

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ  
«КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМ. А.Н. ТУПОЛЕВА – КАИ»  
ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ЗАЩИТЫ ИНФОРМАЦИИ  
ОТДЕЛЕНИЕ СПО ИКТЗИ (КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ)

ЛАБОРАТОРНАЯ РАБОТА №6

ПО ДИСЦИПЛИНЕ: ИНФОРМАТИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГРУППЫ: 4243

ГЕРАСИМОВ СЕРГЕЙ АЛЕКСАНДРОВИЧ

ПРИНЯЛ

ПРЕПОДАВАТЕЛЬ: ТУМБИНСКАЯ М.В.

КАЗАНЬ 2020

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

## ТЕМА: «MS WORD»

- В результате выполнения лабораторной работы №1 я научился проверять и оценивать навыки работы с текстовым процессором «MS Word»

Первое что я сделал, выполнил «открытку о себе» ответил на заданные вопросы, прикрепил фотки, и указал информацию о себе.

### Задание 1.

**Меня зовут Герасимов Сергей Александрович.  
Я родился в городе Мамадыш.**



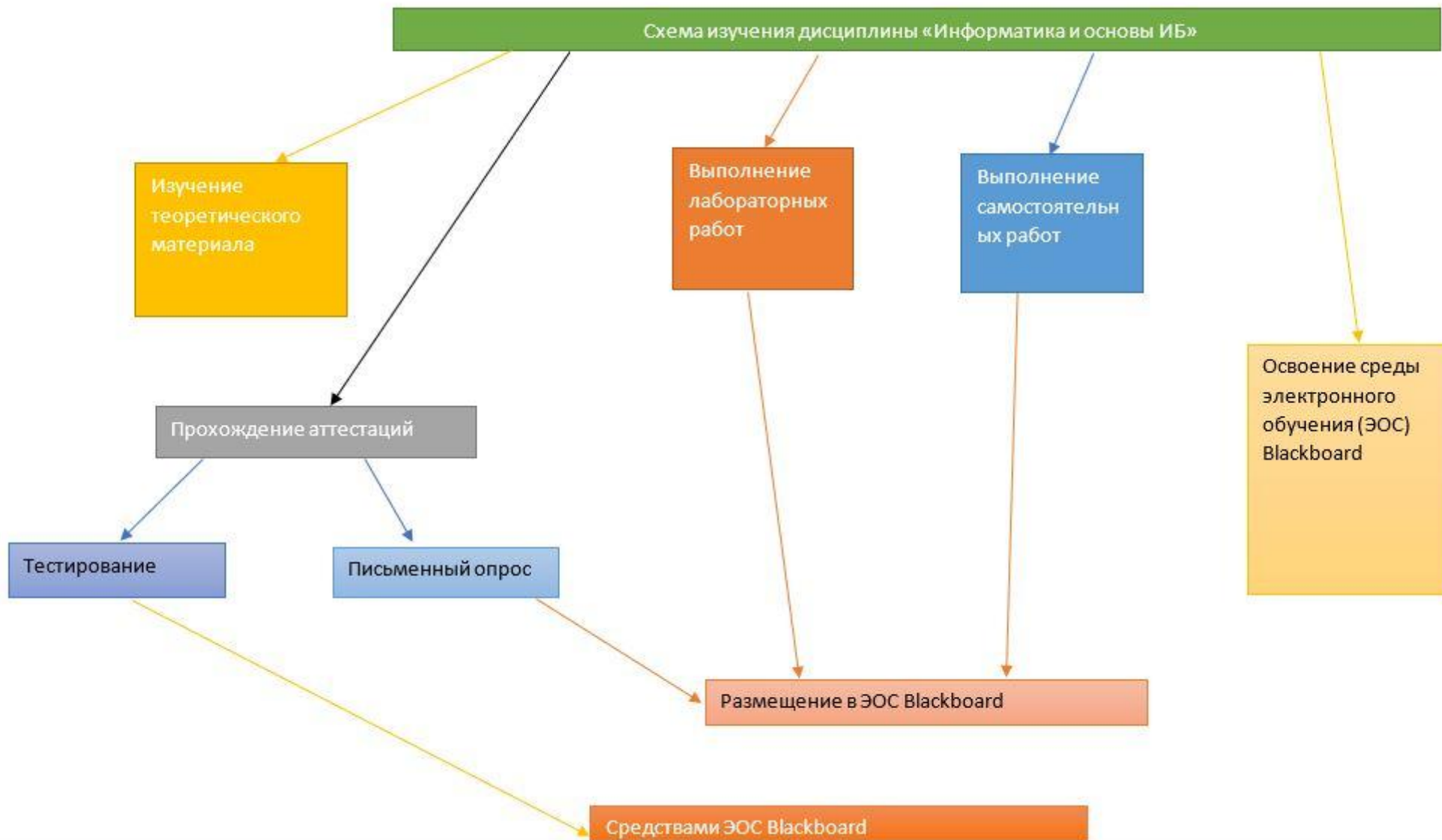
*Я учился в лицее №2, окончил учебу в 2019 году.*

