-75	Петров		
7	62-85-92	Соколов		
8				

В данном примере функцию ВПР не применить, поскольку просматриваемый столбец не является крайним левым. Именно в таких случаях можно использовать функцию ПРОСМОТР. Формула будет выглядеть следующим образом:

	A	B	C	D	E	F
1	34-25-30	Анисимов	Игнатов	=ПРОСМОТР(C1;B1:B7;A1:A7)		
2	82-12-63	Егоров				
3	36-15-25	Игнатов				
4	75-14-12	Карманов				
5	32-20-74	Павлюченко				
6	45-16-75	Петров				
7	62-85-92	Соколов				
8						

Первым аргументом функции ПРОСМОТР является ячейка C1, где мы указываем искомое значение, т.е. фамилию. Диапазон B1:B7 является просматриваемым, его еще называют просматриваемый вектор. Из соответствующей ячейки диапазона A1:A7 функция ПРОСМОТР возвращает результат, такой диапазон также называют вектором результатов. Нажав Enter, убеждаемся, что все верно.

	A	B	C	D	E
1	34-25-30	Анисимов	Игнатов	36-15-25	
2	82-12-63	Егоров			
3	36-15-25	Игнатов			
4	75-14-12	Карманов			
5	32-20-74	Павлюченко			
6	45-16-75	Петров			
7	62-85-92	Соколов			
8					

Пример 2

Функцию ПРОСМОТР в Excel удобно использовать, когда векторы просмотра и результатов относятся к разным таблицам, располагаются в отдаленных частях листа или же вовсе на разных листах. Самое главное, чтобы оба вектора имели одинаковую размерность.

На рисунке ниже Вы можете увидеть один из таких примеров:

	A	B	C	D	E
1	34-25-30	Петров	=ПРОСМОТР(B1;D5:D11;A1:A7)		
2	82-12-63				
3	36-15-25				
4	75-14-12				
5	32-20-74			Анисимов	
6	45-16-75			Егоров	
7	62-85-92			Игнатов	
8				Карманов	
9				Павлюченко	
10				Петров	
11				Соколов	
12					

Как видите, диапазоны смещены друг относительно друга, как по вертикали, так и по горизонтали, но формула все равно вернет правильный результат. Главное, чтобы размерность векторов совпадала. Нажав Enter, мы получим требуемый результат:

	A	B	C	D	E
1	34-25-30	Петров	45-16-75		
2	82-12-63				
3	36-15-25				
4	75-14-12				
5	32-20-74			Анисимов	
6	45-16-75			Егоров	
7	62-85-92			Игнатов	
8				Карманов	
9				Павлюченко	
10				Петров	
11				Соколов	
12					

При использовании функции ПРОСМОТР в Excel значения в просматриваемом векторе должны быть отсортированы в порядке возрастания, иначе она может вернуть неверный результат.

Так вот коротко и на примерах мы познакомились с функцией ПРОСМОТР и научились использовать ее в рабочих книгах Excel. Надеюсь, что данная информация оказалась для Вас полезной, и Вы обязательно найдете ей применение. Всего Вам доброго и успехов в изучении Excel.

Использование функции ЕПУСТО

Функция ЕПУСТО в Excel используется для наличия текстовых, числовых, логических и прочих типов данных в указанной ячейке и возвращает логическое значение ИСТИНА, если ячейка является пустой. Если в указанной ячейке содержатся какие-либо данные, результатом выполнения функции ЕПУСТО будет логическое значение ЛОЖЬ.

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ФУНКЦИИ ЕПУСТО В EXCEL

Пример 1. В таблице Excel находятся результаты (баллы) по экзамену, который проводился в учебном заведении. В данной электронной ведомости напротив некоторых студентов оценки не указаны, поскольку их отправили на пересдачу. В столбце рядом вывести текстовую строку «Сдал» напротив тех, кому выставили оценки, и «На пересдачу» - напротив не сдавших с первого раза.

Исходные данные:

	A	B	C
1	Ведомость		
2	Код студента	Оценка	Сдал/пересдача
3	01012	3	
4	01013	5	
5	01014		
6	01015		
7	01016	5	
8	01017		
9	01018	4	
10	01019		
11	01020	5	
12	01021		
13	01022	4	
14	01023	4	
15	01024		
16	01025	3	
17	01026	4	
18	01027	5	

Выделим ячейки C3:C18 и запишем следующую формулу:

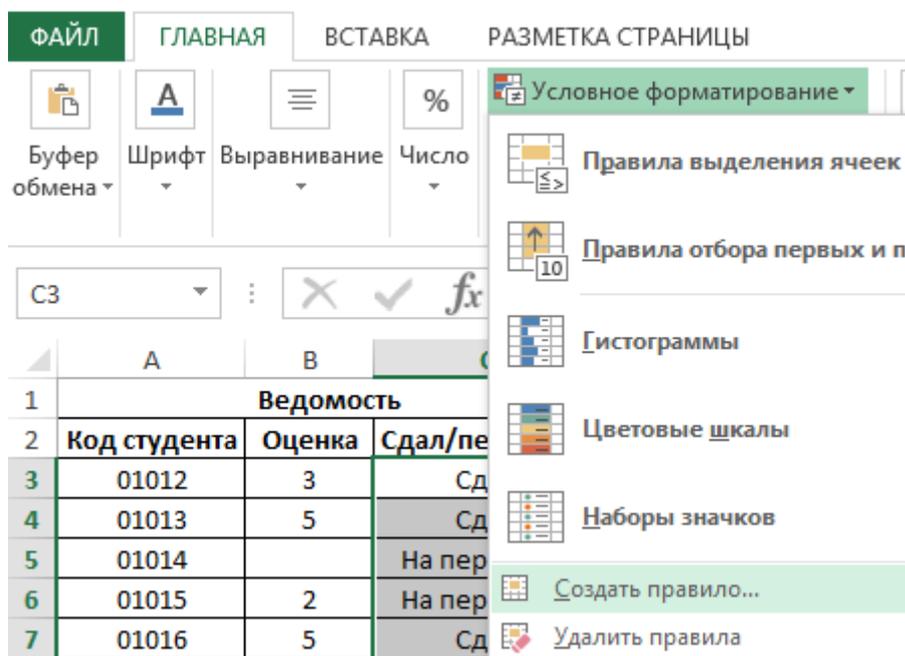
=ЕСЛИ(ИЛИ

Формула ЕСЛИ выполняет проверку возвращаемого результата функции ЕПУСТО для диапазона ячеек В3:В18 и возвращает один из вариантов ("На пересдачу" или "Сдал"). Результат выполнения функции:

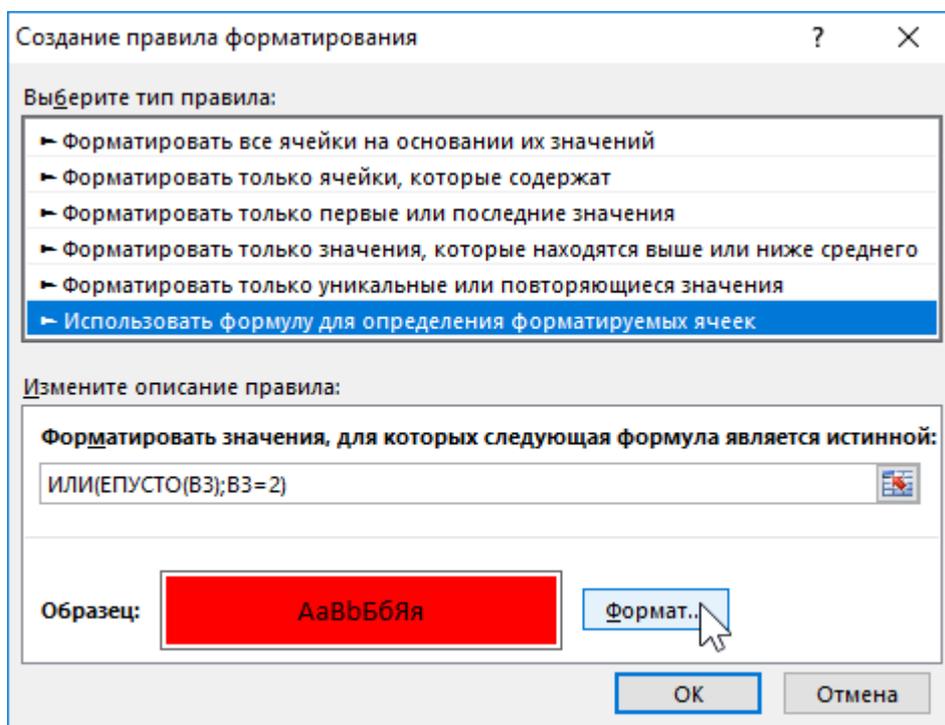
C3	:	<i>fx</i>	=ЕСЛИ(ИЛИ(ЕПУСТО(B3);B3=2);"На пересдачу";"Сдал")					
	A	B	C	D	E	F	G	H
1	Ведомость							
2	Код студента	Оценка	Сдал/пересдача					
3	01012	3	Сдал					
4	01013	5	Сдал					
5	01014		На пересдачу					
6	01015	2	На пересдачу					
7	01016	5	Сдал					
8	01017		На пересдачу					
9	01018	4	Сдал					
10	01019		На пересдачу					
11	01020	5	Сдал					
12	01021		На пересдачу					
13	01022	4	Сдал					
14	01023	4	Сдал					
15	01024	2	На пересдачу					
16	01025	3	Сдал					
17	01026	4	Сдал					
18	01027	5	Сдал					

Теперь часть данной формулы можно использовать для условного форматирования:

1. Выделите диапазон ячеек C3:C18 и выберите инструмент: «ГЛАВНАЯ»-«Стили»-«Условное форматирование»-«Создать правило».



2. В появившемся окне «Создание правила форматирования» выберите опцию: «Использовать формулы для определения форматируемых ячеек» и введите следующую формулу:



3. Нажмите на кнопку «Формат» (как на образцу), потом укажите в окне «Формат ячеек» красный цвет заливки и нажмите ОК на всех открытых окнах:

	А	В	С
1	Ведомость		
2	Код студента	Оценка	Сдал/пересдача
3	01012	3	Сдал
4	01013	5	Сдал
5	01014		На пересдачу
6	01015	2	На пересдачу
7	01016	5	Сдал
8	01017		На пересдачу
9	01018	4	Сдал
10	01019		На пересдачу
11	01020	5	Сдал
12	01021		На пересдачу
13	01022	4	Сдал
14	01023	4	Сдал
15	01024	2	На пересдачу
16	01025	3	Сдал
17	01026	4	Сдал
18	01027	5	Сдал

На против незаполненных (пустых) ячеек или двоек мы получаем соответственное сообщение «На пересдачу» и красную заливку.

ПОЧЕМУ НУЖНО ИСПОЛЬЗОВАТЬ ФУНКЦИЮ ЕПУСТО ПРИ ПРОВЕРКЕ ПУСТЫХ ЯЧЕЕК

У выше указанном примере можно изменить формулу используя двойные кавычки ("") в место функции проверки ячеек на пустоту, и она также будет работать:

=ЕСЛИ(ИЛИ(В3="";В3=2);"На пересдачу";"Сдал")

Но не всегда! Все зависит от значений, которые могут содержать ячейки. Обратите внимание на то как по-разному себя ведут двойные кавычки, и функция ЕПУСТО если у нас в ячейках находятся одни и те же специфические значения:

	A	B	C	D
1		ПУСТО	<-- =ЕСЛИ(A1="" ; "ПУСТО"; "НЕ ПУСТО")	
2		НЕ ПУСТО	<-- =ЕСЛИ(ЕПУСТО(A1); "ПУСТО"; "НЕ ПУСТО")	

Как видно на рисунке в ячейке находится символ одинарной кавычки. Первая формула (с двойными кавычками вместо функции) ее не видит. Более того в самой ячейке A1 одинарная кавычка не отображается так как данный спецсимвол в Excel предназначенный для отображения значений в текстовом формате. Это удобно, например, когда нам нужно отобразить саму формулу, а не результат ее вычисления как сделано в ячейках D1 и D2. Достаточно лишь перед формулой ввести одинарную кавычку и теперь отображается сама формула, а не возвращаемый ее результат. Но функция ЕПУСТО видит, что в действительности ячейка A1 не является пустой!

ПРОВЕРКА НА ПУСТУЮ ЯЧЕЙКУ В ТАБЛИЦЕ EXCEL

Пример 2. В таблице Excel записаны некоторые данные. Определить, все ли поля заполнены, или есть как минимум одно поле, которое является пустым.

Исходная таблица данных:

	А	В
1	Данные	
2	Переменная	Значение
3	Var_1	2
4	Var_2	text
5	Var_3	ИСТИНА
6	Var_4	
7	Var_5	0
8	Var_6	12
9	Var_7	"
10	Var_8	123
11	Var_9	12.07.2018
12	Var_10	16:35
13	Var_11	
14	Var_12	имя
15	Var_13	12%
16	Var_14	12 000 ₽
17	Var_15	

Чтобы определить наличие пустых ячеек используем следующую формулу массива (CTRL+SHIFT+Enter):

`=ЕСЛИ(СУМ`

Функция СУММ используется для определения суммы величин, возвращаемых функцией --ЕПУСТО для каждой ячейки из диапазона В3:В17 (числовых значений, поскольку используется двойное отрицание). Если запись СУММ(--ЕПУСТО(В3:В17)) возвращает любое значение >0, функция ЕСЛИ вернет значение ИСТИНА.

Результат вычислений:

B19		={ЕСЛИ(СУММ(--ЕПУСТО(B3:B17));ИСТИНА;ЛОЖЬ)}						
	A	B	C	D	E	F	G	H
1	Данные							
2	Переменная	Значение						
3	Var_1	2						
4	Var_2	text						
5	Var_3	ИСТИНА						
6	Var_4							
7	Var_5	0						
8	Var_6	12						
9	Var_7	"						
10	Var_8	123						
11	Var_9	12.07.2018						
12	Var_10	16:35						
13	Var_11							
14	Var_12	имя						
15	Var_13	12%						
16	Var_14	12 000 Р						
17	Var_15							
18								
19	Пустые поля?	ИСТИНА						

То есть, в диапазоне B3:B17 есть одна или несколько пустых ячеек.

Примечание: в указанной выше формуле были использованы символы «--». Данный вид записи называется двойным отрицанием. В данном случае двойное отрицание необходимо для явного преобразования данных логического типа к числовому. Некоторые функции Excel не выполняют автоматического преобразования данных, поэтому механизм преобразования типов приходится запускать вручную. Наиболее распространенными вариантами преобразования текстовых или логических значений к числовому типу является умножение на 1 или добавление 0 (например, =ИСТИНА+0 вернет число 1, или =«23»*1 вернет число 23. Однако использование записи типа =--ИСТИНА ускоряет работу функций (по некоторым оценкам прирост производительности составляет до 15%, что имеет значение при обработке больших объемов данных).

КАК ПОСЧИТАТЬ КОЛИЧЕСТВО ПУСТЫХ ЯЧЕЕК В EXCEL

Пример 3. Рассчитать средний возраст работников офиса. Если в таблице заполнены не все поля, вывести соответствующее сообщение и не выполнять расчет.

Таблица данных:

	A	B
1	Работники	
2	Код работника	Возраст
3	101	23
4	102	35
5	103	55
6	104	28
7	105	
8	106	34
9	107	
10	108	55
11	109	
12	110	52

Формула для расчета (формула массива):

=ЕСЛИ(СУМ

Функция ЕСЛИ выполняет проверку диапазона на наличие пустых ячеек (выражение СУММ(--ЕПУСТО(B3:B12))). Если СУММ вернула значение >0, будет выведено сообщение, содержащее количество незаполненных данными ячеек (СЧИТАТЬПУСТОТЫ) и строку «поля не заполнены», которые склеены знаком «&» (операция конкатенации).

Результат вычислений:

B14 : ✕ ✓ fx
{=ЕСЛИ(СУММ(--ЕПУСТО(B3:B12));СЧИТАТЬПУСТОТЫ(B3:B12)&" поля не заполнены";СРЗНАЧ(B3:B12))}

	A	B	C	D	E	F	G
1	Работники						
2	Код работника	Возраст					
3	101	23					
4	102	35					
5	103	55					
6	104	28					
7	105						
8	106	34					
9	107						
10	108	55					
11	109						
12	110	52					
13							
14	Средний возраст:	3 поля не заполнены					

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ФУНКЦИИ ЕПУСТО В EXCEL

Функция ЕПУСТО в Excel относится к числу логических функций (выполняющих проверку какого-либо условия, например, ЕСЛИ, ЕССЫЛКА, ЕЧИСЛО и др., и возвращающих результаты в виде данных логического типа: ИСТИНА, ЛОЖЬ). Синтаксическая запись функции:

=ЕПУСТО(значение)

Единственный аргумент является обязательным для заполнения и может принимать ссылку на ячейку или на диапазон ячеек, в которых необходимо определить наличие каких-либо данных. Если функция принимает диапазон ячеек, функция должна быть использована в качестве формулы массива.

Примечания:

1. Если в качестве аргумента функции было явно передано какое-либо значение (например, =ЕПУСТО(ИСТИНА), =ЕПУСТО(«текст»), =ЕПУСТО(12)), результат ее выполнения – значение ЛОЖЬ.
2. Если требуется, чтобы функция возвращала значение ИСТИНА, если ячейка не является пустой, ее можно использовать совместно с функцией НЕ. Например, =НЕ(ЕПУСТО(A1)) вернет ИСТИНА, если A1 не является пустой.
3. Запись типа =ЕПУСТО(АДРЕС(х;у)) всегда будет возвращать значение ложь, поскольку функция АДРЕС(х;у) возвращает ссылку на ячейку, то есть непустое значение.
4. Функция возвращает значение ЛОЖЬ даже в тех случаях, когда в переданной в качестве аргумента ячейке содержится ошибка или ссылка на ячейку. Это суждение справедливо и для случаев, когда в результате выполнения какой-либо функции в ячейку была выведена пустая строка. Например, в ячейку A1 была введена формула =ЕСЛИ(2>1;"";ЛОЖЬ), которая вернет пустую строку «». В этом случае функция =ЕПУСТО(A1) вернет значение ЛОЖЬ.
5. Если требуется проверить сразу несколько ячеек, можно использовать функцию в качестве формулы массива (выделить требуемое количество пустых ячеек, ввести формулу «=ЕПУСТО(» и в качестве аргумента передать диапазон исследуемых ячеек, для выполнения использовать комбинацию клавиш Ctrl+Shift+Enter)

Использование функции ОКРУГЛ

Пример 1. Стоимость валюты составляет 70 рублей. Определить наибольшее количество валюты, которое можно приобрести, если покупатель

взял с собой 2200 рублей. Как известно, в обменных пунктах выдают только купюры.

Исходные данные:

	A	B
1	Покупка валюты	
2	Курс руб/доллар	70,00 ₺
3	Рубли в наличии	2 200,00 ₺
4	Сумма обмена, USD	?

Для решения используем следующую формулу:

=ЕСЛИ(ОКР

Функция ОКРУГЛ принимает следующие параметры:

- B3/B2 – частное от деления суммы имеющихся рублей на стоимость одной валюты;
- 0 – значение, указывающее на то, что округление будет выполняться до целого числа.

Если произведение округленного значения и курса валюты превышает сумму имеющихся рублей, то из первого будет вычтена единица, иначе будет возвращено просто округленное значение.

Результат расчетов:

В4 :  =ЕСЛИ(ОКРУГЛ(B3/B2;0)*B2>B3; ОКРУГЛ(B3/B2;0)-1;ОКРУГЛ(B3/B2;0))

	A	B	C	D	E	F
1	Покупка валюты					
2	Курс руб/доллар	70,00 ₺				
3	Рубли в наличии	2 200,00 ₺				
4	Сумма обмена, USD	\$ 31,00				

То есть, 2200 рублей можно поменять не более чем на 31 у.е.

КАК ПЕРЕВЕСТИ МИЛИ В КИЛОМЕТРЫ ИСПОЛЬЗУЯ ФУНКЦИЮ ОКРУГЛ В EXCEL

Пример 2. Спидометр автомобиля отображает скорость в единицах измерения мили/ч. 1 миля = 1609 м (1,609 км). Определить скорость автомобиля в километрах часов (округленно до одного знака после запятой), если на данный момент спидометр показывает скорость 75 миль/ч.

Таблица данных:

	A	B
1	Перевод мили/ч в км/ч	
2	1 миля = километров	1,609
3	Текущая скорость миль	75
4	Скорость, км/ч	?

Формула для решения:

=ОКРУГЛ(В3*В2;1)

Описание аргументов:

- В3*В2 – перевод миль/ч в км/ч;
- 1 – количество знаков после запятой для округления.

Полученный результат:

В4 :    =ОКРУГЛ(В3*В2;1)

	A	B	C	D
1	Перевод мили/ч в км/ч			
2	1 миля = километров	1,609		
3	Текущая скорость миль	75		
4	Скорость, км/ч	120,7		

КАЛЬКУЛЯТОР РАСЧЕТА МОЩНОСТИ КОНДИЦИОНЕРА ПО ПЛОЩАДИ КОМНАТЫ В EXCEL

Пример 3. Для помещения площадью 60 кв. м и высотой потолка 2,7 м необходимо подобрать кондиционер по мощности. Поставщики предлагают кондиционеры мощностью от 2 кВт с шагом 0,5 кВт. Определить подходящий кондиционер.

Таблица данных:

	A	B
1	Расчет кондиционера	
2	Площадь	60
3	Высота потолка	2,7
4	Коэффициент q (теплопритоки)	35
5	Мощность кондиционера	?

Искомая мощность рассчитывается как произведение площади, высоты потолка и коэффициента q. Используем следующую формулу:

=В2*В3*В4/1000

B5		: X ✓ fx		=B2*B3*B4/1000
	A	B	C	
1	Расчет кондиционера			
2	Площадь	60		
3	Высота потолка	2,7		
4	Коэффициент q (теплопритоки)	35		
5	Мощность кондиционера	5,67		

Для автоматического выбора кондиционера используем формулу:

=ЕСЛИ(ИЛИ

В данном случае функция ЕСЛИ выполняет проверку дробной части найденного значения мощности на принадлежность к промежутку значений от 0 до 0,25 и от 0,5 до 0,75. Это необходимо для правильного выбора кондиционера по мощности. Например, если бы результатом расчетов являлась мощность 5,2 кВт, функция ОКРУГЛТ(5,2;0,5) вернула бы значение 5. Однако мощность выбранного кондиционера должна быть равной или больше расчетной. Поэтому в этом случае к результату будет добавлено значение 0,5.

Результат расчета мощности кондиционера по площади комнаты:

B6		: X ✓ fx		=ЕСЛИ(ИЛИ(И(B5-ЦЕЛОЕ(B5)>0;B5-ЦЕЛОЕ(B5)<0,25);И(B5-ЦЕЛОЕ(B5)>0,5;B5-ЦЕЛОЕ(B5)<0,75));ОКРУГЛТ(B5;0,5)+0,5;ОКРУГЛТ(B5;0,5))				
	A	B	C	D	E	F	G	H
1	Расчет кондиционера							
2	Площадь	60						
3	Высота потолка	2,7						
4	Коэффициент q (теплопритоки)	35						
5	Мощность кондиционера	5,67						
6	Выбранный кондиционер	6						

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ФУНКЦИЙ ОКРУГЛ И ОКРУГЛТ В EXCEL

Обе функции используются для округления числовых значений наряду с прочими функциями ОКРУГЛВНИЗ, ОКРУГЛВВЕРХ. Функция ОКРУГЛ имеет следующую синтаксическую запись:

=ОКРУГЛ(число;число_разрядов)

Описание аргументов:

- число – обязательный аргумент, характеризующий числовое значение, для которого требуется выполнение операции округления;

- число_разрядов – обязательный аргумент, характеризующий количество знаков после запятой в числе после округления.

Функция ОКРУГЛТ имеет следующий синтаксис:

=ОКРУГЛТ(число;точность)

Описание аргументов:

- число – обязательный аргумент, соответствует одноименному аргументу функции ОКРУГЛ;
- точность – обязательный аргумент, характеризующий числовое значение точности округления числа¹¹.

Использование функции СУММЕСЛИ

Функция СУММЕСЛИ позволяет суммировать ячейки, которые удовлетворяют определенному критерию (заданному условию). Аргументы команды следующие:

1. Диапазон – ячейки, которые следует оценить на основании критерия (заданного условия).
2. Критерий – определяет, какие ячейки из диапазона будут выбраны (записывается в кавычках).
3. Диапазон суммирования – фактические ячейки, которые необходимо просуммировать, если они удовлетворяют критерию.

Получается, что у функции всего 3 аргумента. Но иногда последний может быть исключен, и тогда команда будет работать только по диапазону и критерию.

КАК РАБОТАЕТ ФУНКЦИЯ СУММЕСЛИ В EXCEL?

Рассмотрим простейший пример, который наглядно продемонстрирует, как использовать функцию СУММЕСЛИ и насколько удобной она может оказаться при решении определенных задач.

Имеем таблицу, в которой указаны фамилии сотрудников, их пол и зарплата, начисленная за январь-месяц. Если нам нужно просто посчитать общее количество денег, которые требуется выдать работникам, мы используем функцию СУММ, указав диапазоном все заработные платы.

¹¹ <https://exceltable.com/funkcii-excel/primery-funkcij-okrgul-i-okruglt>

Но как быть, если нам нужно быстро посчитать заработные платы только продавцов? В дело вступает использование функции СУММЕСЛИ.

Прописываем аргументы.

1. Диапазоном в данном случае будет являться список всех должностей сотрудников, потому что нам нужно будет определить сумму заработных плат. Поэтому проставляем E2:E14.
2. Критерий выбора в нашем случае – продавец. Заключаем слово в кавычки и ставим вторым аргументом.
3. Диапазон суммирования – это заработные платы, потому что нам нужно узнать сумму зарплат всех продавцов. Поэтому F2:F14.

F17 fx =СУММЕСЛИ(E2:E14;"продавец";F2:F14)

	А	В	С	Д	Е	Ф
1	Фамилия	Имя	Отчество	Пол	Должность	З/п за январь, руб
2	Акимов	Андрей	Денисович	м	продавец	24200
3	Белова	Людмила	Матвеевна	ж	продавец	22500
4	Беляев	Денис	Олегович	м	менеджер	28400
5	Гусева	Ирина	Валерьевна	ж	продавец-кассир	23600
6	Иванов	Иван	Иванович	м	менеджер	25700
7	Карпова	Евгения	Сергеевна	ж	продавец-кассир	24100
8	Макимова	Анна	Дмитриевна	ж	уборщица	12000
9	Морозов	Василий	Михайлович	м	директор	37400
10	Петров	Петр	Петрович	м	охранник	18400
11	Пронина	Мария	Павловна	ж	продавец	28600
12	Романова	Ольга	Ивановна	ж	менеджер	30200
13	Семенов	Павел	Андреевич	м	продавец	17600
14	Сидоров	Олег	Семенович	м	охранник	18400
15						
16					ИТОГО:	311100
17					З/п всех продавцов	92900
18					З/п всех менеджеров	
19					З/п всех продавцов-кассиров	
20					З/п всех охранников	

Получилось 92900. Т.е. функция автоматически проработала список должностей, выбрала из них только продавцов и просуммировала их зарплаты.

Аналогично можно подсчитать зарплаты всех менеджеров, продавцов-кассиров и охранников. Когда табличка небольшая, кажется, что все можно сосчитать и вручную, но при работе со списками, в которых по несколько сотен позиций, целесообразно использовать СУММЕСЛИ.

Использование функции СЧЁТЕСЛИ

Посчитаем числовые значения в одном диапазоне. Условие подсчета – один критерий.

У нас есть такая таблица:

	А	В
1	Стол	124
2	Стул	235
3	Табурет	380
4	Кроват	100
5	Шкаф 1	59
6	Стол	87
7	Табурет	210
8	Шкаф 2	241
9	Тумба	386
10	Стул	45
11	Банкет	87

Посчитаем количество ячеек с числами больше 100. Формула: =СЧЁТЕСЛИ(В1:В11;">100"). Диапазон – В1:В11. Критерий подсчета – «>100». Результат:

fx =СЧЁТЕСЛИ(В1:В11;">100")		
	D	E
>100		6

Если условие подсчета внести в отдельную ячейку, можно в качестве критерия использовать ссылку:

fx =СЧЁТЕСЛИ(В1:В11;D2)		
	D	E
>100		6

Посчитаем текстовые значения в одном диапазоне. Условие поиска – один критерий.

Формула: =СЧЁТЕСЛИ(А1:А11;"табурет"). Или:

fx =СЧЁТЕСЛИ(А1:А11;D3)		
	D	E
>100		6
табурет		2

Во втором случае в качестве критерия использовали ссылку на ячейку.

Формула с применением знака подстановки: =СЧЁТЕСЛИ(A1:A11;"таб*").

Для расчета количества значений, оканчивающихся на «и», в которых содержится любое число знаков: =СЧЁТЕСЛИ(A1:A11;"*и"). Получаем:

	D	E	F
>100		6	
табуреты		2	
*и		2	

Формула посчитала «кровати» и «банкетки».

Используем в функции СЧЕТЕСЛИ условие поиска «не равно».

Формула: =СЧЁТЕСЛИ(A1:A11;"<>"&"стулья"). Оператор «<>» означает «не равно». Знак амперсанда (&) объединяет данный оператор и значение «стулья».

	D	E	F	G
>100		6		
табуреты		2		
*и		2		
<>стулья		9		

При применении ссылки формула будет выглядеть так:

	D	E	F
>100		6	
табуреты		2	
*и		2	
<>стулья		9	
100		10	

Часто требуется выполнять функцию СЧЕТЕСЛИ в Excel по двум критериям. Таким способом можно существенно расширить ее возможности. Рассмотрим специальные случаи применения СЧЕТЕСЛИ в Excel и примеры с двумя условиями.

1. Посчитаем, сколько ячеек содержат текст «столы» и «стулья». Формула: =СЧЁТЕСЛИ(A1:A11;"столы")+СЧЁТЕСЛИ(A1:A11;"стулья"). Для указания нескольких условий используется несколько выражений СЧЕТЕСЛИ. Они объединены между собой оператором «+».

	A	B	C	D	E	F
1	Стол	124				
2	Стул	235		4		
3	Табуреты	380				
4	Кровати	100				
5	Шкафы 1	59				
6	Стол	87				
7	Табуреты	210				
8	Шкафы 2	241				
9	Тумбы	386				
10	Стул	45				
11	Банкетки	87				

2. Условия – ссылки на ячейки. Формула: =СЧЁТЕСЛИ(A1:A11;A1)+СЧЁТЕСЛИ(A1:A11;A2). Текст «столы» функция ищет в ячейке A1. Текст «стулья» - на базе критерия в ячейке A2.

	A	B	C	D	E
1	Стол	124			
2	Стул	235		4	
3	Табуреты	380			
4	Кровати	100			
5	Шкафы 1	59			
6	Стол	87			
7	Табуреты	210			
8	Шкафы 2	241			
9	Тумбы	386			
10	Стул	45			
11	Банкетки	87			

3. Посчитаем число ячеек в диапазоне В1:В11 со значением большим или равным 100 и меньшим или равным 200. Формула: =СЧЁТЕСЛИ(В1:В11;">=100")-СЧЁТЕСЛИ(В1:В11;">200").

fx		=СЧЁТЕСЛИ(В1:В11;">=100")-СЧЁТЕСЛИ(В1:В11;">200")				
	А	В	С	Д	Е	F
1	Стол	124				
2	Стол	235		2		
3	Табуреты	380				
4	Кровати	100				
5	Шкафы 1	59				
6	Стол	87				
7	Табуреты	210				
8	Шкафы 2	241				
9	Тумбы	386				
10	Стол	45				
11	Банкетки	87				

4. Применим в формуле СЧЕТЕСЛИ несколько диапазонов. Это возможно, если диапазоны являются смежными. Формула: =СЧЁТЕСЛИ(А1:В11;">=100")-СЧЁТЕСЛИ(А1:В11;">200"). Ищет значения по двум критериям сразу в двух столбцах. Если диапазоны несмежные, то применяется функция СЧЕТЕСЛИМН.

fx		=СЧЁТЕСЛИ(А1:В11;">=100")-СЧЁТЕСЛИ(А1:В11;">200")				
	А	В	С	Д	Е	F
1	Стол	124				
2	Стол	235		2		
3	Табуреты	380				
4	Кровати	100				

5. Когда в качестве критерия указывается ссылка на диапазон ячеек с условиями, функция возвращает массив. Для ввода формулы нужно выделить такое количество ячеек, как в диапазоне с критериями. После введения аргументов нажать одновременно сочетание клавиш Shift + Ctrl + Enter. Excel распознает формулу массива.

fx {=СЧЁТЕСЛИ(A1:A11;D2:D4)}		
С	Д	Е
	Условия	Результат
	столы	2
	стулья	2
	табуреты	2

СЧЕТЕСЛИ с двумя условиями в Excel очень часто используется для автоматизированной и эффективной работы с данными. Поэтому продвинутому пользователю настоятельно рекомендуется внимательно изучить все приведенные выше примеры.

ПРОМЕЖУТОЧНЫЕ.ИТОГИ И СЧЕТЕСЛИ

Посчитаем количество реализованных товаров по группам.

1. Сначала отсортируем таблицу так, чтобы одинаковые значения оказались рядом.

	А	В
1	Банкетки	87
2	Кровати	100
3	Стол	124
4	Стол	87
5	Стуль	235
6	Стуль	45
7	Табуреты	380
8	Табуреты	210
9	Тумбы	386
10	Шкафы 1	59
11	Шкафы 2	241

2. Первый аргумент формулы «ПРОМЕЖУТОЧНЫЕ.ИТОГИ» - «Номер функции». Это числа от 1 до 11, указывающие статистическую функцию для расчета промежуточного результата. Подсчет количества ячеек осуществляется под цифрой «2» (функция «СЧЕТ»).

	A	B	C	D
1	Банкетки	87		
2	Кровати	100		2
3	Стол	124		
4	Стол	87		
5	Стулья	235		
6	Стулья	45		
7	Табуреты	380		
8	Табуреты	210		
9	Тумбы	386		
10	Шкафы 1	59		
11	Шкафы 2	241		

Формула нашла количество значений для группы «Стулья». При большом числе строк (больше тысячи) подобное сочетание функций может оказаться полезным¹².

Практическая работа № 6. Базы данных.

Цель: изучение основных этапов создания базы данных.

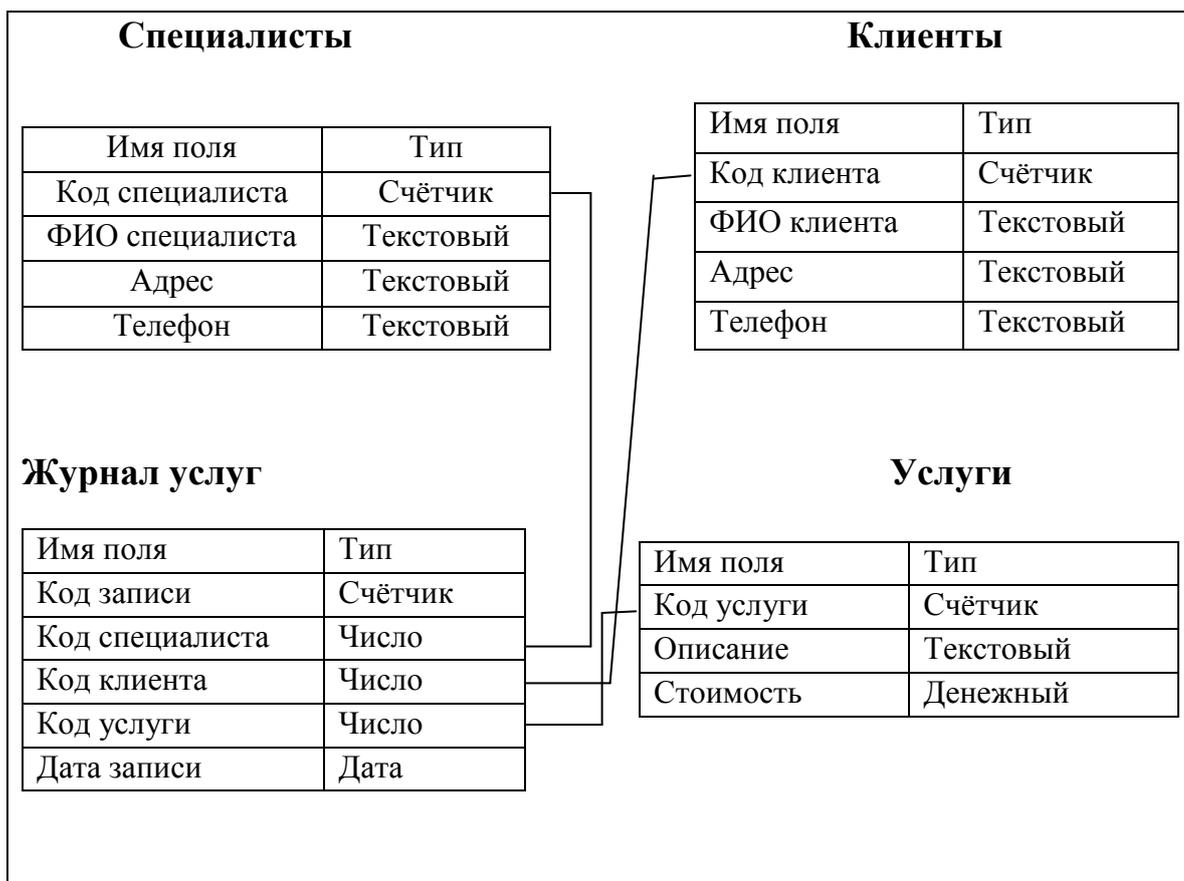
Практическая работа №1. Создание базы данных.

Требуется создать файл базы данных для предприятия «Автосервис», которое занимается оказанием услуг по ремонту автомобилей. На предприятии работают специалисты, каждый из которых имеет свой профиль по ремонту, и необходимо учесть их работу. У предприятия складывается своя клиентура и нужно иметь данные на каждого клиента. Так же необходимо вести систематический учёт оказываемых услуг. Файл базы данных должен допускать возможность получения с помощью запросов информации о выполненной конкретным специалистом работе, об услугах конкретному клиенту, оказании конкретных услуг¹³.

1. Создайте файл базы данных с именем «Автосервис».
2. В файле базы данных «Автосервис» создайте таблицы с полями согласно образцу:

¹² <https://exceltable.com/funkcii-excel/funkciya-schetesli-primery>

¹³ Соболев, Б. В. Практикум по информатике: / Б. В. Соболев [и др.]; под ред. Б.В. Соболя. – Ростов н / Д: Феникс, 2009. – 509 [1] с. – (Высшее образование).



3. Введите произвольные данные в эти таблицы.
4. Запустить программу MS Access и выбрать пункт меню «Создать» и «Новая база данных». Появится окно **Файл новой базы данных**. Выбрать папку, в которой необходимо сохранять файл, ввести имя **Автосервис** и нажать кнопку **Создать**.
5. Выбрать объекты **Таблицы** и **Создание в режиме конструктора**. В появившемся окне задать имена и типы данных для полей. При этом в нижней части окна задать **Свойства поля**. После указания имён и типов полей щёлкнуть правой кнопкой мыши поле **Код специалиста** (в первой клетке этой строки) и в контекстном меню выбрать **Ключевое поле**. Затем закрыть окно Microsoft Access- [Таблица1: таблица], на запрос программы **Сохранить изменения?** выбрать **Да**, и в окне **Сохранить как** задать имя таблицы **Специалисты**.
6. Аналогично создать таблицы **Клиенты**, **Услуги** и **Журнал услуг**.
7. Для создания связей выполнить команду **Схема данных**, в котором откроется окно **Добавить таблицу**.
8. Ввести данные в таблицы. Для этого достаточно открыть таблицу и ввести непосредственно в строки таблицы исходные данные.

Практическая работа №2. Создание реляционной базы данных в Microsoft Access. Запросы, формы.

Первый запрос «Сведения о работе специалиста» должен запросить код специалиста, выдать фамилию специалиста, которому соответствует введённый код, выдать сведения о выполненных данным специалистом услугах и фамилию клиента.

1. Выполните команду *Создание запроса в режиме конструктора*, в окне **Добавить таблицу** выбрать все четыре таблицы.

В окне **Запрос1**: запрос на выборку двойным щелчком левой кнопкой мыши выбрать следующие поля: **Код специалиста** из таблицы Журнал услуг, **ФИО специалиста** из таблицы Специалисты, **Описание** из таблицы Услуги, **ФИО клиента** из таблицы Клиенты.

Для поля Код специалиста снять флажок в строке Вывод на экран, а в строке Условие отбора ввести [Введите Код специалиста].

Теперь закрыть окно Запрос1: запрос на выборку и сохранить запрос под именем Сведения о работе специалиста.

Открыть созданный запрос. Для этого выполнить двойной щелчок указателем мыши на имени запроса. Появится окно Введите значение параметра.

Задание. Создайте отчёт для таблицы Клиенты, в который вывести фамилии и номера телефонов клиентов.

Задание. Создайте отчёт для запроса Сведения о работе специалиста.

1. Открыть файл базы данных Автосервис, выбрать закладку Отчёты и щёлкнуть строку Создание отчёта с помощью мастера. В окне Создание отчётов выбрать таблицу Клиенты, а затем с помощью кнопки со знаком «>» выбрать поля ФИО, Телефон, включаемые в отчёт, и нажать кнопку Готово. Получим документ для печати, который содержит два столбца, соответствующих выбранным полям.
2. Открыть файл базы данных Автосервис, выбрать закладку Отчёты и щёлкнуть строку Создание отчёта с помощью мастера. В окне Создание отчётов выбрать запрос Сведения о работе специалиста и выбрать все поля с помощью кнопки со знаком «>>». Нажать кнопку Готово. Так запрос запрашивает ввод параметра, то при формировании отчёта также будет запрашиваться значение параметра Код специалиста.

Отображение данных в форме.

Создайте форму **Просмотр и редактирование**, с помощью которой можно просматривать полные сведения об оказанных услугах и добавлять соответствующие записи в таблицы при оформлении новой услуги.

1. Открыть файл базы данных Автосервис, выбрать вкладку **Формы** и щёлкнуть строку **Создание формы с помощью Мастера**.
2. В окне **Создание формы** выбрать из каждой таблицы все поля, кроме поля типа **Счётчик**, в следующей последовательности:

- Код специалиста из таблицы журнал услуг;
- ФИО специалиста, Адрес и Телефон из таблицы Специалисты;
- Код клиента из таблицы Журнал услуг;
- ФИО клиента, Адрес и Телефон из таблицы Клиенты;
- Код услуги из таблицы Журнал услуг;
- Описание и Стоимость из таблицы Услуги;
- Дата обслуживания из таблицы Журнал услуг. Сохранить форму под именем Просмотр и редактирование.

Открыть форму **Просмотр и редактирование** и пролистать записи, пользуясь кнопками в нижней части окна. Если необходимо внести изменения в записях формы, то соответствующие изменения будут внесены в таблицы.

С помощью данной формы можно также добавлять сведения о новой услуге.

2 семестр

Практическая работа №1. Электронные презентации. MS Power Point.

Цель: научиться создавать простую презентацию.

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

Слайд – это одна страница визуального материала вне зависимости от того, куда она будет направлена – на экран дисплея, принтер или фотопленку.

1. Запустите программу MS Power Point.
2. Заполните титульный лист презентации: заголовок-Мини-пекарня «Смак», подзаголовок -Бизнес -План.
3. Создайте новый слайд презентации с заголовком Содержание презентации. Пункты списка содержания: Ассортимент выпускаемой продукции, Расходы на рекламу, Арендуемые помещения, Прогнозируемые объёмы продаж.
4. Создайте новый слайд Ассортимент выпускаемой продукции. Пункты списка продолжите в две колонки. Для этого необходимо изменить макет слайда: Формат-Разметка слайда-Макеты текста – Заголовок и текст в две колонки. Пункты: Хлеб «Ржаной», хлеб «Пшеничный», хлеб «Бородинский», Хлеб с тмином; Батон «Дорожный», Плетёнка, Булка «Сладкая», Бублик «Московский».