Выбрал эту специальность, потому что люблю работать с компьютером и данная специальность довольно перспективна. Это учебное заведение мне посоветовал мой брат.

*В школе по информатике мы изучали строение компьютера, программное обеспечение компьютера и т.д.*

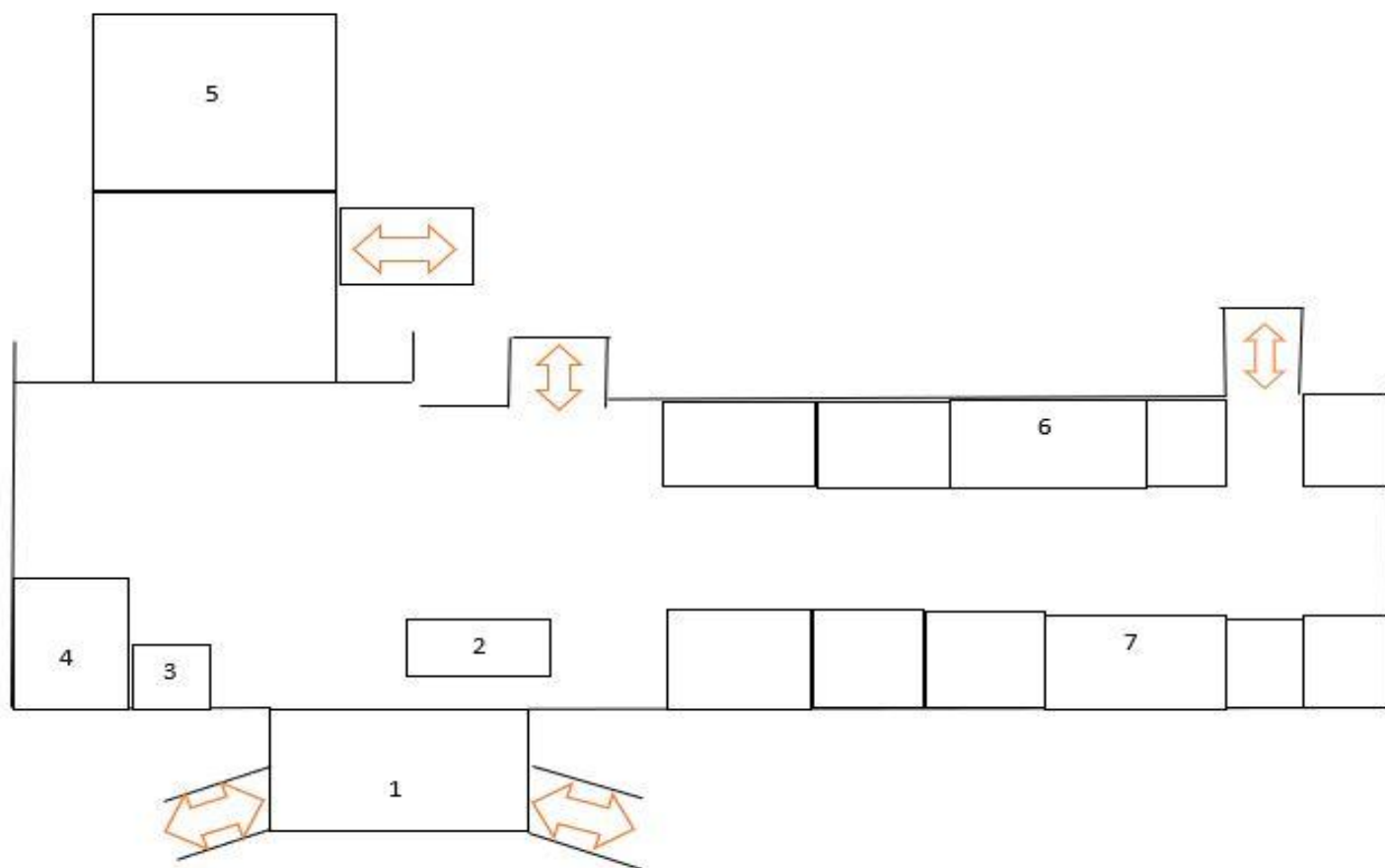
ВТОРЫМ МОИМ ШАГОМ СТАЛ ГРАФИЧЕСКИЙ РИСУНОК, ЕГО НУЖНО БЫЛО СГРУППИРОВАТЬ, И ОТФОРМАТИРОВАТЬ, ЧТОБЫ РИСУНОК БЫЛ ОБРАЗНЫМ.