5. Создайте новый слайд Расходы на рекламу. Разместите на нём диаграмму Вставка/Диаграмма. Выберите тип диаграммы. Появится окно программы Excel. Удалите все строки из таблицы и занесите следующие значения:

	1 кв	2 кв	3 кв	4 кв
Наружная реклама	18	28	90	18
Реклама на радио	28	38	32	27
Реклама в прессе	53	48	42	42

6. Создайте новый слайд Арендные помещения. Разместите на слайде таблицу: Вставка/Таблица. Занесите следующие данные:

Виды помещения	Площадь	Стоимость аренды на единицу площади	Сумма арендной платы за месяц
Производственные	150	5	750
Подсобные	50	5	250
Подвальные	200	2	400
Всего	400	-	1400

7. Создайте новый слайд Прогнозируемые объёмы продаж. Разместите на нём график: Формат-Разметка слайда- Макеты содержимого -Заголовок и объект. В центре слайда выберите значок Добавление диаграммы. Тип диаграммы выберите График и нажмите ОК. Появится окно программы Excel. Удалите все строки из таблицы и занесите следующие:

	Январь	Февраль	Март
Хлеб «Ржаной»	10000	12000	11000
Хлеб «Пшеничный»	7000	7500	8000
Хлеб с тмином	5000	7000	9000

Измените параметры графика (Контекстное меню-Параметры диаграммы). На вкладке Заголовки: Ось X-месяцы, Ось Y- объём продаж. На вкладке Легенда: снять флажок Добавить легенду. На вкладке Таблица данных: поставить флажок Таблица данных. На вкладке Подписи данных: поставить флажок Значения.

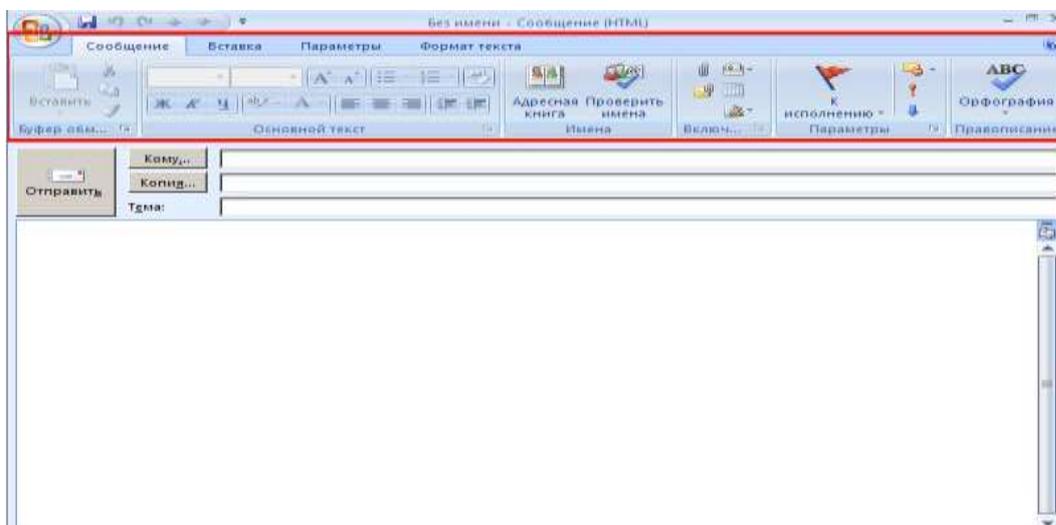
8. Перейдите в режим предварительного просмотра слайдов. Если необходимо, отредактируйте слайды.
9. Сохраните презентацию: Файл-Сохранить как...

Практическая работа №2. Сети.

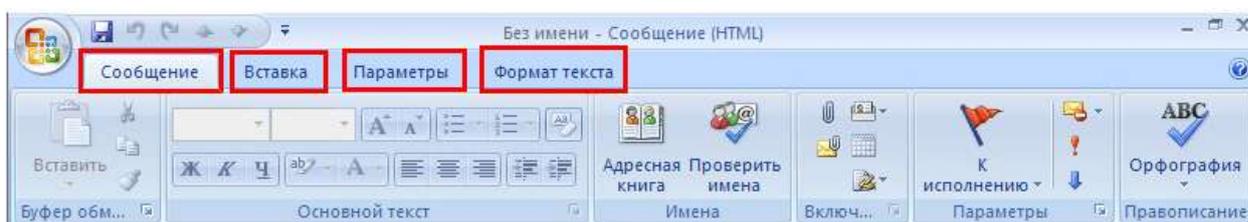
Цель: научиться создавать сообщения в программе Outlook.

Для создания нового исходящего сообщения электронной почты нажмите кнопку «Создать». Для создания сообщений в Outlook используется редактор

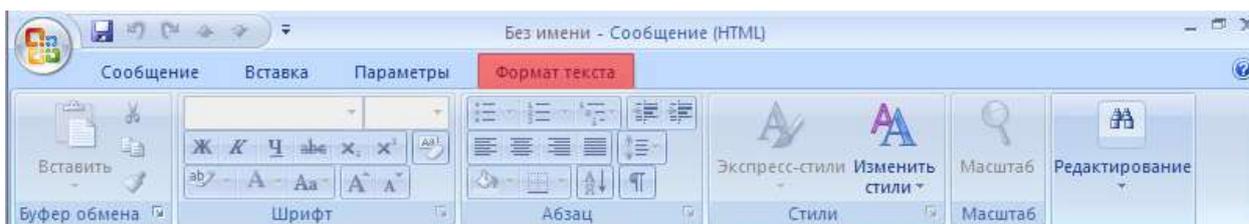
на основе Microsoft Office Word. Как и в большинстве программ, входящих в состав Microsoft Office, в основе интерфейса окна редактирования лежит «лента» - многостраничная область, расположенная в верхней части главного окна.



Каждая страница «ленты» содержит набор логически связанных управляющих элементов. На странице «Сообщение» собраны основные инструменты, используемые для составления сообщений.



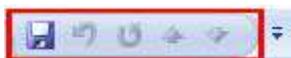
Некоторые команды могут быть продублированы и на других страницах «ленты». На странице «Вставка» находятся команды, с помощью которых можно присоединить к сообщению файл или один из элементов Outlook, вставить в тело письма рисунок, таблицу, гиперссылку и так далее. На странице «Параметры» собраны инструменты настройки параметров, относящихся к сообщению в целом. Это общий стиль и формат документа, запрос уведомления о доставке или прочтении, место для хранения сообщения и так далее. Управляющие элементы, расположенные на странице «Формат текста», используются для форматирования отдельных фрагментов текста сообщения. Для экономии экранного пространства область «ленты», занятую кнопками, можно скрыть. Выполните двойной щелчок на заголовке активной вкладки.



В таком режиме отображаются только закладки страниц. Щелкните один раз на закладке, чтобы развернуть соответствующую страницу.



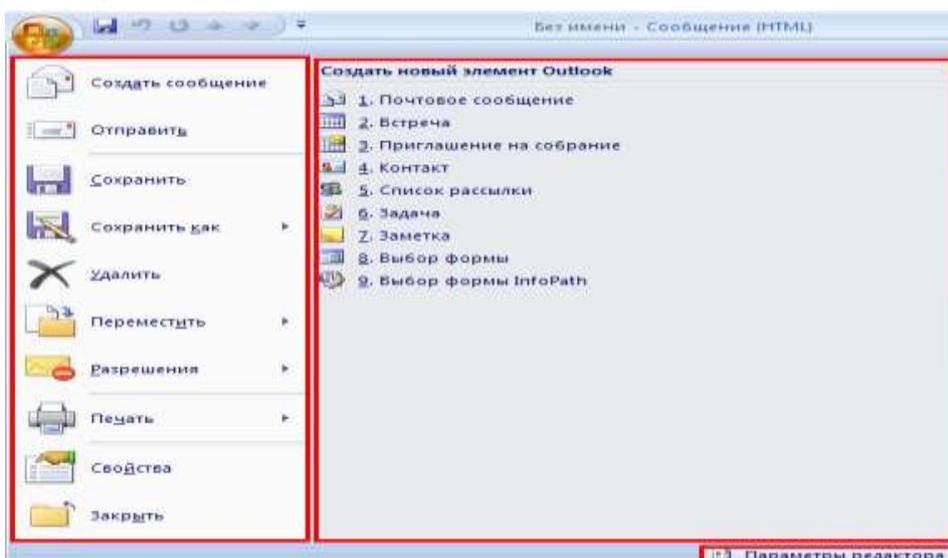
Щелкните за пределами «ленты», чтобы снова свернуть страницу. Дважды щелкните на любой закладке, чтобы вернуть «ленту» в исходное состояние. В верхней части окна редактирования сообщения находится панель быстрого доступа, предназначенная для вызова наиболее часто используемых команд.



В числе таких команд может быть быстрое сохранение сообщения, отмена последнего действия и так далее.

Для доступа к основным файловым операциям и настройкам редактора нажмите кнопку «Office» («Офис»), расположенную в левом верхнем углу окна сообщения.

В левой части открывшегося окна диалога собраны команды создания, загрузки и сохранения сообщений, отправки их по электронной почте или на печать, редактирования свойств и другие.

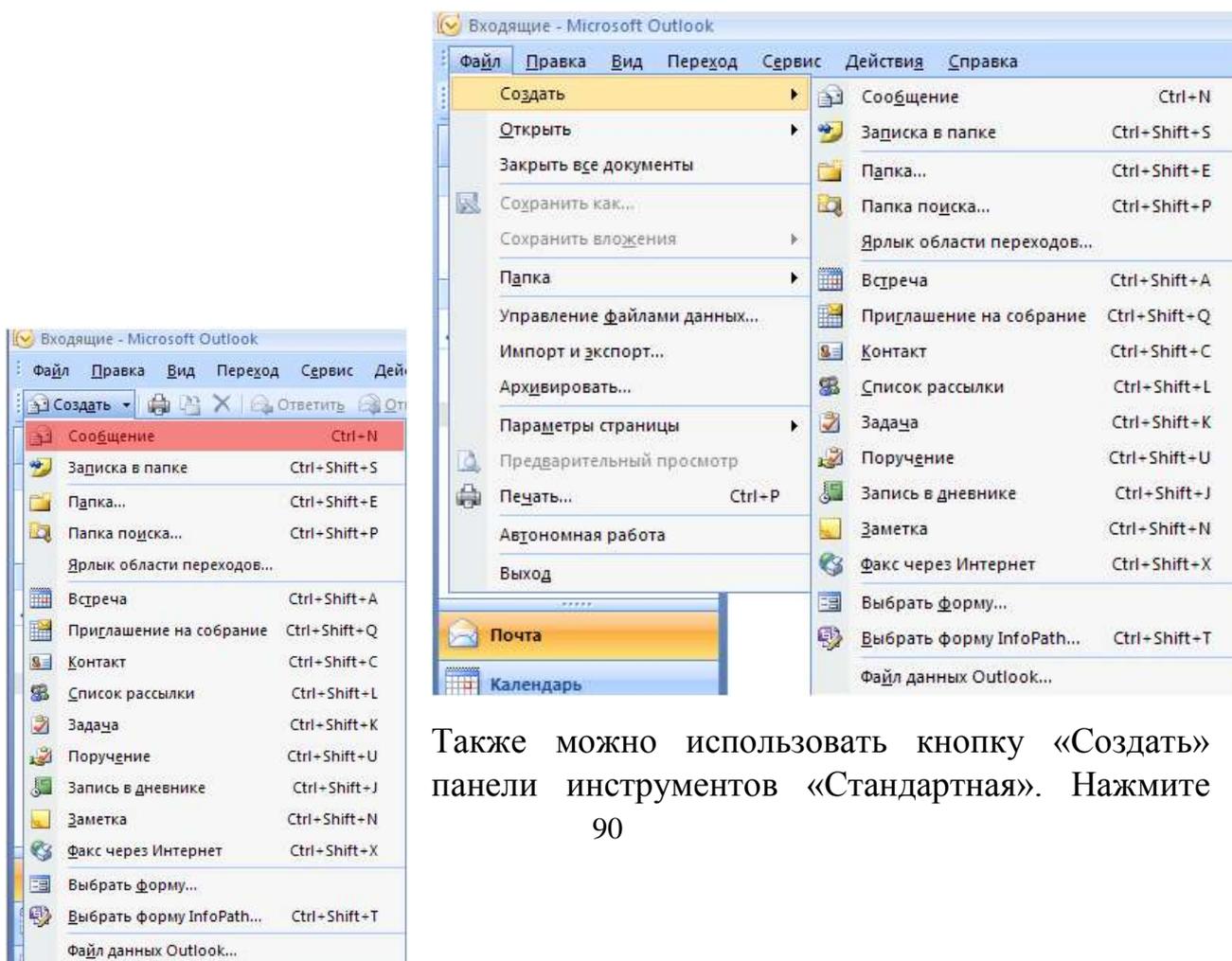


В правой части меню находится список команд, использующихся для быстрого создания элементов Outlook различного типа – письмо, заметка, контакт и так далее.

В правом нижнем углу расположена кнопка, открывающая диалог настройки редактора сообщений электронной почты.

Создание нового сообщения

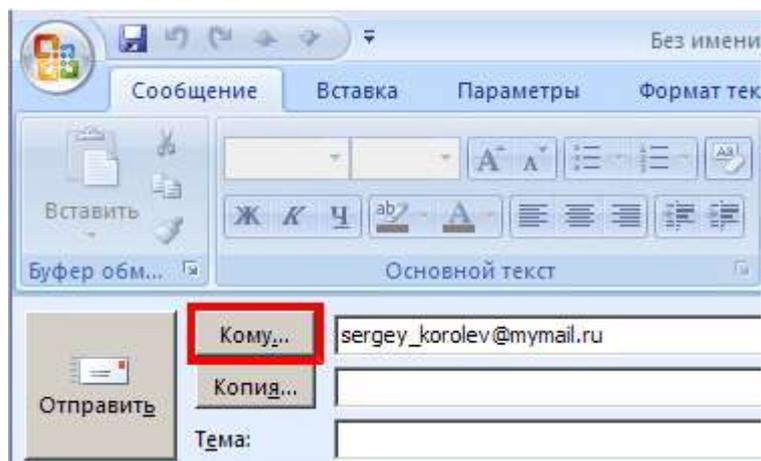
Для создания нового элемента Microsoft Office Outlook – почтового сообщения, задачи, контакта, заметки, - можно использовать команды меню «Файл», подменю «Создать».



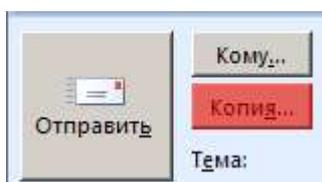
Также можно использовать кнопку «Создать» панели инструментов «Стандартная». Нажмите

стрелочку справа от кнопки «Создать» и выберите команду «Сообщение».

В поле «Кому» нужно ввести адрес электронной почты получателя.

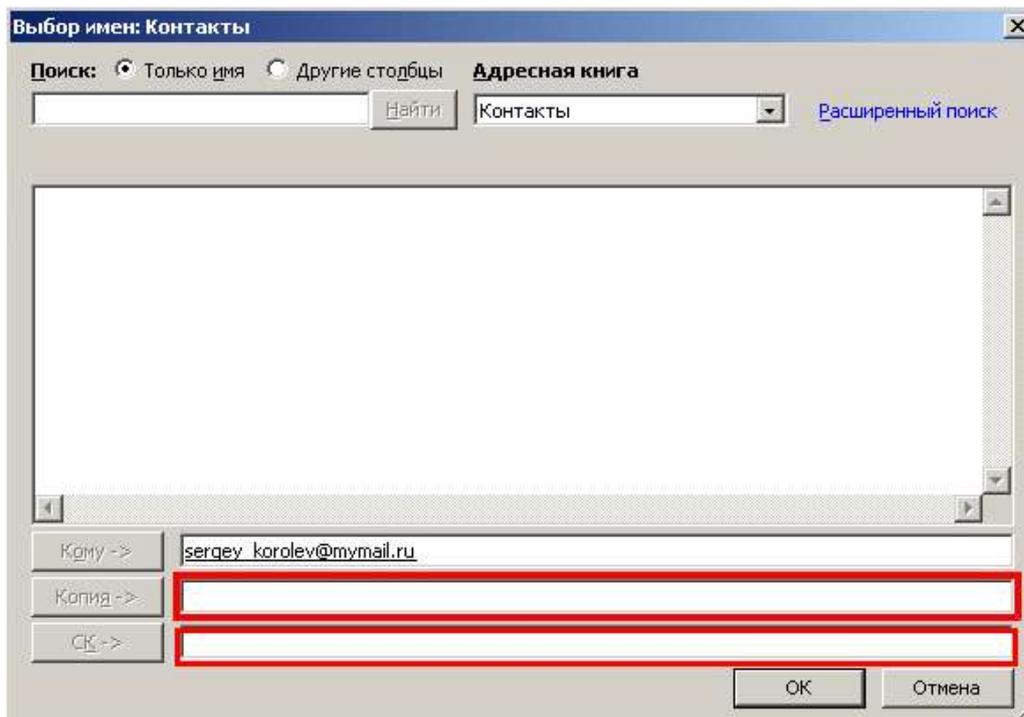


Если получателей сообщения несколько, их адреса нужно разделять точкой с запятой. Можно выбрать имена получателей из списка в адресной книге, воспользовавшись кнопкой «Кому...», расположенной слева от поля ввода.

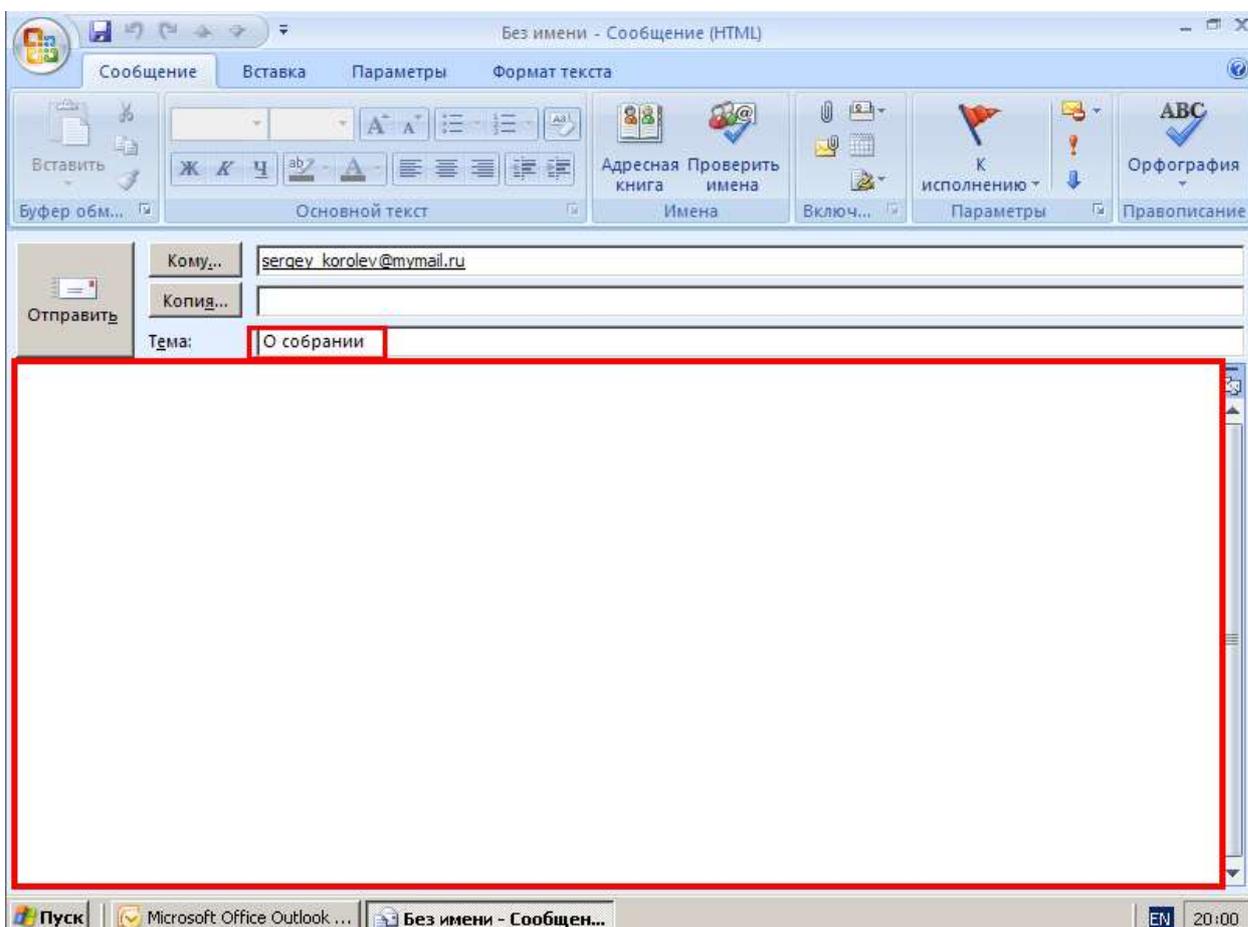


Если надо отправить копию сообщения, адреса получателей копии вводятся в поле «Копия».

Нажмите кнопку «Копия...» слева от поля ввода. В открывшемся окне диалога можно выбрать адреса получателей из адресной книги или папки «Контакты». Если в книге нужного адреса нет, можно ввести его вручную в поле «Копия» или «СК» - скрытая копия.

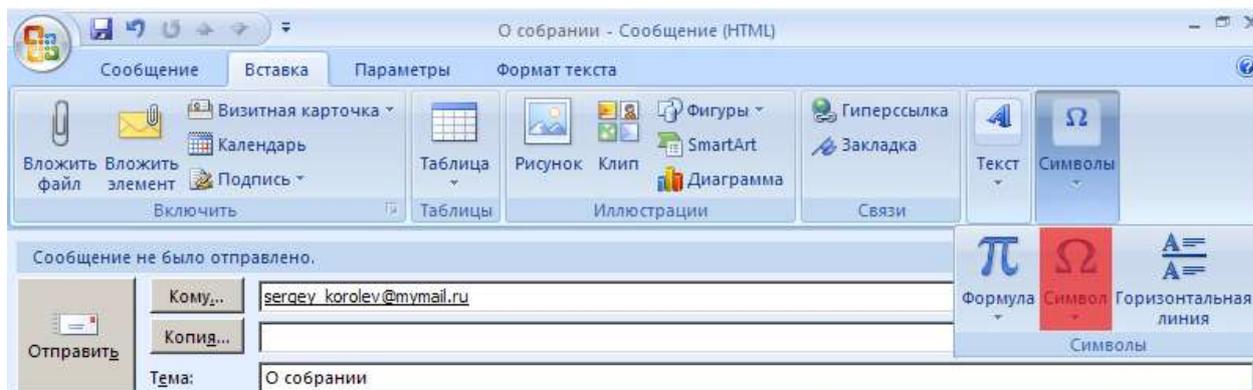


Если адрес введен в поле «СК», его не будут видеть другие адресаты. Если отправка копий сообщения не требуется, нажмите кнопку «Отмена». В поле «Тема» вводится тема сообщения, а в области просмотра и редактирования набирается его текст.

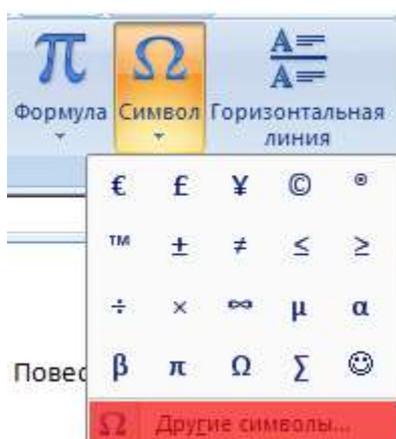


Для ввода отсутствующих на клавиатуре символов можно использовать команду «ленты» «Символы».

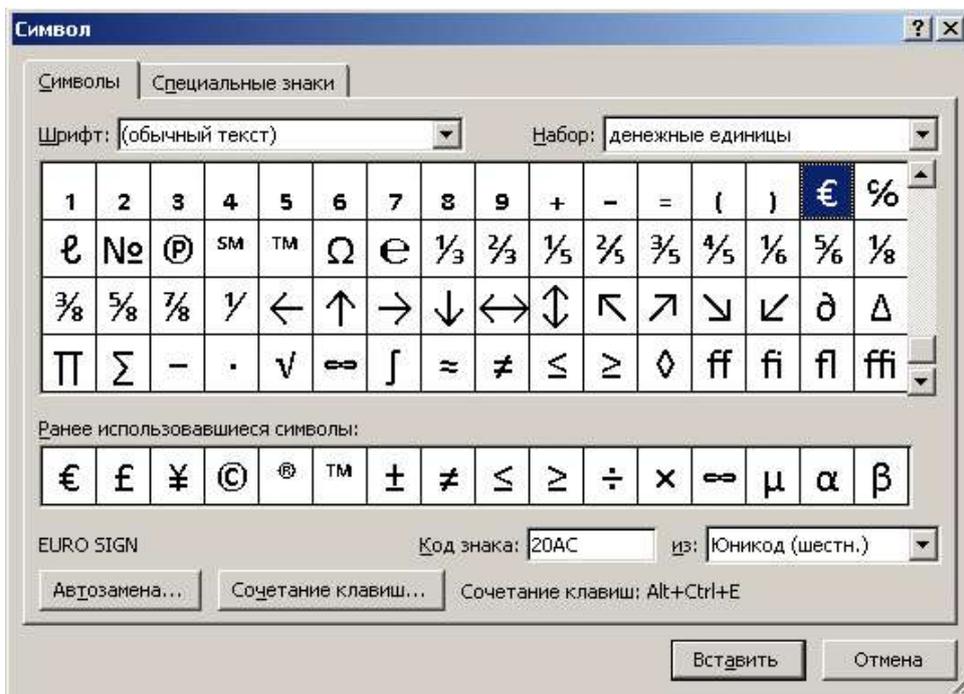
Перейдите на страницу «Вставка» и раскройте раздел «Символы». В открывшемся меню нажмите кнопку «Символ».



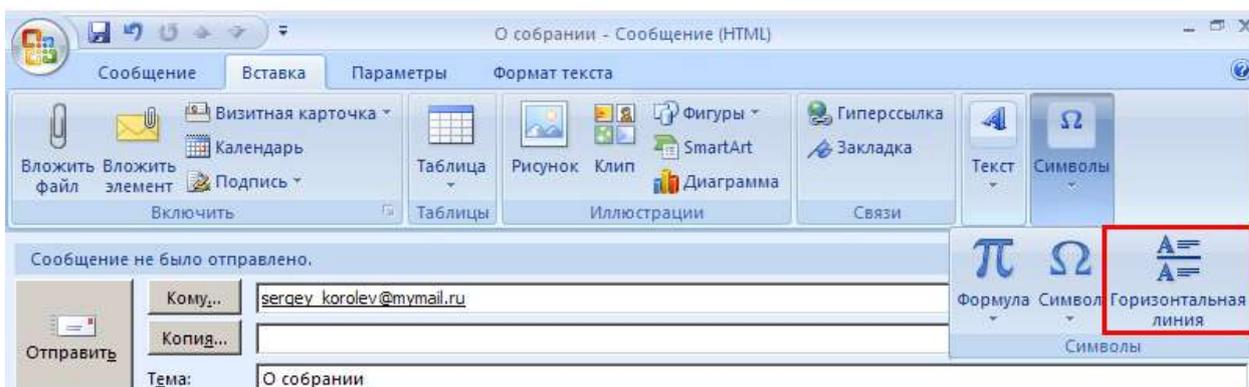
Если [нужного символа нет в галерее](#), щелкните на команде «Другие СИМВОЛЫ...».



Для вставки символа в текст сообщения достаточно выполнить на нем двойной щелчок левой кнопкой мыши.

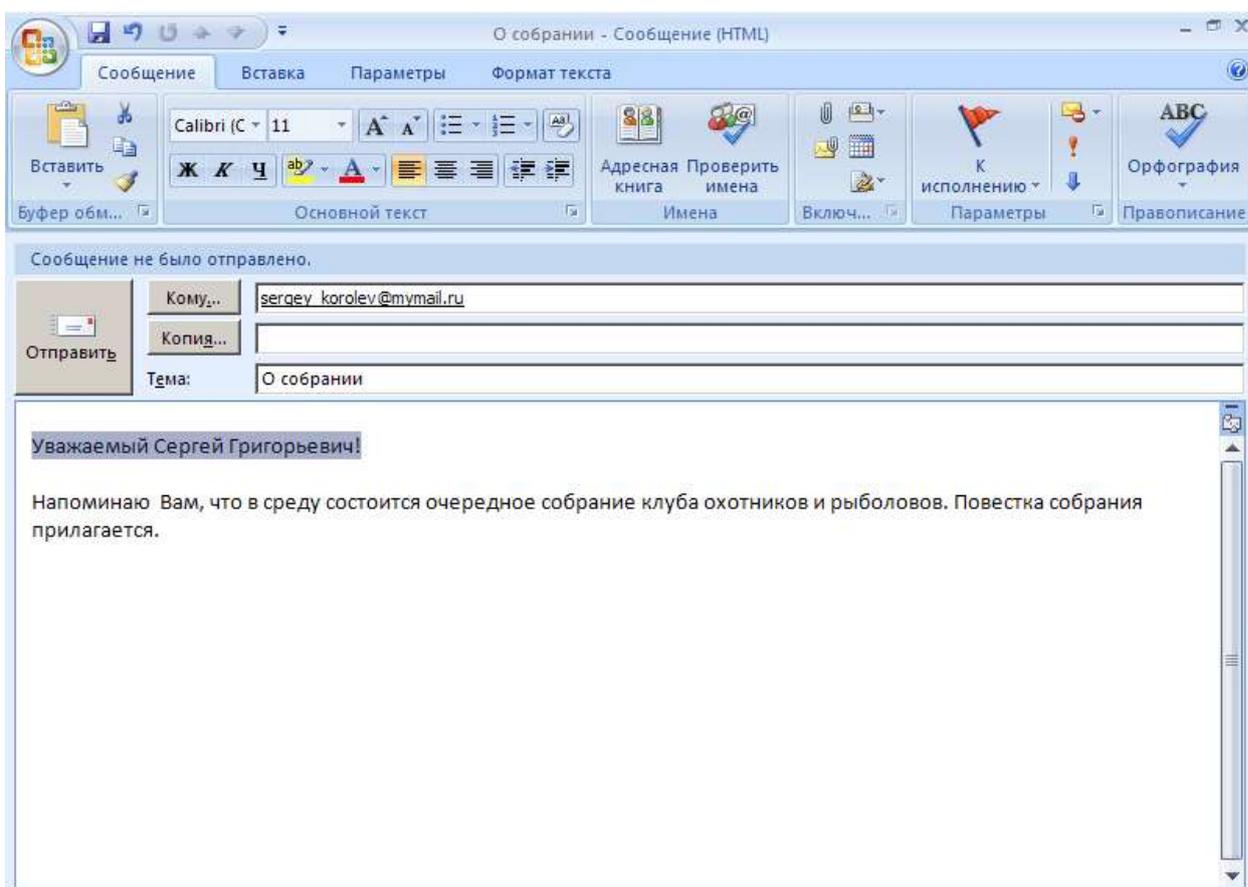


С помощью команды «Символы» можно также вставлять в сообщение разделительные линии.

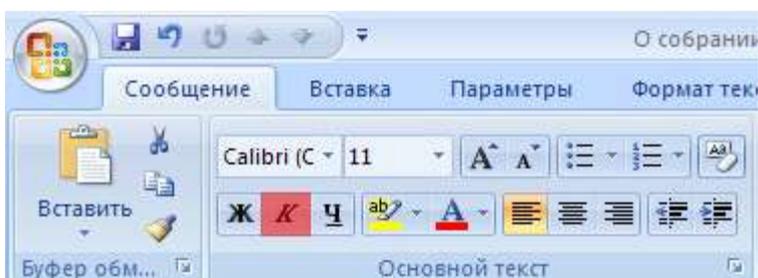


Оформление сообщения

Для оформления текста можно использовать инструменты на страницах «ленты» «Сообщение» или «Формат текста». Выделим часть текста.

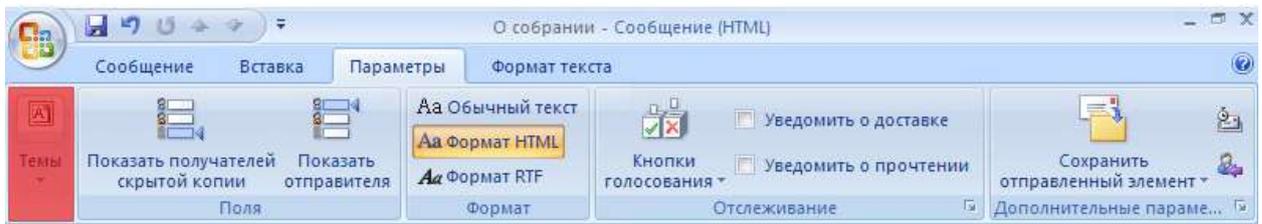


Например, чтобы выделить строку текста курсивом, нажмите кнопку «ленты» «Курсив» в разделе «Основной текст».

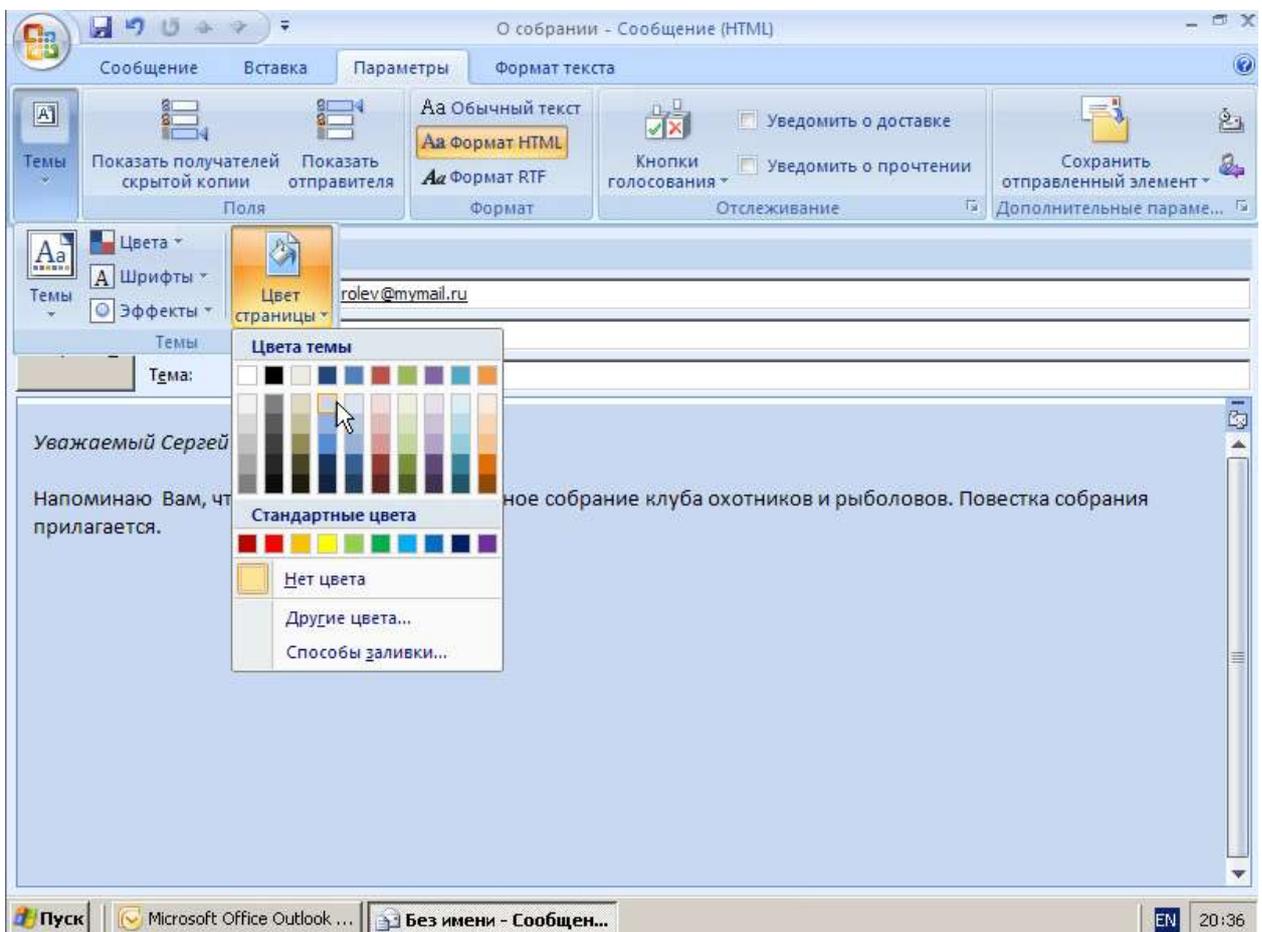


Для быстрого изменения фона сообщения, стилей основного текста и заголовка, вида разделяющих линий и прочих элементов можно выбрать одну из тем Outlook.

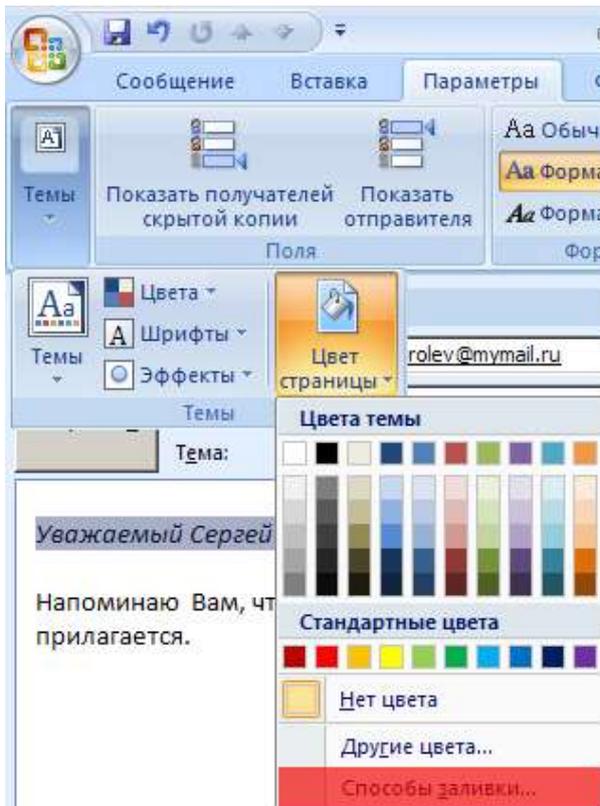
Тема представляет собой набор единообразных элементов оформления и цветовых схем и помогает создавать профессионально оформленные сообщения. Перейдите на страницу «ленты» «Параметры» и раскройте раздел «Темы».



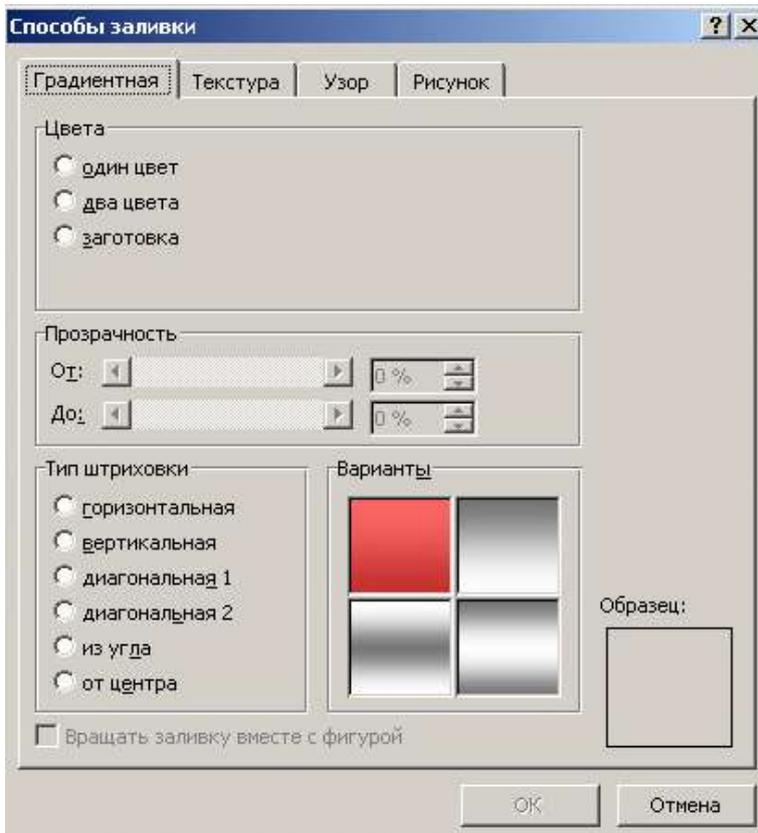
Для того чтобы применение готовых тем было доступно в Outlook, на компьютере должна быть установлена также программа Microsoft Office Word. Выбранный инструмент позволяет также настроить отдельные элементы оформления вручную. Нажмите кнопку «Цвет страницы». В открывшейся галерее можно выбрать цвет фона сообщения. При перемещении курсора на один из элементов галереи в окне просмотра сообщения можно видеть, как будет выглядеть сообщение после выбора данного элемента.



Можно применить при оформлении фона градиентную заливку, текстуру или рисунок. Для этого щелкните на ссылке «Способы заливки...»



в открывшемся окне диалога выберите нужный вариант.



Для подтверждений выбора нажмите кнопку «ОК».

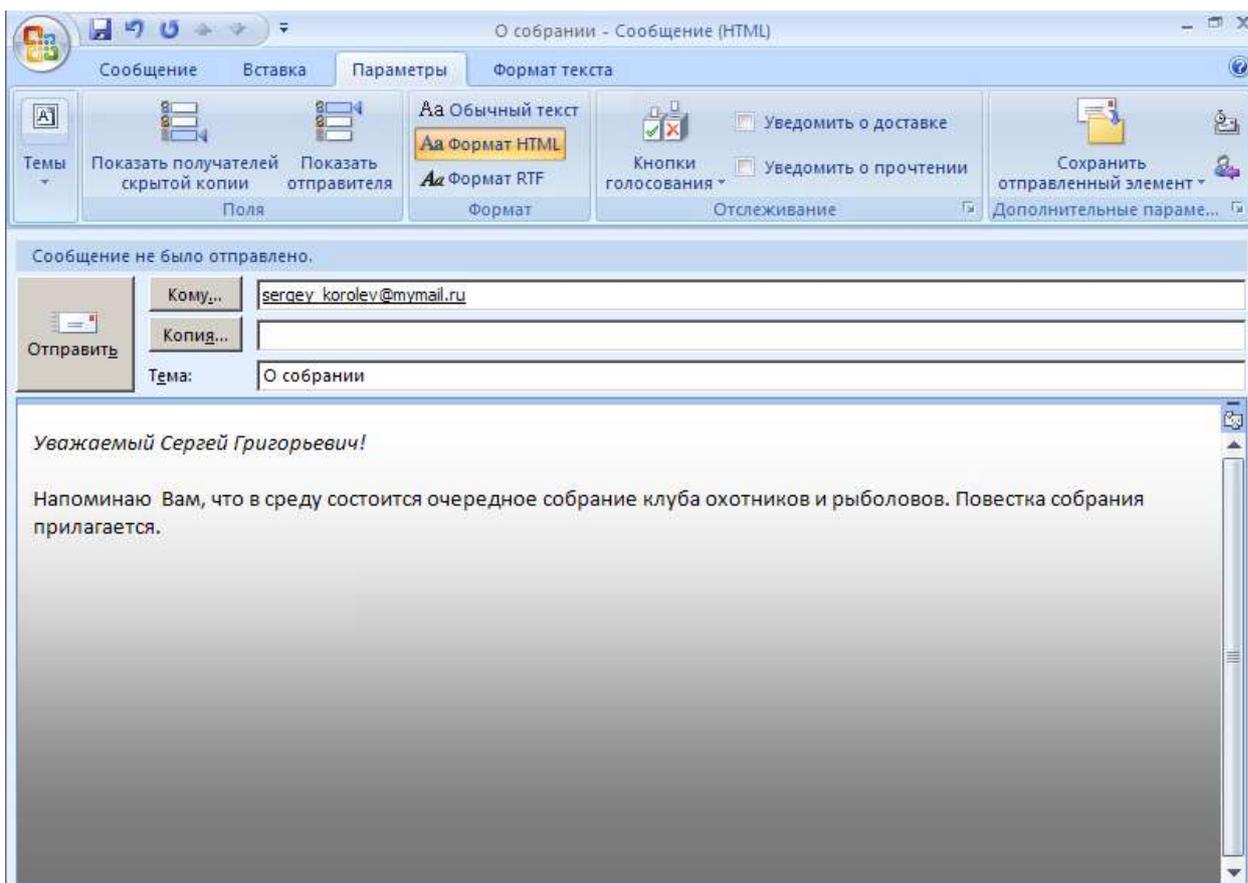
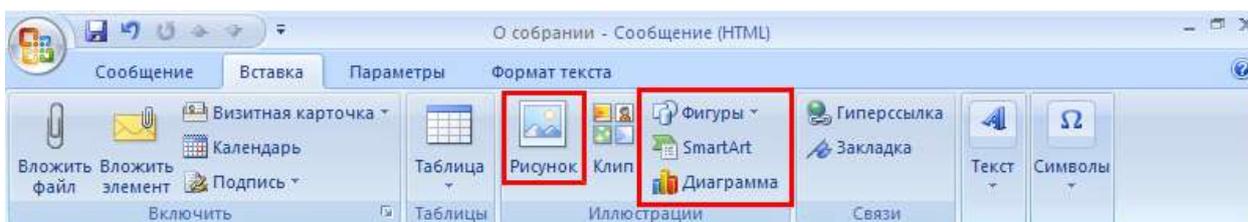
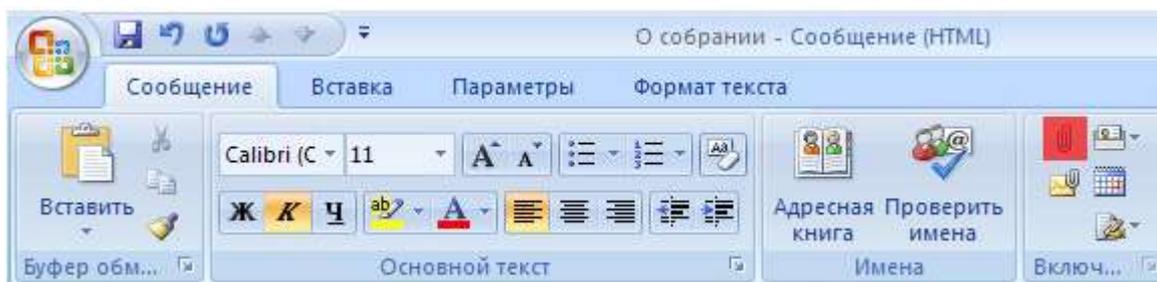