### Задание 2.



В ТРЕТЬЕМ ЗАДАНИИ Я СДЕЛАЛ, ПЛАН ПЕРВОГО ЭТАЖА В КНИТУ – КАИ (7 ЗДАНИЕ), ПОСТАРАЛСЯ МАКСИМАЛЬНО ТОЧНО ПОКАЗАТЬ В ПЛАНЕ ВСЕ КОМНАТЫ И ПОМЕЩЕНИЯ, УКАЗАЛ ВСЕ ТУАЛЕТЫ, ЭТАЖИ, КОМНАТУ ДЛЯ МЕД. ОБСЛУЖИВАНИЯ. НИЖЕ ПЛАНА УКАЗАЛ СОКРАЩЕННОЕ НАИМЕНОВАНИЕ.

### Задание 3.



- 1 – вход
- 2 – пост охраны
- 3 - медпункт
- 4 - гардероб
- 5 - столовая
- 6 - деканат
- 7 - музей

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2 ТЕМА: ВЫПОЛНЕНИЕ ТАБЛИЦ В «MS WORD»

- В результате выполнения второй лабораторной работы я научился создавать и редактировать таблицы, выполнять формулы, строить схемы и группировать их между собой.
- В ходе выполнения первого задания мы сделали таблицы 1.19, 1.20. Далее выполнили таблицу 1.1

Месяц	Цех	Участок	Табельный номер	Категория	Вид оплат и удержаний	начисления по табельным	номерам начисления по видам	Удержания по видам оплат	Удержания по табельным	номерам Выдано в листы
2	2	2	3	1	2	5	5	5	5	5

Листок № на доплату к наряду	Дата	Цех	Участок	Заказ	Табельный Номер	Вид оплат
3	4	2	2	6	3	2
Деталь	Операция	Разряд работы	Шифр причин и виновника отключения	Нормы на доплату	Нормированное время	Сумма к оплате
5 цифровых и 3 алфавитных	2	1	3	норма всех расценки	3 4	5
Начальник цеха _____ Технолог _____					Всего времени нормированного по документу	Всего сумма к доплате

НАКЛАДНАЯ №						
ДАТА	ВИД ОПЕРАЦИЙ		СКЛАД-ПОЛУЧАТЕЛЬ	ЦЕХ ОТПРАВИТЕЛЬ		
P1	P2		P3	P4		
НАИМЕНОВАНИЕ МАТЕРИАЛА	НОМЕНКЛАТУРНЫЙ НОМЕР	ЕДИНИЦА ИЗМЕРЕНИЯ	КОЛИЧЕСТВО	ЦЕНА	СУММА	
			ОТПРАВЛЕННО	ПРИНЯТО		
P5	P6	P7	P8	P9	P10	P11