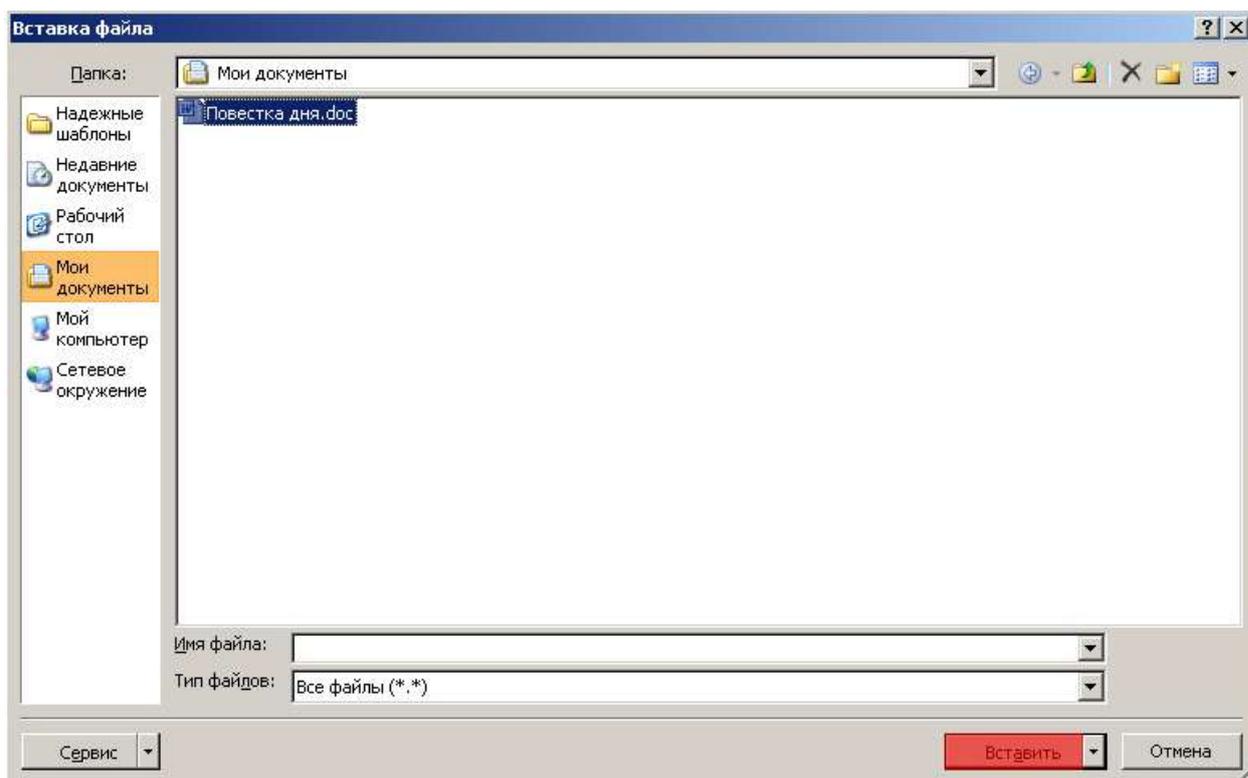
Рисунок можно добавить к письму либо как вложенный файл, либо как часть основного сообщения. На странице «ленты» «Вставка» в разделе «Иллюстрации» можно выбрать различные варианты добавления рисунков в тело письма – вставить из файла, использовать макет рисунка SmartArt, добавить фигуру нужной формы.



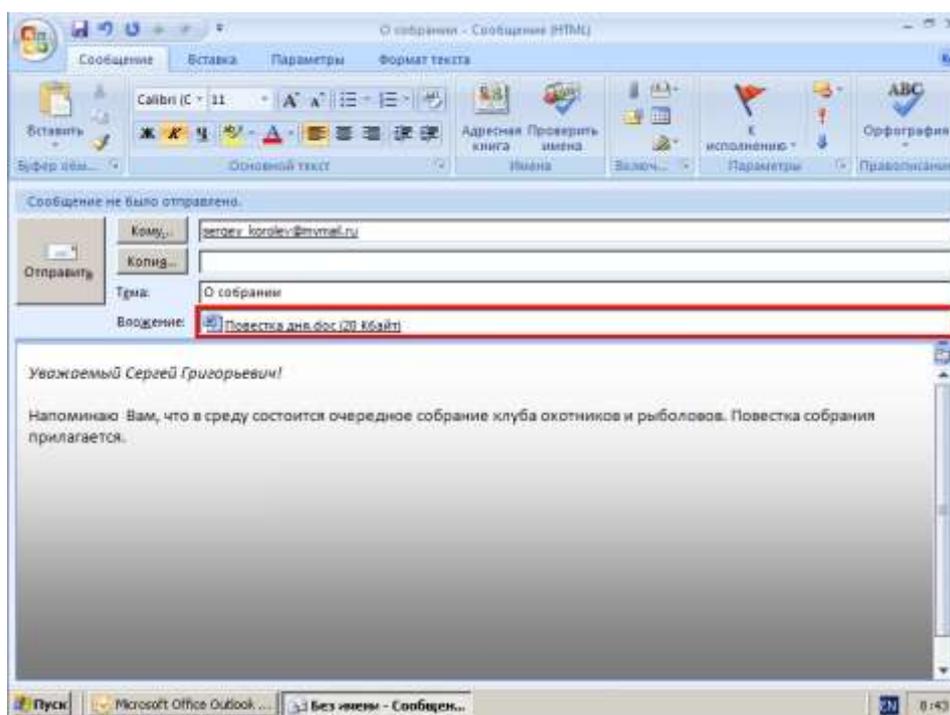
Для добавления к сообщению присоединенного файла перейдите на страницу «ленты» «Сообщение» и в разделе «Включить» нажмите кнопку «Вложить файл».



В открывшемся окне найдите и выделите нужный файл

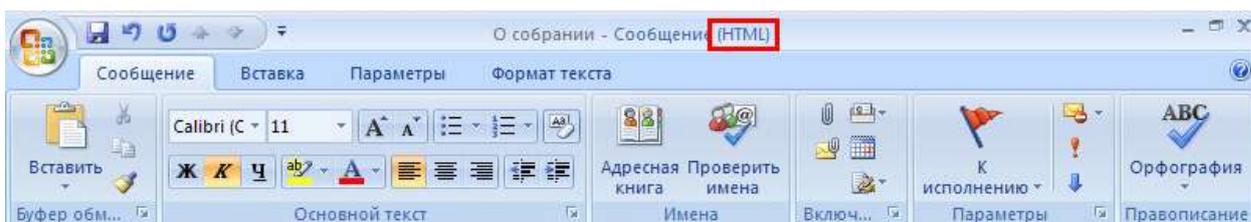


и нажмите кнопку «Вставить».



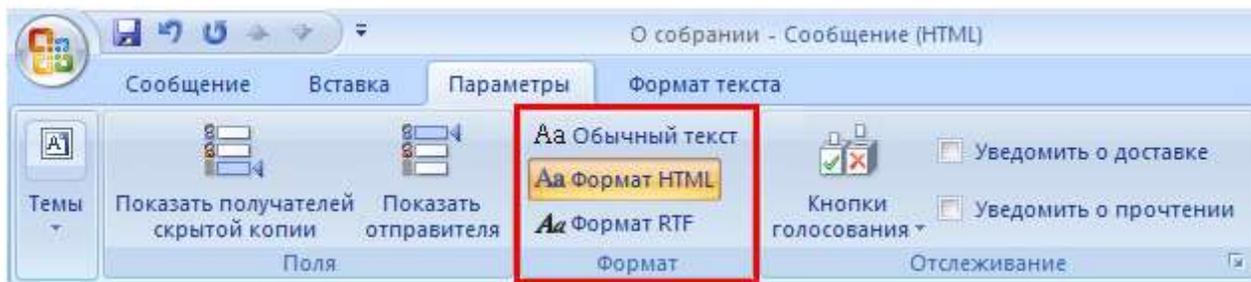
По умолчанию в Outlook блокируются потенциально опасные вложения - файлы с расширениями «BAT», «EXE», «VBS» и «JS», которые могут содержать вирусы. При вложении такого файла в сообщение

электронной почты Outlook предлагает подтвердить отправку. Можно вложить сразу несколько файлов, выбрав их и перетащив мышью на открытое сообщение Outlook. Как правило, имена вложенных файлов отображаются в поле «Вложение», здесь же можно видеть размер этих файлов. Формат сообщения отображается в строке заголовка сообщения.

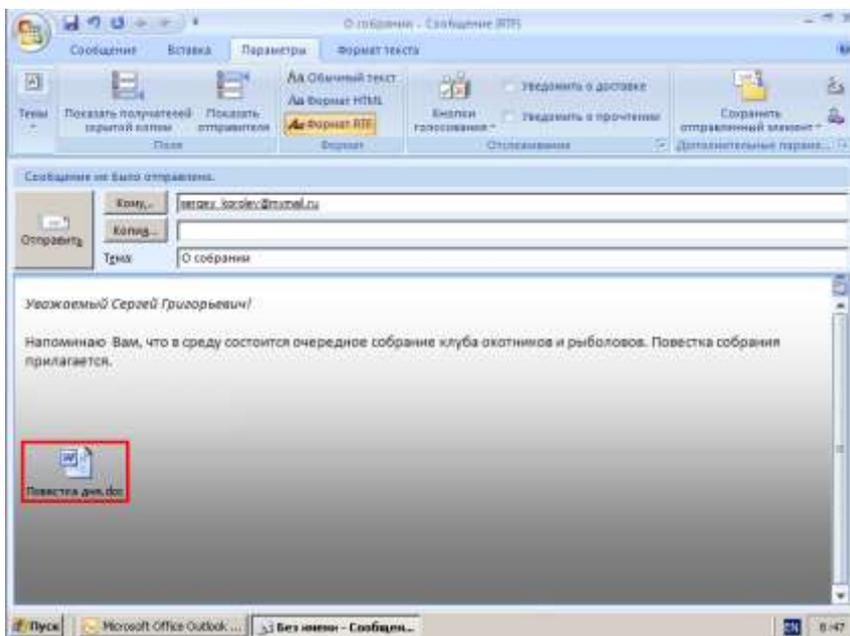
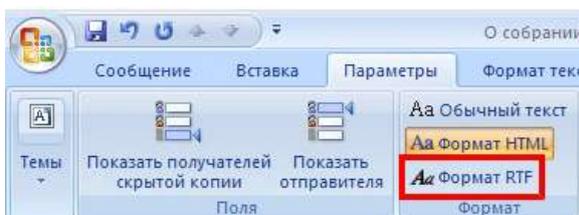


Изменить формат сообщения можно при помощи вкладки «ленты» «Параметры». Переключитесь на эту вкладку.

Формат сообщения изменяется в разделе «Формат».



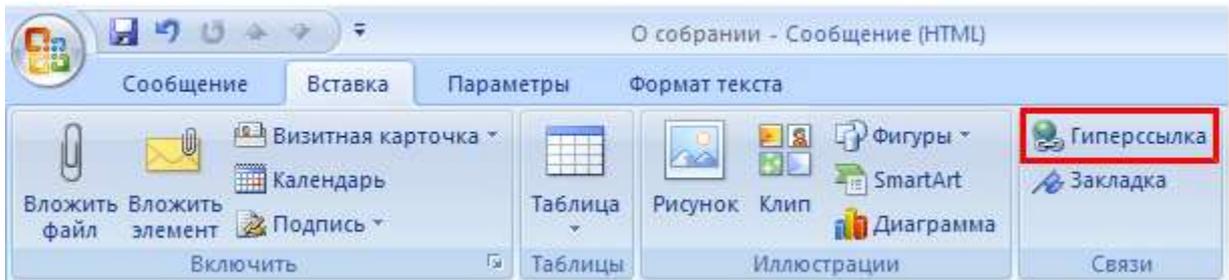
Если для сообщения выбран формат «RTF»,



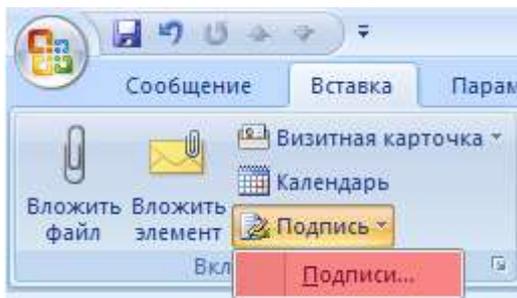
вложения отображаются в основном тексте сообщения.

В текст сообщения можно вставить гиперссылку на другие документы и веб-узлы. Щелкнув на такой ссылке, получатель письма сможет перейти к указанному файлу или веб-странице в сети Интернет. Перейдите на страницу «Вставка».

Добавить гиперссылку можно с помощью команды «Гиперссылка».



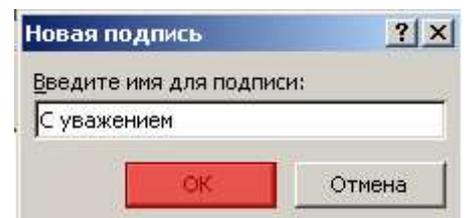
Подпись можно вставлять в каждое сообщение вручную либо создать подпись по умолчанию, которая будет добавляться во все отправляемые сообщения автоматически. Для задания подписи по умолчанию в разделе «Включить» нажмите кнопку «Подпись», затем выберите команду «Подписи...».



В левой части окна диалога «Подписи и бланки» можно добавлять в список и редактировать различные варианты подписи. Нажмите кнопку «Создать» для добавления новой подписи в список.

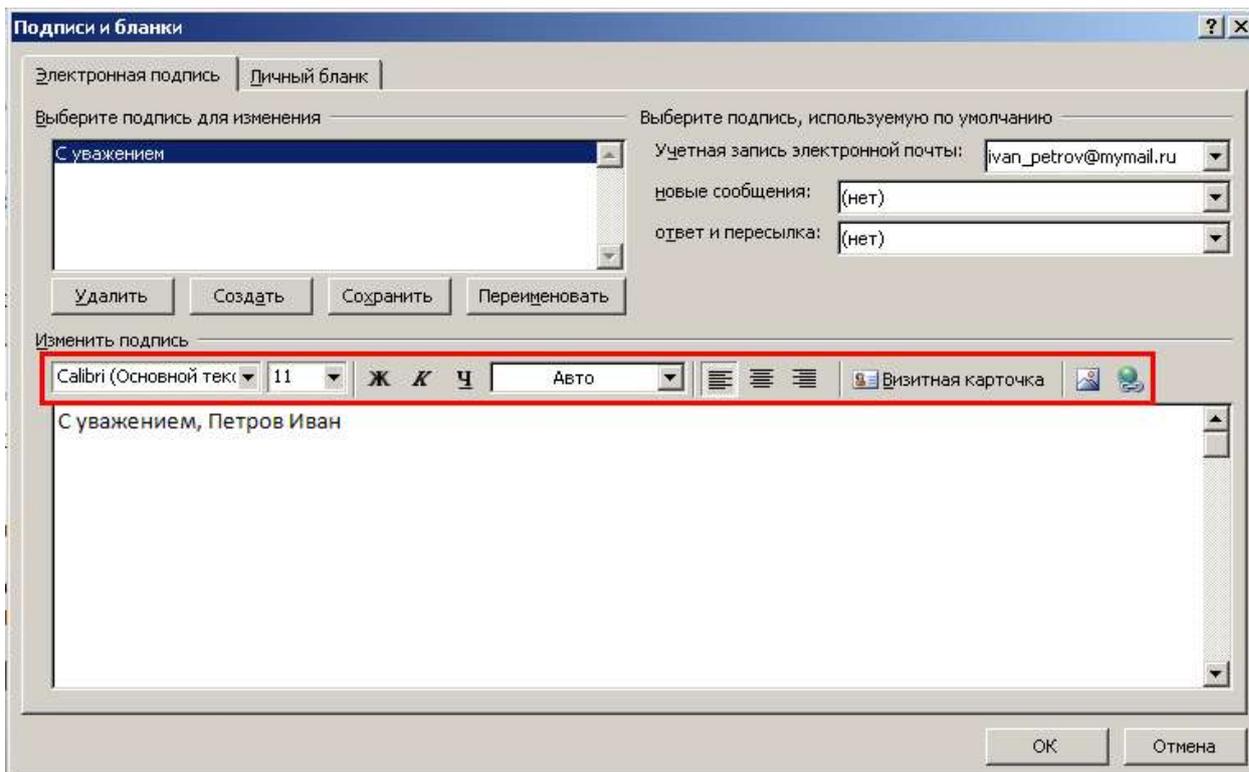


В поле нужно ввести имя, под которым она будет отображаться в списке.



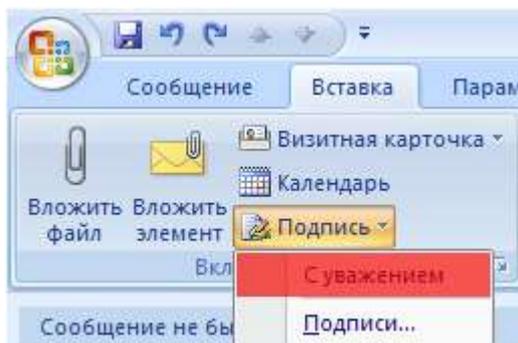
Нажмите кнопку «ОК».

В окне редактирования можно набрать текст новой подписи. При помощи инструментов форматирования можно изменить внешний вид текста подписи.

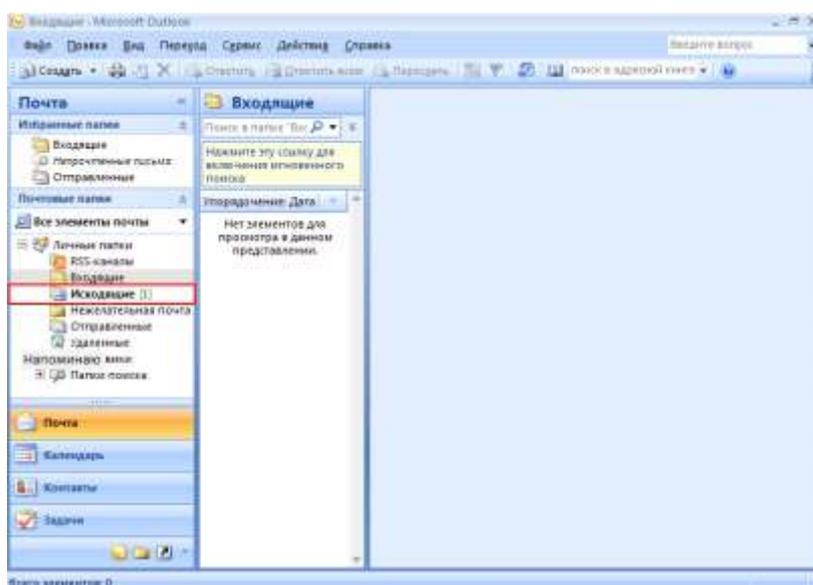
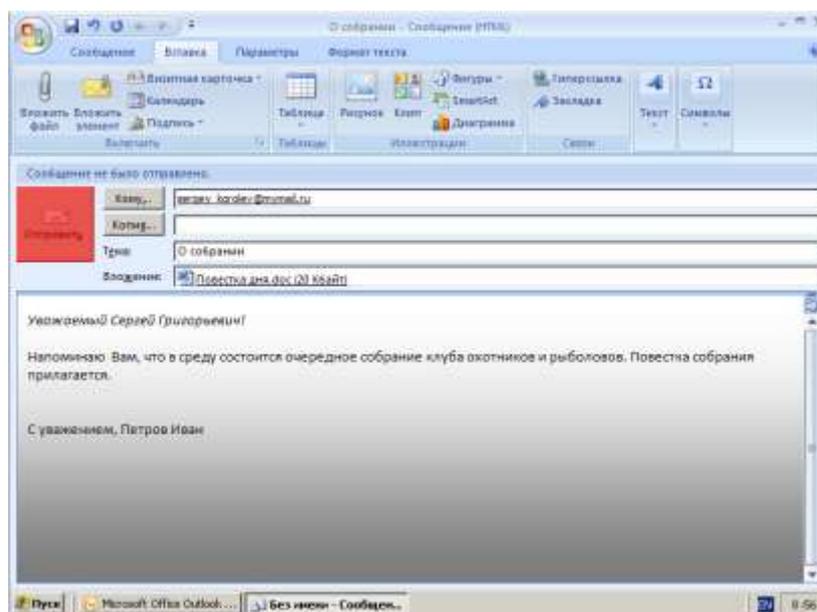


Для подтверждения сделанных изменений нажмите кнопку «ОК».

Теперь при создании нового сообщения подпись будет добавляться в текст письма автоматически. Чтобы вручную вставить подпись, нажмите кнопку «Подпись» и в списке выберите нужную подпись.



Для отправки готового сообщения нажмите кнопку «Отправить».



В зависимости от настроек Outlook сообщение может быть немедленно отправлено или помещено в папку «Исходящие».

Практическая работа №3. Pascal ABC.NET.

Цель: выработка навыков составления алгоритмов линейной и разветвляющейся структуры.

Методические указания

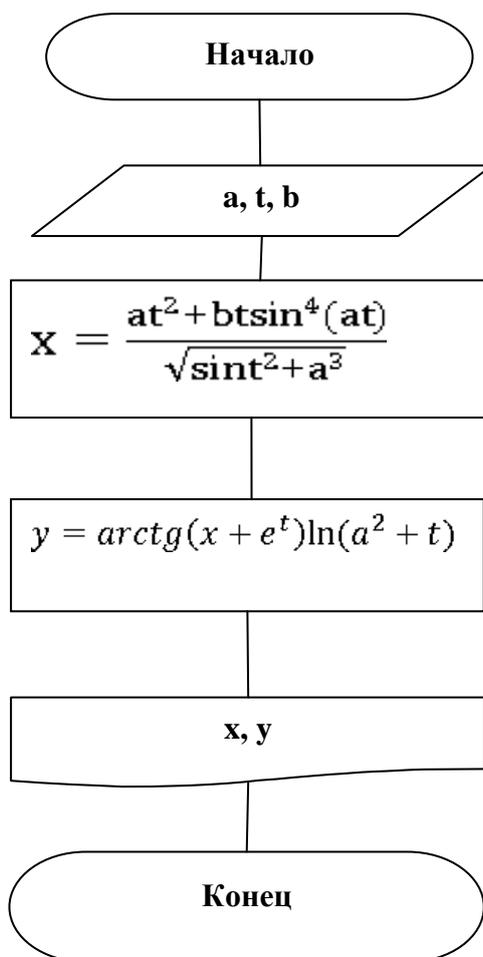
Алгоритм – точный порядок действий, определяющий процесс, ведущий от исходных данных к искомому результату и обладающий следующими свойствами: определённой, массовостью, результативностью, дискретностью, конечностью.

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

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.

Пример 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
```

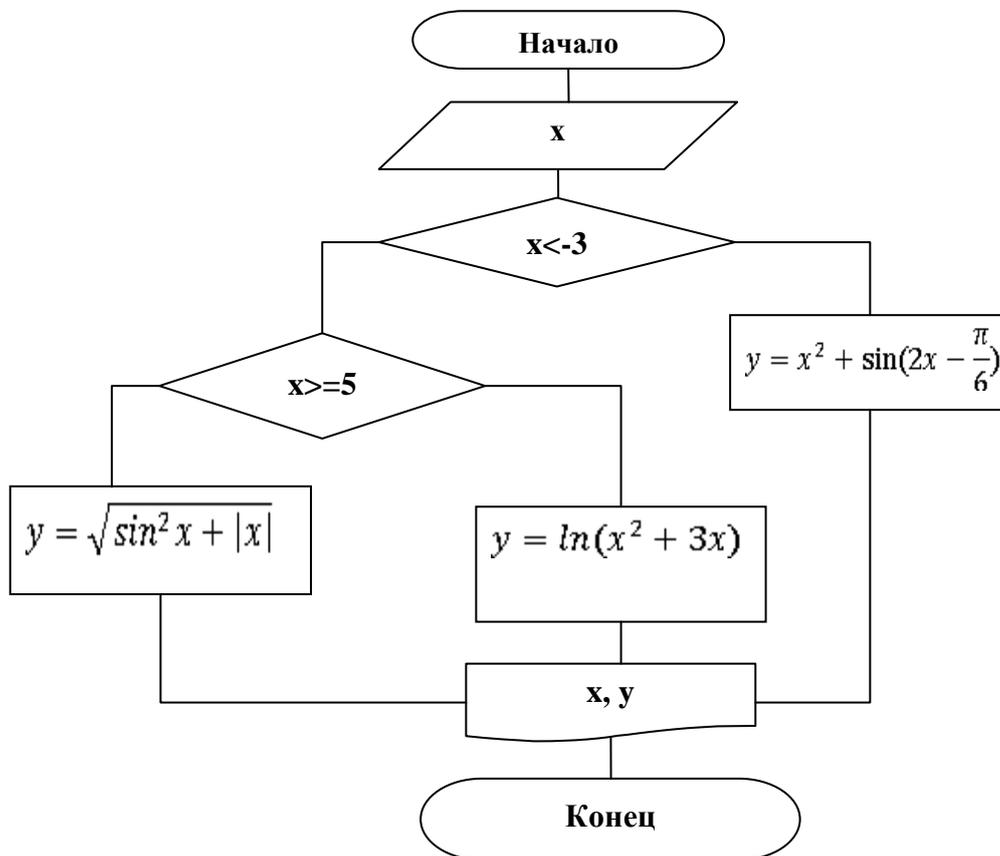
```

Writeln (' введите переменные a,t,b');
Readln (a, t, b);
X:=a*t*t+b*t*sqr(sqr(sin(a*t)))/sqr(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.

```

Пример 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);

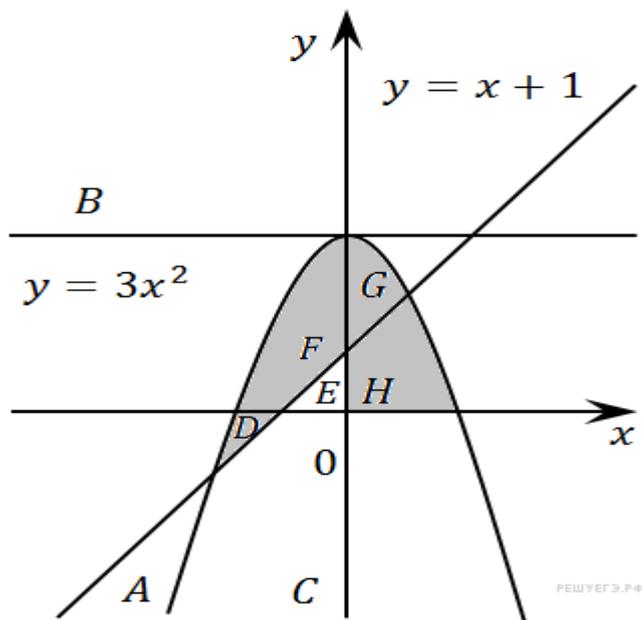
```

```

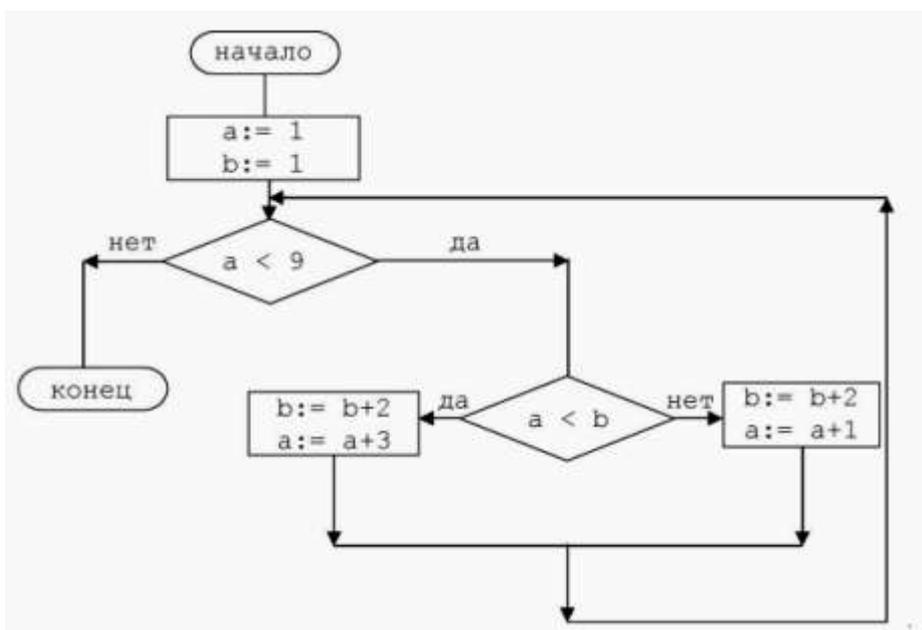
If x < -3 then y := x * x + sin(2 * x + pi / 6)
Else
If x >= 5 then Y := ln(x * x + 3 * x) else y := sqrt(sqrt(sin(x)) + abs(x));
Writeln('x=', x:7:3, 'y=', y:7:3);
End.

```

Пример 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).



Пример 4. Определите значение переменной b после выполнения фрагмента алгоритма:



Практическая работа №4. Циклические алгоритмы. Рекурсия.

Цель: научиться составлять алгоритмы 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 задачи¹⁴. 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	$i=N+h$

¹⁴ 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).

3-й шаг цикла и т.д.	$i=N+2h$
Последний шаг цикла	$i=K$

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

The screenshot shows the PascalABC.NET IDE with a program named 'Program5.pas'. The program calculates the sine and cosine of values from 0.0 to 0.9 in increments of 0.1. The output window displays a table with three columns: 'x', 'sin(x)', and 'cos(x)'. The compilation is successful.

```

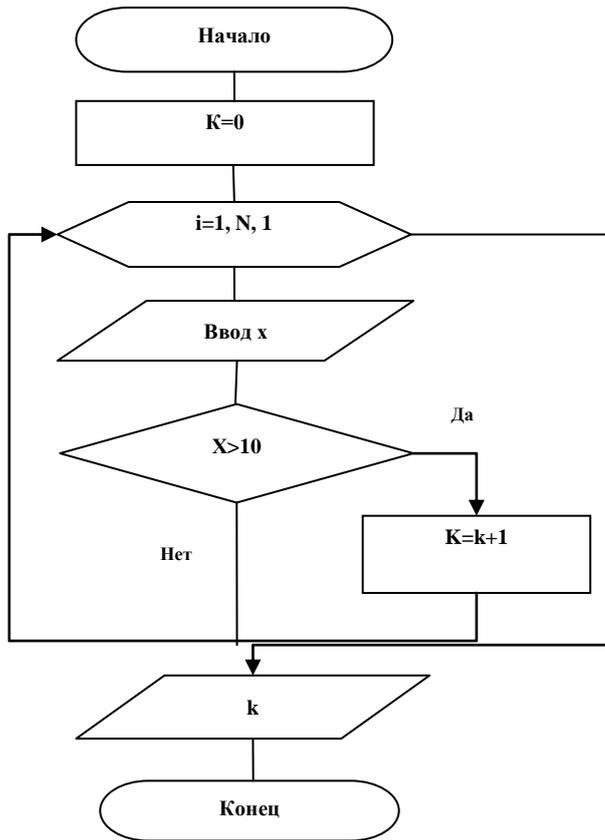
program primer_1;
var x:double;
k:integer; {целое, счётчик циклов}
begin
writeln('x', '      sin(x)', '      cos(x)');
x:=0;
for k:=1 to 10 do {тело цикла повторяется 10 раз}
begin {начало тела цикла}
writeln(x:4:1, sin(x):10:5, cos(x):10:5);
x:=x+0.1;
end;{конец тела цикла}
end.

```

x	sin(x)	cos(x)
0.0	0.00000	1.00000
0.1	0.09983	0.99500
0.2	0.19867	0.98007
0.3	0.29552	0.95534
0.4	0.38942	0.92106
0.5	0.47943	0.87758
0.6	0.56464	0.82534
0.7	0.64422	0.76484
0.8	0.71736	0.69671
0.9	0.78333	0.62161

Компиляция прошла успешно (12 строк) Строка 12 Столбец 5

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



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

Подсчёт суммы осуществляется следующим образом:

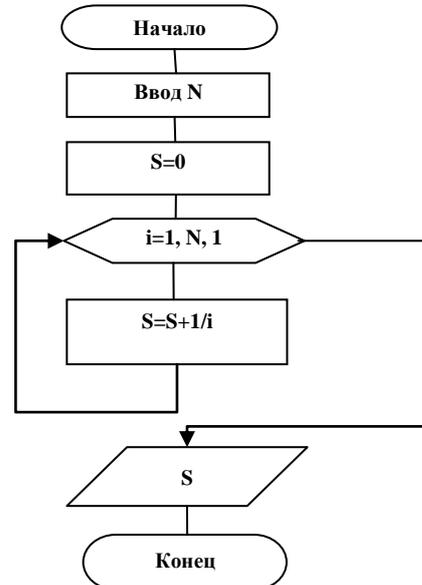
- 1) $S=1$.
- 2) $S=S+1/2$.
- 3) $S=S+1/3$.

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. Конеч.



The screenshot shows the PascalABC.NET IDE with a window titled "Program5.pas*". The code in the editor is as follows:

```
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.
```

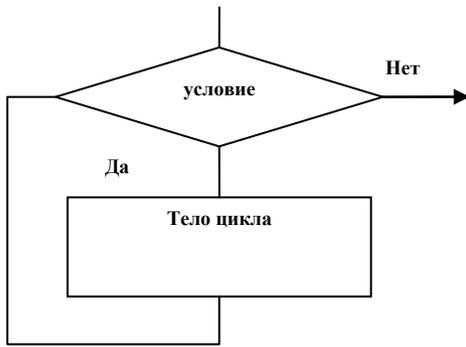
Below the code editor is an "Окно вывода" (Output window) showing the program's execution:

```
введите n
3
s=  1.833
```

At the bottom of the IDE, a status bar indicates "Компиляция прошла успешно (12 строк)" (Compilation successful (12 lines)) and "Строка 10 Столбец 5" (Line 10 Column 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.

Пример 2. 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

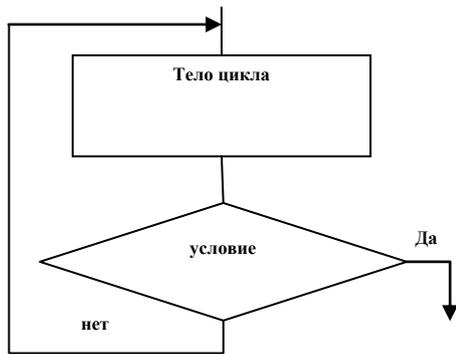
```

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.



Пример 3. 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;
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

Пример 4. 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 $x = 0,235$.

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).

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;

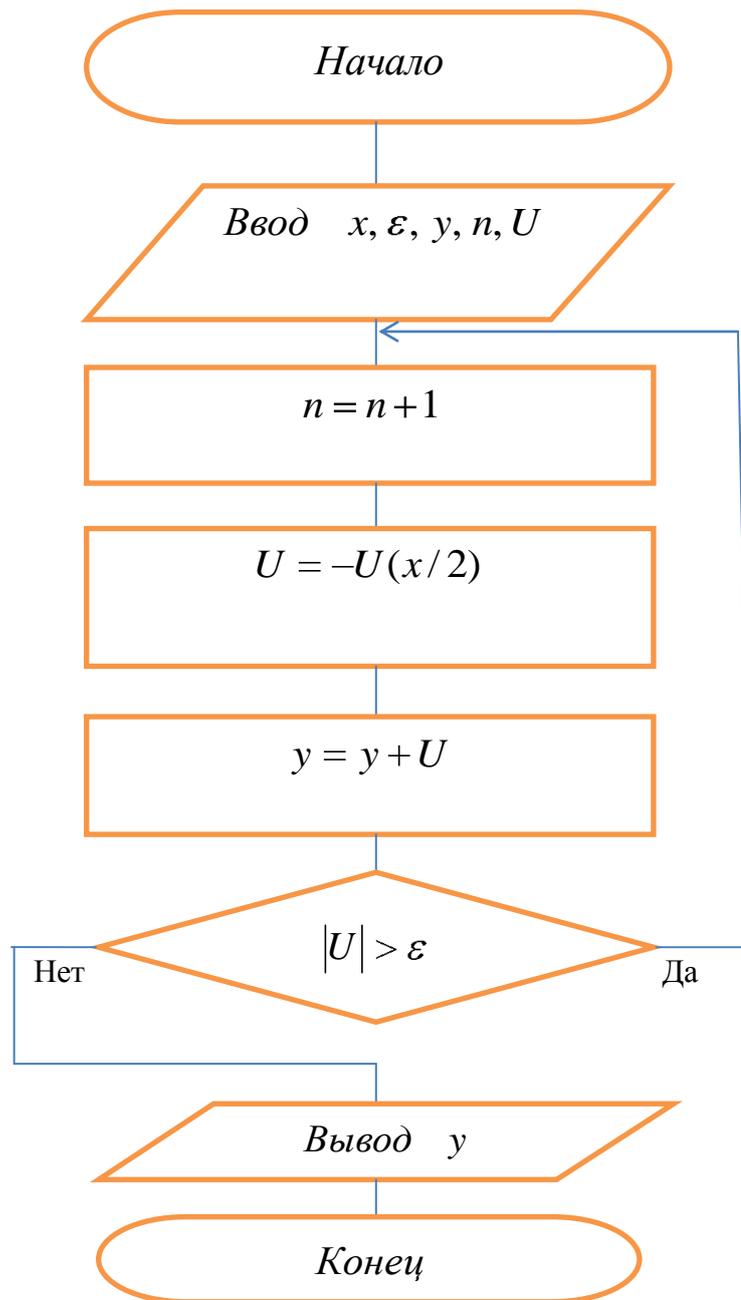
 u:= - u * x / 2;

 y:= y+u;

until abs (u)<= eps;

 writeln ('y=', y:9:4);

end.



Рекурсивный алгоритм

Рекурсивным называют алгоритм, организованный таким образом, что в процессе выполнения команд на каком-либо шаге он прямо или косвенно обращается сам к себе.

Типичная конструкция рекурсивной процедуры имеет вид¹⁶:

¹⁶ Окулов С.М. Основы программирования – 8-е изд., перераб.- М.: БИНОМ. Лаборатория знаний, 2015.- 336 с.: ил.- (Развитие интеллекта школьников).

```

Procedure Rec (t:integer);
Begin
<действия на входе в рекурсию>;
If <проверка условия> then Rec(t+1);
<действия на выходе из рекурсии>;
End;

```

Пример 1. Вычислить факториал.

```

Function factorial (n: integer): Longint;
Begin
    If n=1 then Factorial:=1
        Else Factorial:=n*Factorial(n-1);
End;

```

Вычисление рекуррентных последовательностей высоких порядков

Под рекуррентными последовательностями высоких порядков понимаются последовательности, для которых следующий элемент определён как функция от не менее чем двух предыдущих членов последовательности:

$a(n) = f(a(n-1), a(n-2), \dots, a(n-k))$, $n \geq k$, $a(0) = a_0$, $a(1) = a_1, \dots, a(k-1) = a_{k-1}$ – начальные условия (фиксированные числа). Здесь $f(x_1, \dots, x_k)$ – некоторая функция от k переменных. Классическим примером является последовательность Фибоначчи, в которой каждый последующий элемент равен сумме двух предыдущих:

$$f(0) = 0, f(1) = 1, f(n) = f(n-1) + f(n-2), n \geq 2.$$

Для вычисления членов последовательности Фибоначчи введём три целые переменные x , y , z и припишем им условное старшинство: $z \rightarrow f(k)$, $y \rightarrow f(k-1)$, $x \rightarrow f(k-2)$. Очевидно, $z = y + x$.

0 1 1, затем 1 1 2, затем 1 2 3 и т.д.

x y z x y z x y z

Видно, что значение переменной x теряется, так как заменяется значением переменной y , переменная y приобретает значение переменной z , а переменная z получает новое значение: $z=y+x$.

Пример 1. В общем виде n -е число Фибоначчи можно определить так:

$$f(n) = \begin{cases} 1, & \text{если } n = 1 \text{ или } n = 2; \\ f(n-1) + f(n-2), & \text{если } n > 2 \end{cases}$$

Function fib (n: integer): integer;

Begin

If $n \leq 2$ then fib:=1

Else fib:=fib(n-1)+fib(n-2);

End;

Пример 2. Вывести на печать все числа ряда Фибоначчи (1, 2, 3, 5, 8,...) до заданного натурального N .

Очередной член ряда F_i определяется как сумма двух предыдущих ($F_i = F_{i-1} + F_{i-2}$). Первые два члена ряда равны 1.

Program Fibonachi;

Var N, F, P, R : integer;

Begin

Write ('Введите натуральное число N');

Readln (N);

F:=1; P:=0;

While F<=N **do**

Begin write (F, ' '); {вывод очередного члена ряда}

R := F; {сохранение значения очередного члена ряда}

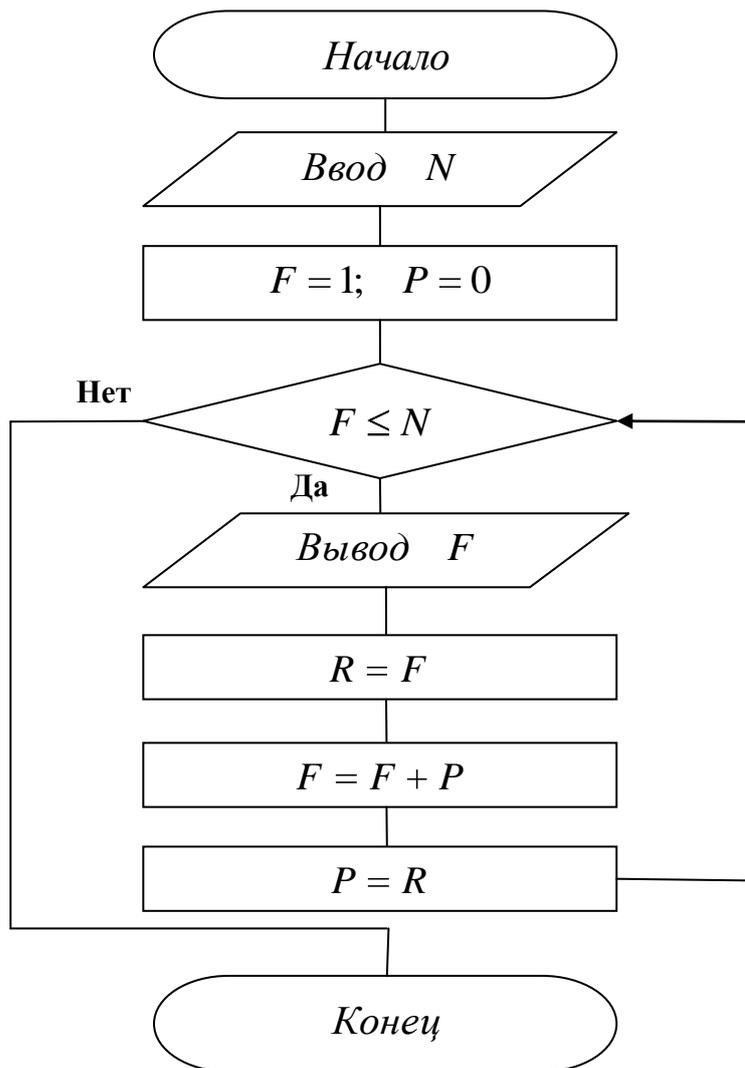
F := F+P; {вычисление следующего значения}

P := R; {восстановление предыдущего значения}

End;

Writeln;

End.



Практическая работа №5. 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:

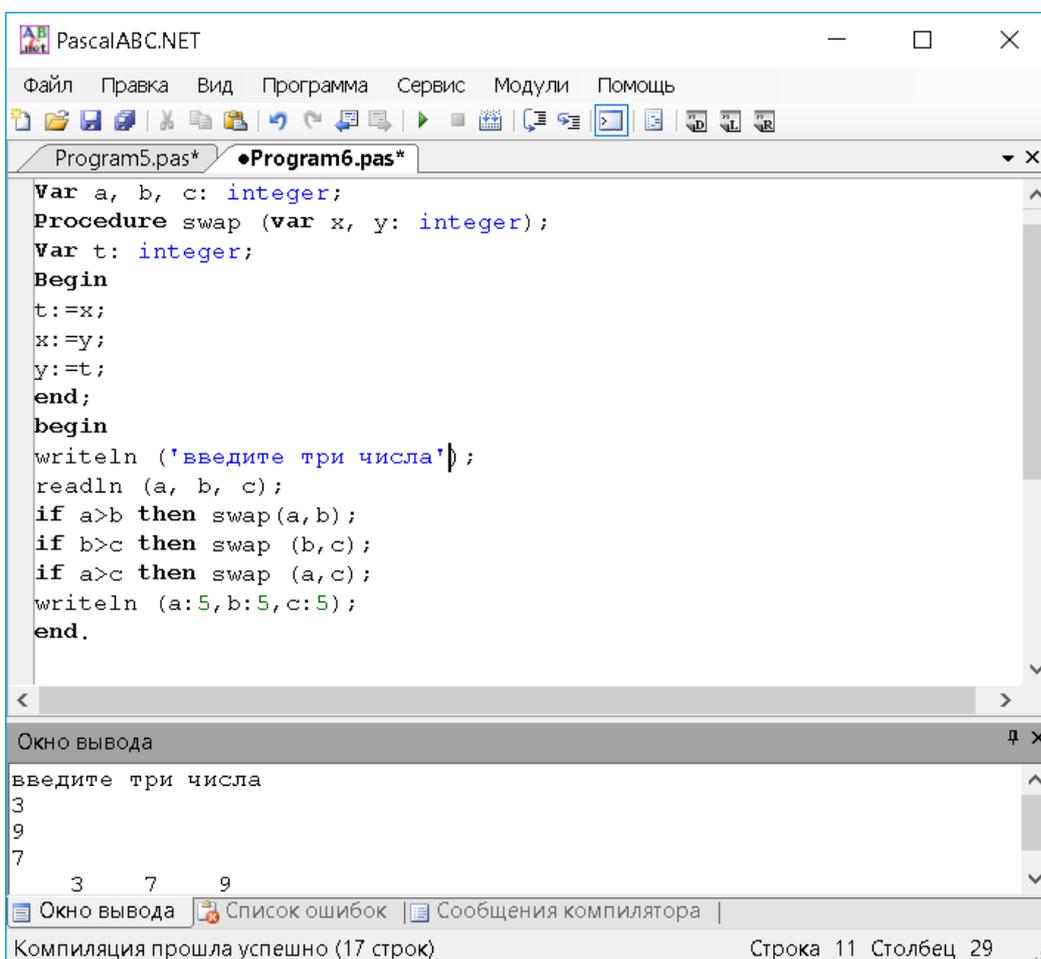
Procedure – служебное слово;

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 $a < b < c$.