МАСТЕР \_\_\_\_\_  
P13

КЛАДОВЩИК \_\_\_\_\_  
P14

ВО ВТОРОМ ЗАДАНИИ МЫ ВЫПОЛНИЛИ ВСЕ ФОРМУЛЫ КОТОРЫЕ ПРИЛАГАЛИСЬ В ЗАДАНИИ И РАССЧИТАЛИ ИХ С ПОМОЩЬЮ MS WORD

### Задание 2

$$1) \int (af(x) + bg(x)) = a \int f(x)dx + b \int g(x)dx \quad \sqrt{(a^2 - x^2)(x^2 - b^2)}$$

$$2) \int f(x)dx = \int (g(y))g(y)dy$$

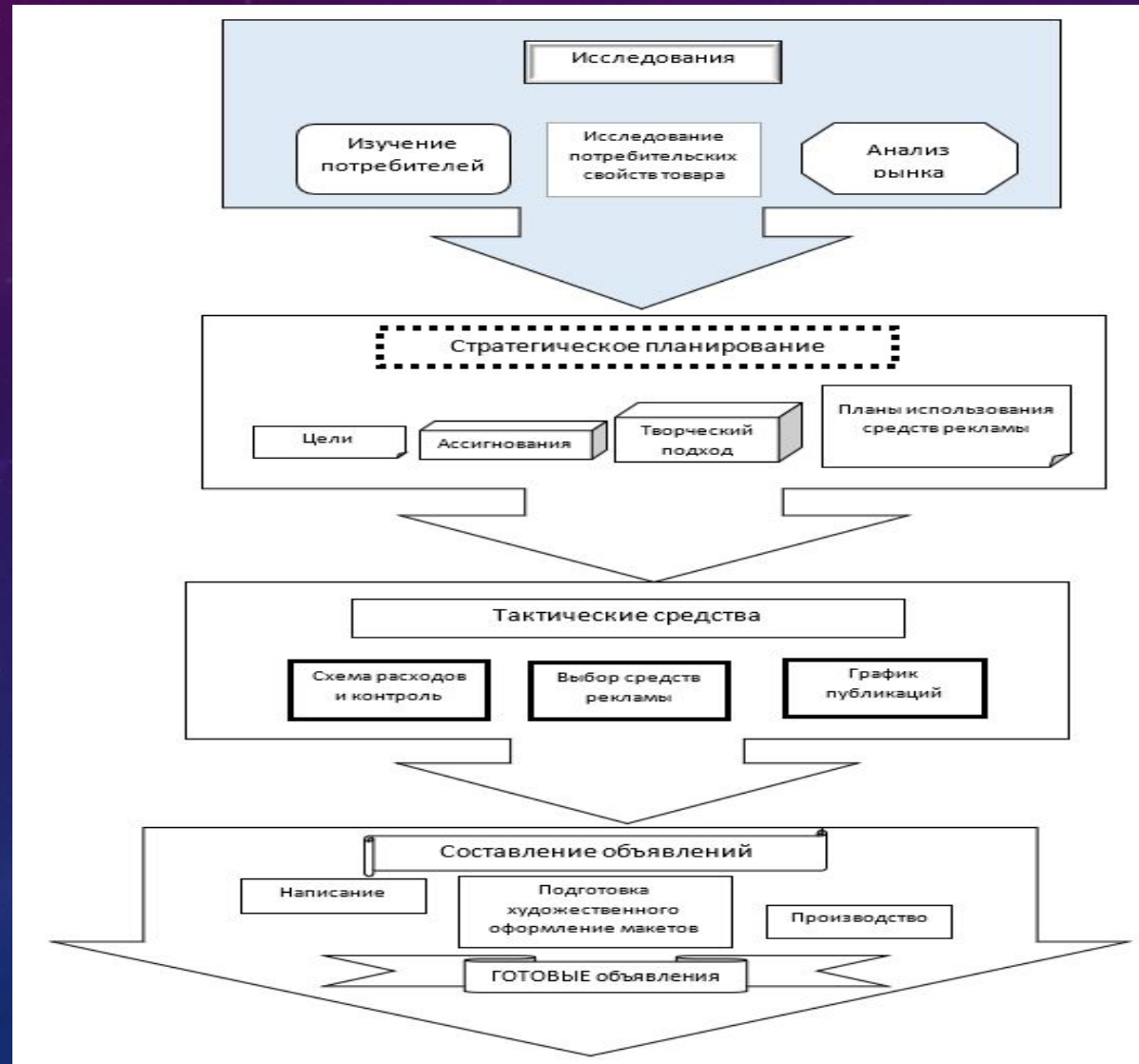
$$3) \int \frac{df(x)}{dx} g(x)f(x)dx = f(x)g(x) - \int f(x) \frac{dg(x)}{dx} dx$$