```
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.
```

Окно вывода

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

Компиляция прошла успешно (17 строк) Строка 11 Столбец 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;
    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:

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

```
Program primer_3;
Var n,m:integer;
Function S(n,m:integer):longint;
Var i:integer;
Rez,cht:longint;
Begin
Rez:=1;
Cht:=1;
For i:=1 to m do begin
Rez:=rez*i;
Cht:=cht*(n-i+1);
End;
S:=cht div rez;
End;
Begin
Writeln('введите два числа');
Readln(n,m);
Writeln(S(n,m));
End.
```

Окно вывода

```
введите два числа
13
5
1287
```

Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (19 строк) Строка 16 Столбец 28

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 №6. Одномерные массивы. Двумерные массивы.

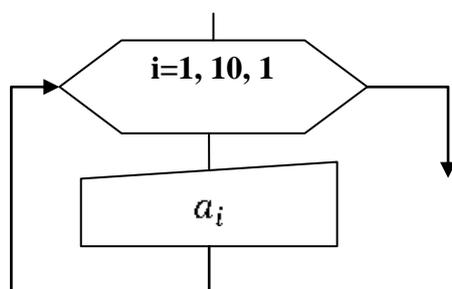
Цель: научиться обрабатывать элементы одномерных и двумерных массивов.

Тип данных, позволяющий хранить вместе под одним именем несколько переменных, называется структурированным. Каждый язык программирования имеет свои структурированные типы¹⁷.

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

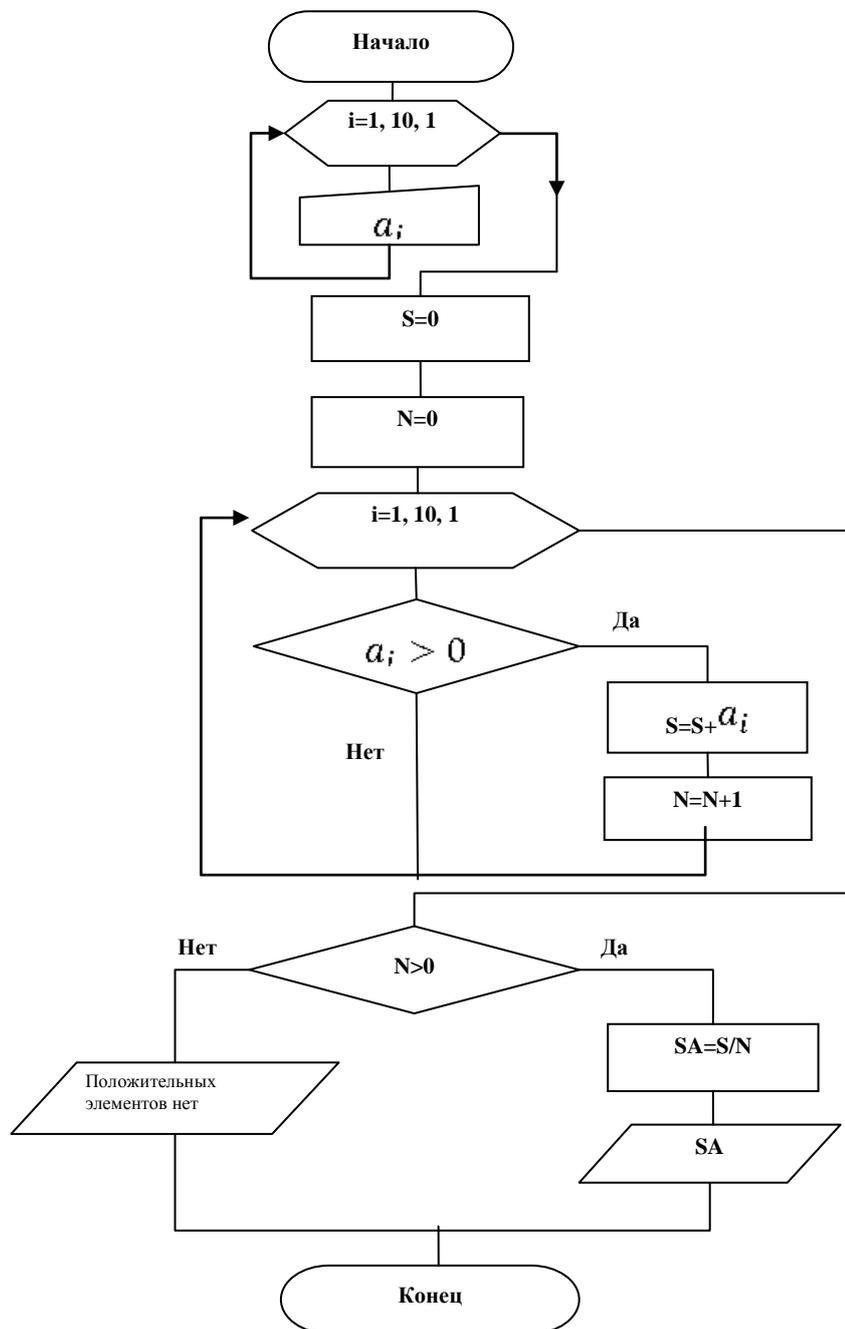
Одномерный массив предполагает наличие у каждого элемента только одного индекса. Примерами одномерных массивов служат арифметическая a_i и геометрическая b_i последовательности, определяющие конечные ряды чисел. Количество элементов массива называют размерностью.

Ввод элементов одномерного массива:



¹⁷ 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).

Пример 1. Вычислить среднее арифметическое положительных элементов массива.

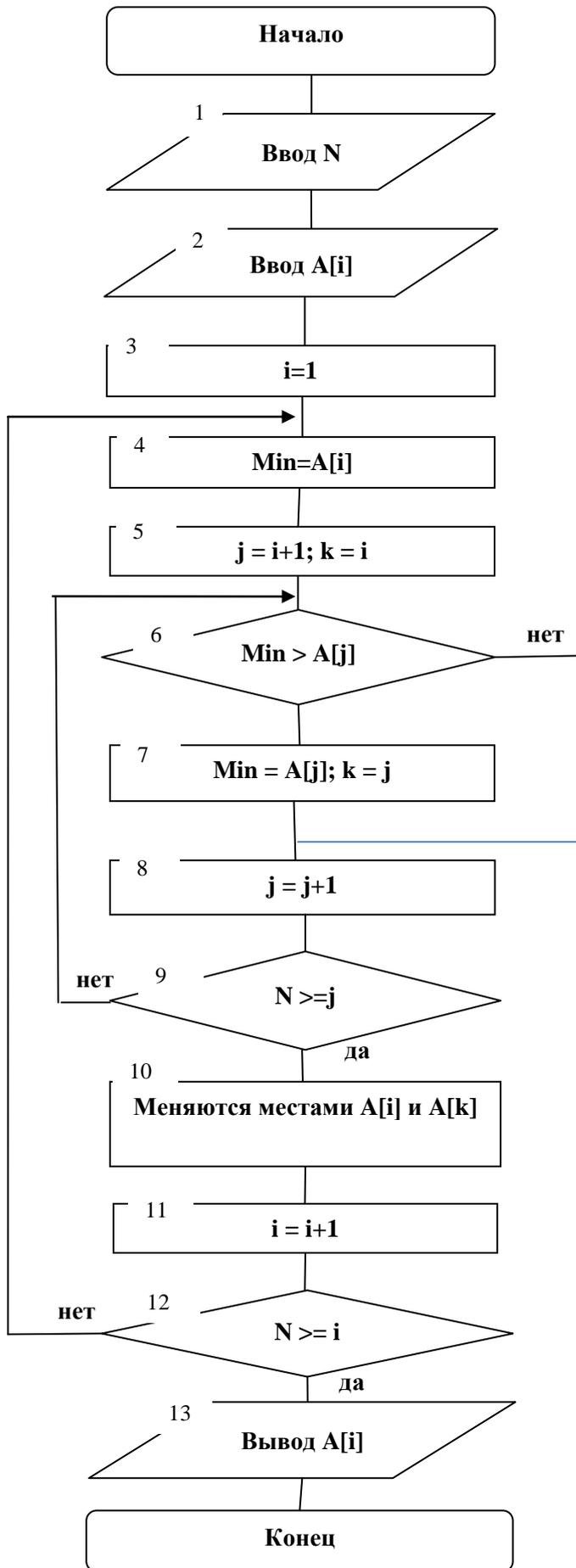


Пример 2. Выполнить сортировку по возрастанию N целых чисел.

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 border="1"> <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 border="1"> <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 border="1"> <tr><td>1</td><td>2</td><td>3</td><td>11</td><td>4</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	1	2	3	11	4	1	2	3	4	5	3-й проход	$i=3$ 3-й элемент не меняет места
1	2	3	11	4								
1	2	3	4	5								
<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>11</td><td>4</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	1	2	3	11	4	1	2	3	4	5	4-й проход	$i=4$ 4-й элемент меняется с 5-м
1	2	3	11	4								
1	2	3	4	5								
<table border="1"> <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*' open. The code implements a selection sort algorithm. The output window shows the user inputting 5 elements (2, 4, 6, 8, 9) and the program outputting the sorted array (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.

```

Окно вывода

```

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

```

Окно вывода | Список ошибок | Сообщения компилятора

Компиляция прошла успешно (22 строк) Строка 6 Столбец 24

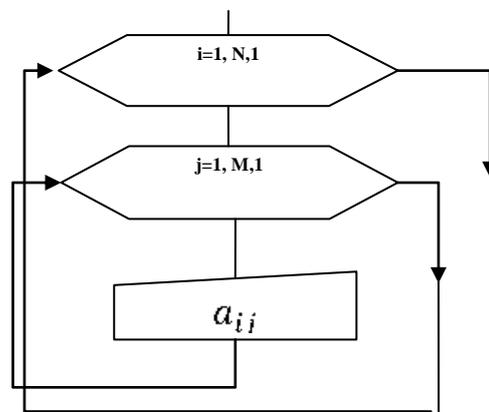
Двумерные массивы

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

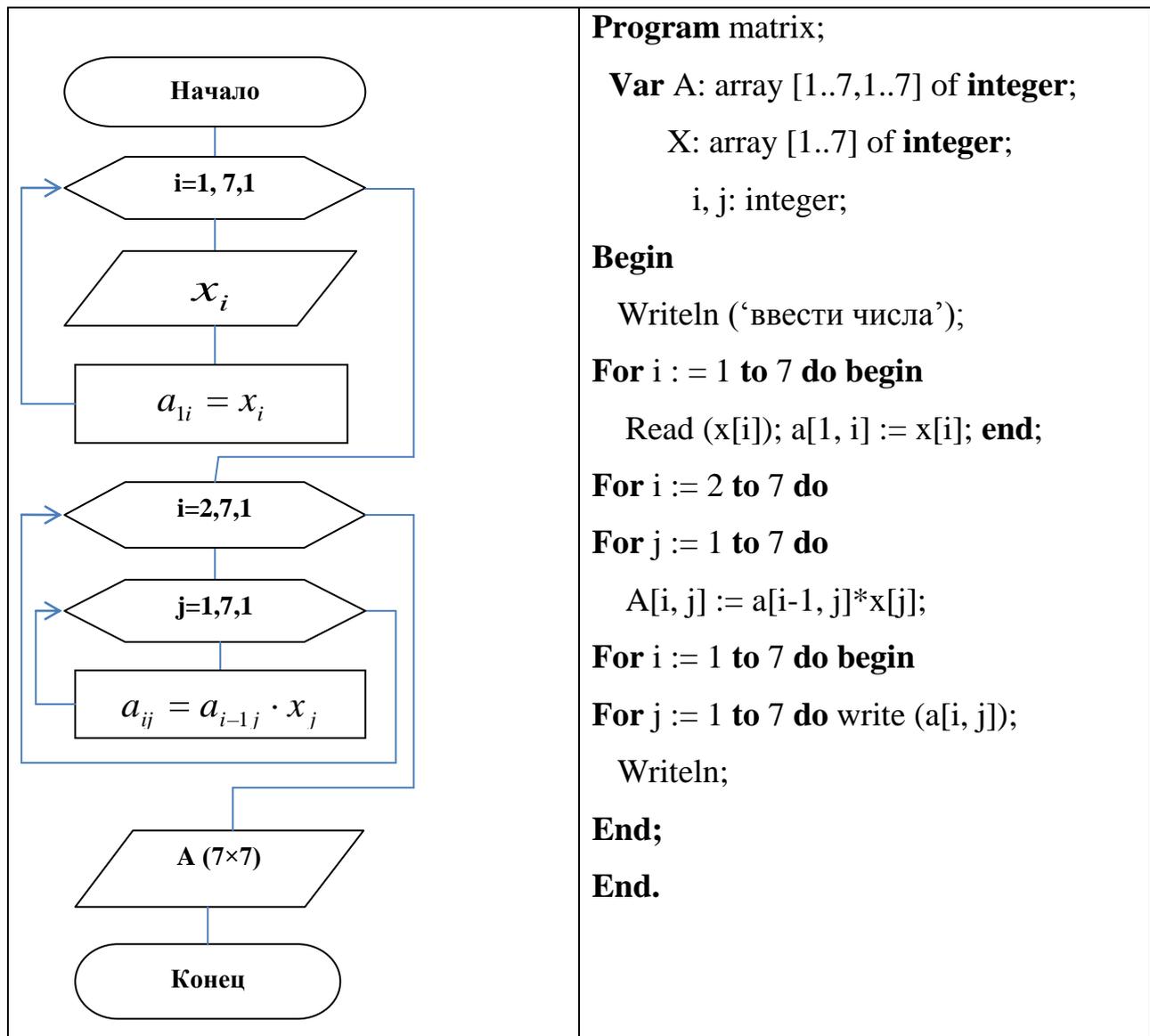
Ввод элементов двумерного массива осуществляется построчно, в свою очередь, ввод каждой строки производится поэлементно, тем самым определяется циклическая конструкция, реализующая вложение циклов. Внешний цикл определяет номер вводимой строки i , внутренний – номер элемента по столбцу j .



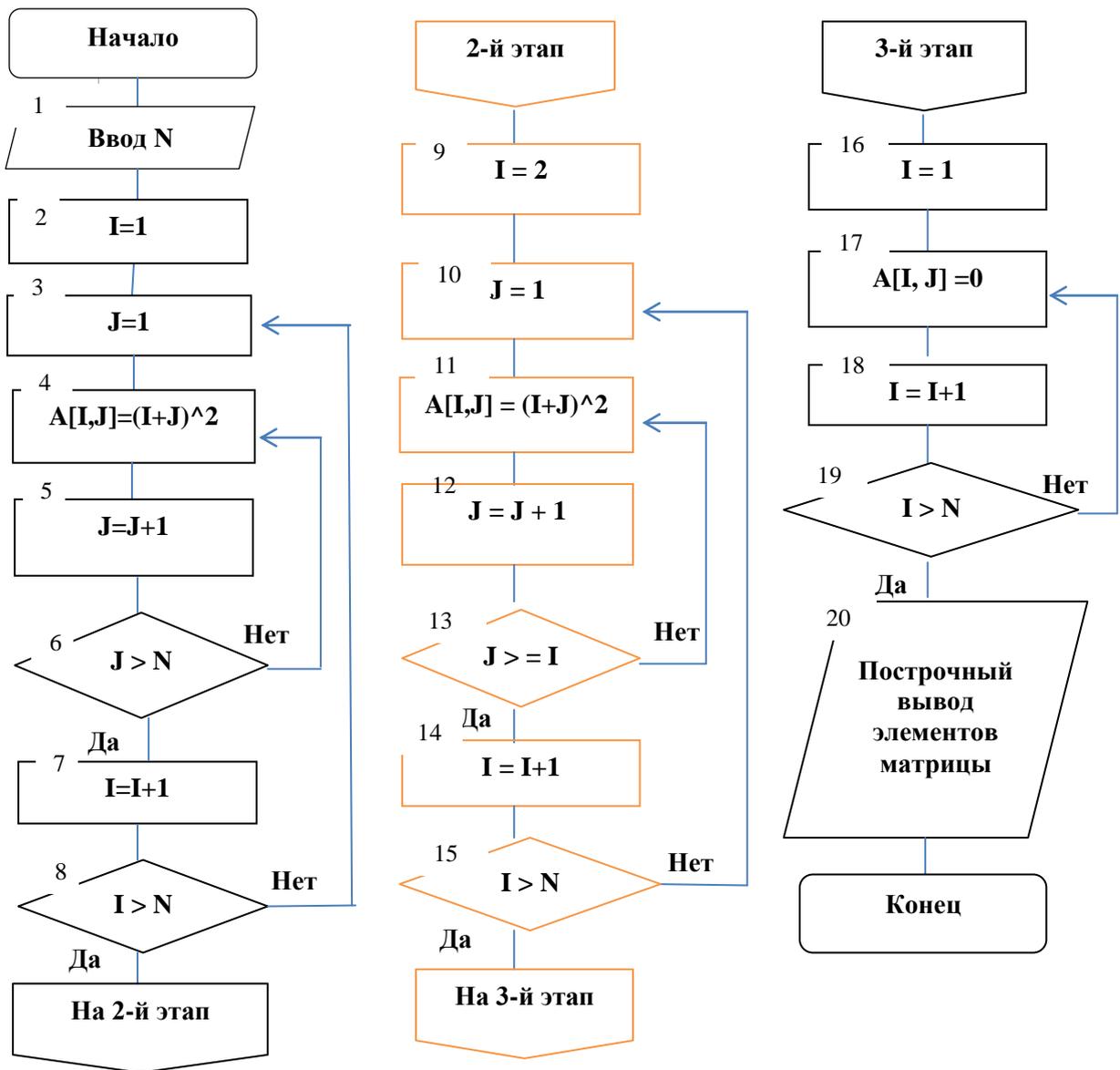
Пример 3. Даны действительные числа x_1, x_2, \dots, x_6 . Получить квадратную

матрицу седьмого порядка:

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



Пример 4. Пусть необходимо построить матрицу, элемент которой a_{ij} вычисляется как $(i + j)^2$, а затем зеркально отобразить её относительно главной диагонали. Элементом 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;
 - ListVox – 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;

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

- 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.

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.

```
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.

```

// 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
V:=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*mх),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 №4. Графические редакторы.

Цель: научиться создавать трёхмерные модели.

Создание таких трёхмерных объектов связано с перемещением плоских фигур (*эскизов*) в пространстве. Эскиз может быть построен в одной из стандартных плоскостей (XY, ZX, ZY), на плоской грани существующего тела или на дополнительной плоскости, которая построена пользователем. Эскизы создаются средствами плоского черчения.

Основные операции, которые можно выполнять с эскизами при построении трёхмерного объекта, следующие: *операция выдавливания* – выдавливание перпендикулярно плоскости эскиза; *операция вращения* – 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 (XY, ZX, ZY) для построения первого эскиза. Выберите плоскость XY .

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.

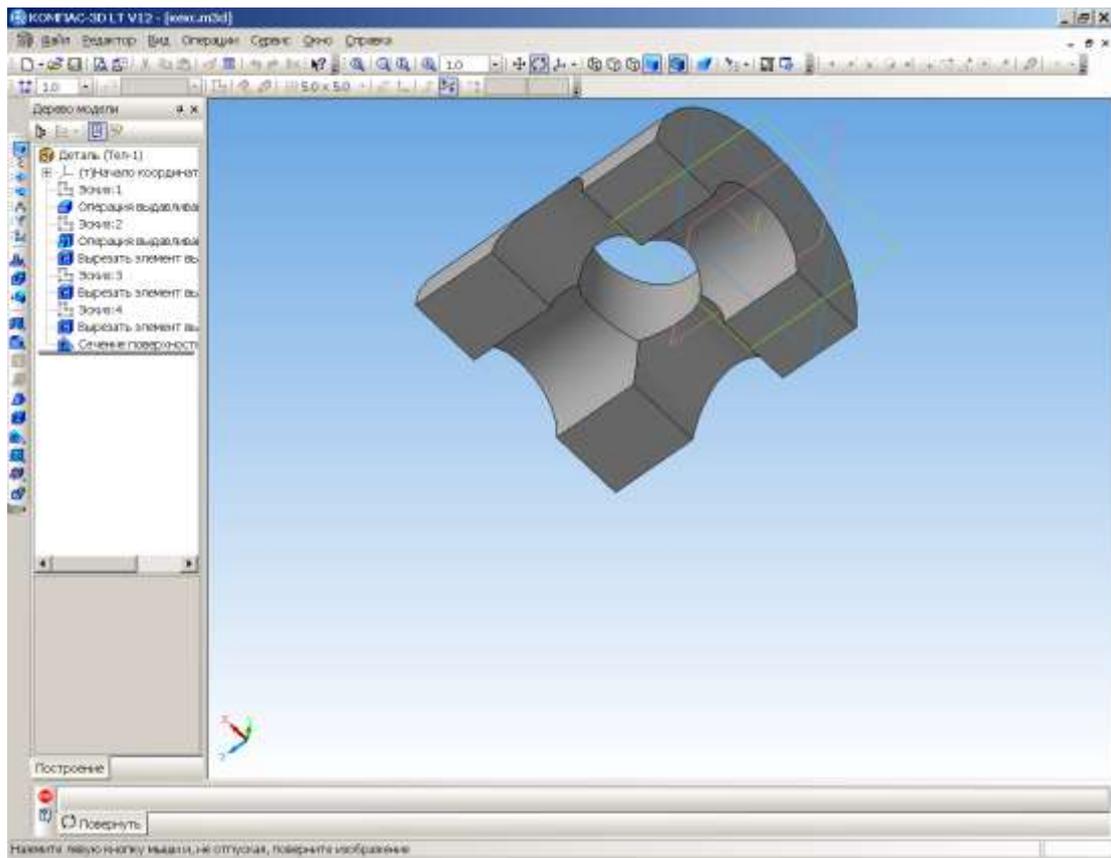
Вид детали на экране может быть задан в нескольких вариантах: *каркас, каркас без видимых линий, невидимые линии тонкие, полутонное или полутонное с каркасом.*

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.

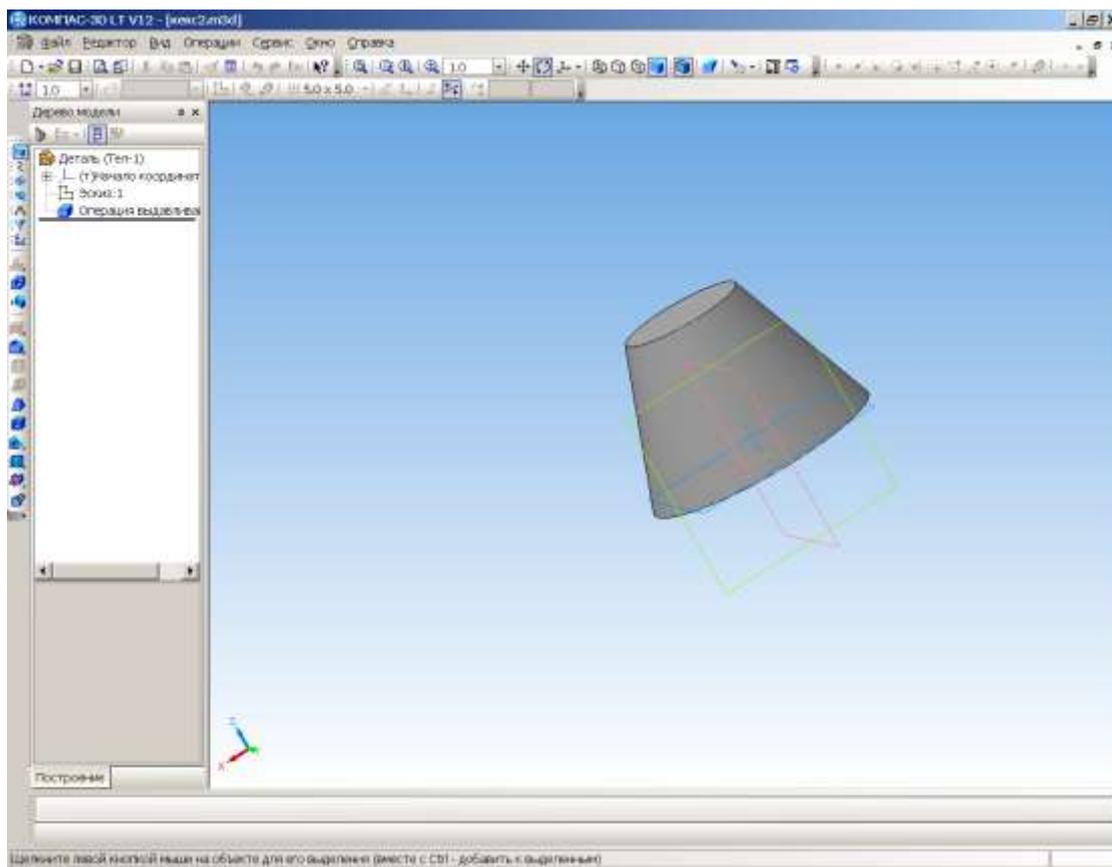
Цвет и оптические свойства детали установлены по умолчанию. Изменить эти параметры можно, выбрав в контекстном меню детали пункт **Свойства**.

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.



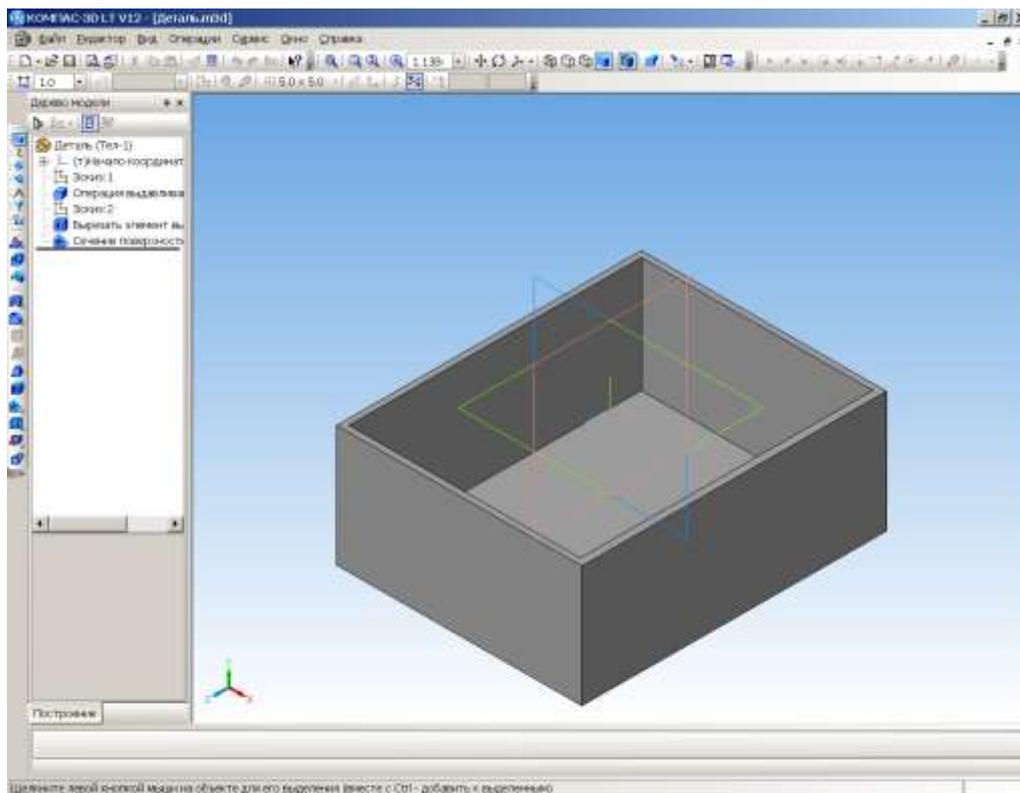
Задача 2

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.



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).

Компьютерные технологии вычислений. Системы Matlab, Scilab.

Цель: познакомиться с основными приёмами работы в системе MatLab.

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:

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:

```
>>V'  
>>V.'
```

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];  
>>V=[2 1 3;1 0 -2;0 5 3];  
>>c=[-4;3;2];  
>>a*V*c
```

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

```
>>C=A.*V  
>>D=A./V (деление элементов первой матрицы на соответствующие  
элементы второй матрицы)  
>>D=A.\V (деление элементов второй матрицы на соответствующие  
элементы первой)  
>>P=A.^2  
>>P1=A.^V
```

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: $x(t) = \frac{\sin t}{t+1} \quad y(t) = \frac{\cos t}{t+1}$.

```
>>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 `x2` и `x3`, расположенные около -2 и -5 .

Для того, чтобы увидеть больше значащих цифр корня `x1`, следует установить формат `long` и вывести `x1` 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.

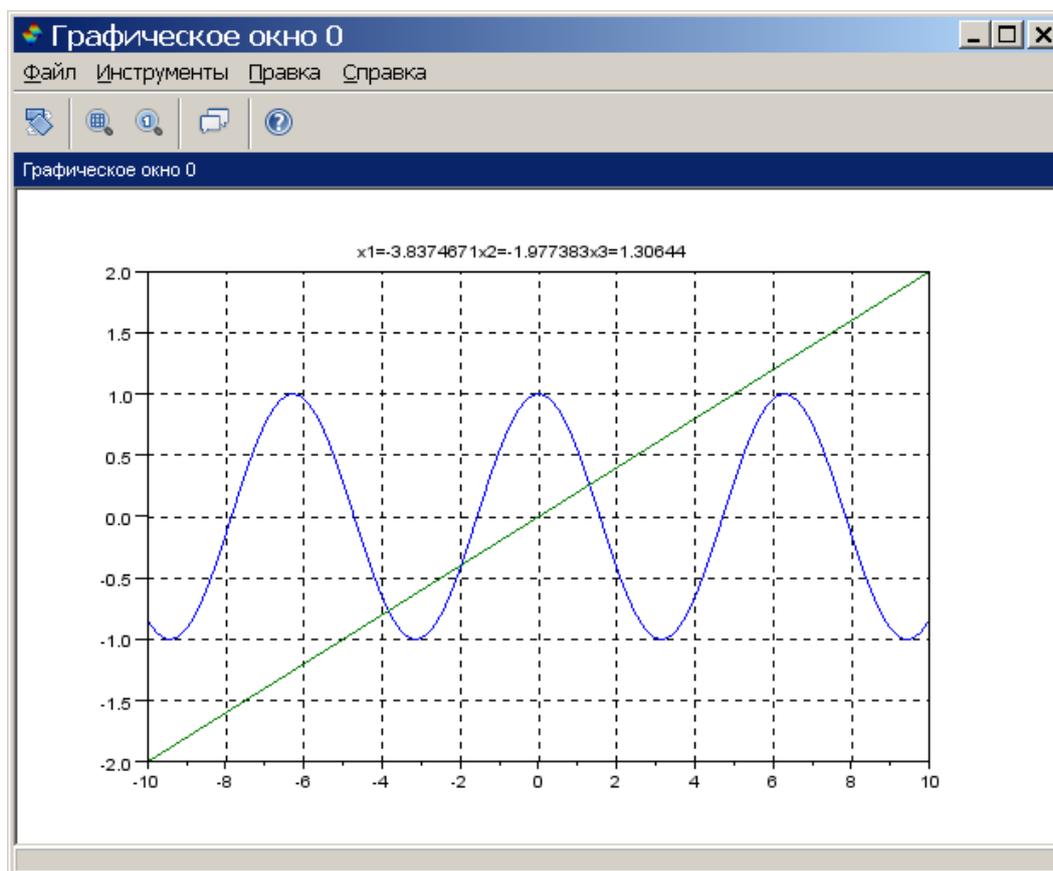
Пример 22. Построить график функции и вывести на экран корни уравнения

$$\cos(x) - 0,2x = 0$$

```

Коллекция окон
Файл Правка Управление Инструменты Справка
Графическое окно
-->function y1=f1(x);y1=cos(x);endfunction
-->function y2=f2(x);y2=0.2*x;endfunction
-->function y3=f3(x);y3=f1(x)-f2(x);endfunction
-->x=-10:0.1:10;
-->plot(x,f1(x),x,f2(x)); xgrid;
-->x0=-4;x1=fsolve(x0,f3)
-->x0=-2;x2=fsolve(x0,f3)
-->x0=1;x3=fsolve(x0,f3)
-->result='x1 = '+string(x1)+' x2= '+string(x2)+'x3= '+string(x3);
-->title(result);

```



Пример 23. Вычислить определённый интеграл $\int_0^1 e^{-x^2} dx$.

```

Клиентское окно
Файл Правка Управление Инструменты Справка
-->function y=f(x), y=exp(-x^2), endfunction;
-->[I,er]=intg(0,1,f)
er =
    8.2910-15
I =
    0.7468241
-->

```

Пример 24. Вычислить определённый интеграл $\int_0^1 e^{-x^2} dx$ заданный в виде таблицы по формуле трапеций [5].

x	e^{-x^2}	x	e^{-x^2}	x	e^{-x^2}
e^{-x^2}	1,000000	0,35	0,8847059	0,70	0,6126264
e^{-x^2}	0,9975031	0,40	0,8521438	0,75	0,5697828
e^{-x^2}	0,9900498	0,45	0,8166865	0,80	0,5272924
e^{-x^2}	0,9777512	0,50	0,7788008	0,85	0,4855369
e^{-x^2}	0,9607894	0,55	0,7389685	0,90	0,4448581
e^{-x^2}	0,9394131	0,60	0,6976763	0,95	0,4055545
e^{-x^2}	0,9139312	0,65	0,6554063	1,00	0,3678794

```

Клиентское окно
Файл Правка Управление Инструменты Справка
-->x=[0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55
-->y=[1 0.9975031 0.9900498 0.9777512 0.9607894 0.9394131
-->inttrap(x,y)
ans =
    0.7466708

```

Задания для самостоятельной работы:

5. Отделите корни уравнения $F(x) = 0$ графически и уточните один корень методом половинного деления в системе Scilab.

$$e^x + x^2 - 3 = 0$$

$$\ln(x-1) + 2x - 1 = 0$$

2. Решить задачу Коши $\frac{dx}{dt} + x = \sin(xt), \quad x(0) = 1.5$

3. Вычислить интеграл $\int_0^1 \frac{t^2}{\sqrt{3 + \sin(t)}} dt$

4. Построить график функции при помощи команды plot3d1: $Z = 5y^2 - x^2$.

Технология построения SIMULINK-моделей.

Цель: познакомиться с основными приёмами построения SIMULINK-моделей

Пример 1. Построить спираль Карно с применением следующих выражений¹⁹:

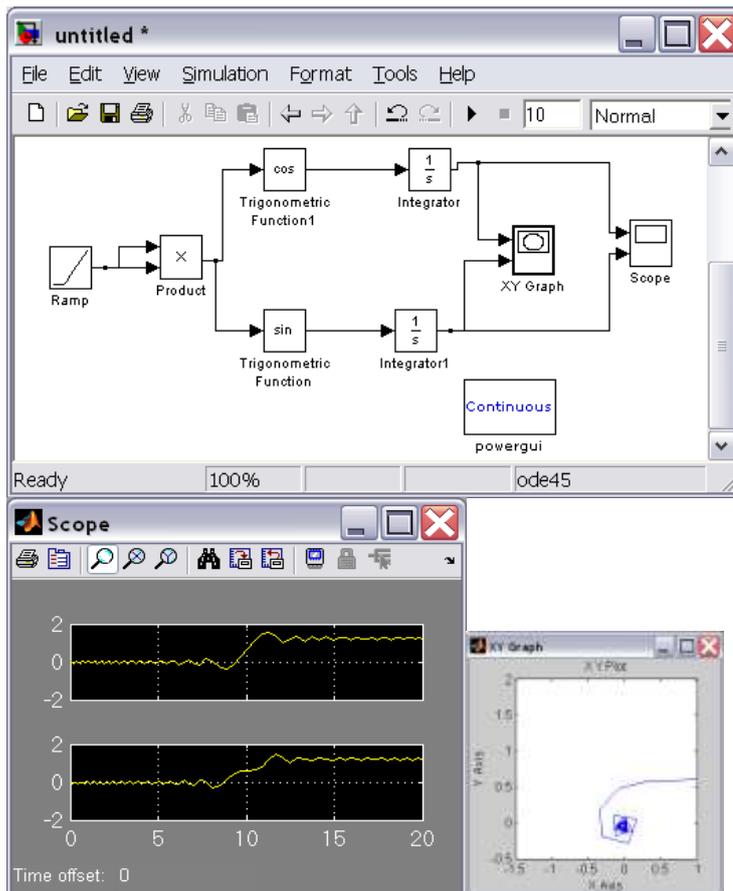
$$x(t) = \int_0^1 \cos(x^2) dt$$

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

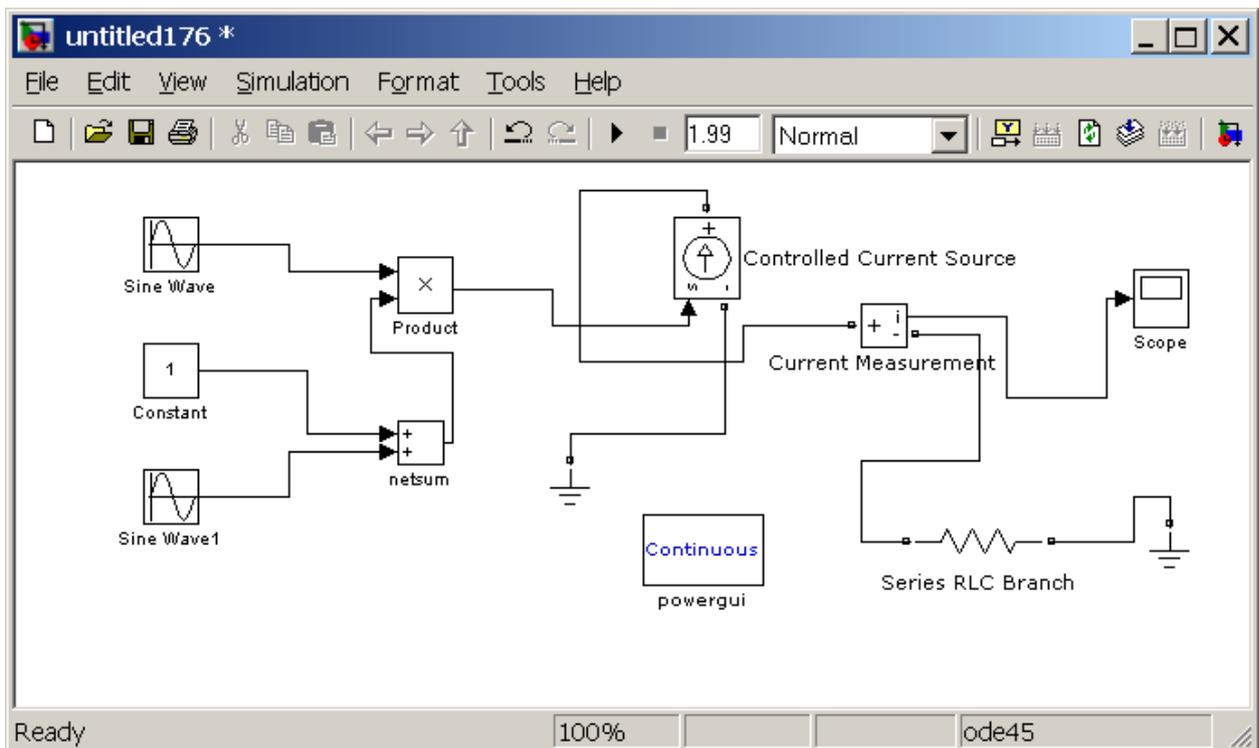
при изменении t от 0 до t_{\max} .

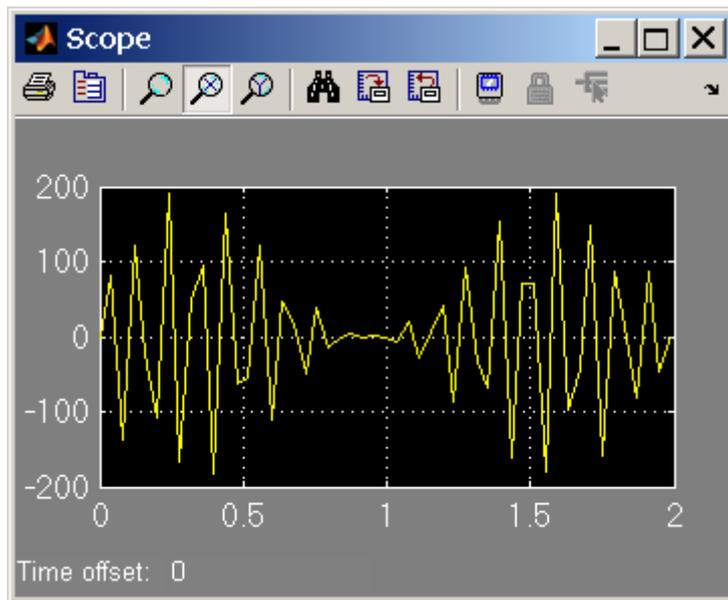
Текущее время формируется генератором нарастающего напряжения, которое начинается с уровня -10 для построения двух ветвей спирали. Затем этот сигнал возводится в квадрат и его синус и косинус интегрируются с помощью блока интеграторов.

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



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.

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://elib.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. Иопа. – 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. Иопа 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 : ВНУ, 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. : ВНУ – 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. Соболя. – Ростов н / Д: Феникс, 2009. – 509 [1] с. – (Высшее образование).