$$4) \int \frac{d}{dx} f(x)dx = f(x) \int (af(x) + bg(x))dx = a \int f(x) + b \int g(x)$$

$$5) \sum_{k=0}^{n-1} \cos \frac{2k+1}{2n+1} \pi \ln x$$

$$6) \frac{P(x)}{Q(x)} = \sum_{i=1}^m \frac{A^i}{x-a_i} \quad I = \frac{R_1(x)+R_2(x)\sqrt{P(x)}}{R_2(x)+R_4(x)\sqrt{P(x)}} dx$$

В ТРЕТЬЕМ ЗАДАНИИ МЫ СОЗДАЛИ СХЕМУ ОРГАНИЗАЦИЯ РЕКЛАМНОЙ ДЕЯТЕЛЬНОСТИ БАНКА ПО ПРОДВИЖЕНИЮ УСЛУГИ» СОБЛЮДАЯ ИХ ВНЕШНИЙ ВИД И СГРУППИРОВАЛИ ИХ. ПОЛЬЗОВАЛИСЬ ПРИ ЭТОМ WORD ART В «MS WORD»



# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

## ТЕМА: РАБОТА В MS EXCEL. ВЫЧИСЛЕНИЕ И ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В СИСТЕМАХ СЧИСЛЕНИЯ С ПРОИЗВОЛЬНЫМИ ОСНОВАНИЯМИ.

### ВЫПОЛНЯЛ 3 ВАРИАНТ.

- В результате выполнения лабораторной работы №3 я научился работать в MS Excel. Создавать таблицы, и делать небольшие вычисления.

**Лабораторная работа №**  
**Работа в MS EXCEL. Вычисление и представление чисел в системах счисления с произвольными основаниями.**

В общем случае *система счисления* представляет собой совокупность приемов и правил для записи чисел цифровыми знаками. Все системы счисления делятся на два больших класса – непозиционные и позиционные.

В *непозиционной* системе счисления значения символов не зависят от положения в числе. Примером непозиционной системы счисления является римская система, использующая набор следующих символов: I, V, X, L, C, D и т.д.

В *позиционной* системе счисления значение каждой цифры изменяется от ее положения (или позиции) в числе.

Десятичная система счисления наиболее распространена в вычислительной практике. Однако своему распространению она обязана не каким-то особым преимуществом перед другими системами, а достаточно случайному обстоятельству – наличию у человека десяти пальцев на руках.

Количество различных цифр, применяемых в позиционной системе счисления, называется ее *основанием*.

Рассмотрим наиболее широко используемые системы счисления.

**Десятичная система:**  
Алфавит: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9  
Основание (количество цифр): 10

сотни    десятки    единицы  
          разряды

$$\begin{array}{ccc} 3 & 7 & 8 \\ \downarrow & \downarrow & \downarrow \\ 3 \cdot 10^2 & + 7 \cdot 10^1 & + 8 \cdot 10^0 \\ \hline 300 & 70 & 8 \end{array}$$

**Двоичная система счисления:**  
Алфавит: 0, 1  
Основание (количество цифр): 2

$10 \rightarrow 2$

$$\begin{array}{r} 19 \ 2 \\ 18 \ 2 \\ \hline 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \end{array}$$

$2 \rightarrow 10$

$$\begin{array}{r} 4 \ 3 \ 2 \ 1 \ 0 \text{ разряды} \\ 10011_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ = 16 + 2 + 1 = 19 \end{array}$$

**Восьмеричная система счисления:**  
Алфавит: 0, 1, 2, 3, 4, 5, 6, 7  
Основание (количество цифр): 8

$10 \rightarrow 8$

$$\begin{array}{r} 100 \ 8 \\ 96 \ 8 \\ \hline 4 \ 4 \ 0 \ 0 \end{array}$$

$8 \rightarrow 10$

$$\begin{array}{r} 2 \ 1 \ 0 \text{ разряды} \\ 144_8 = 1 \cdot 8^2 + 4 \cdot 8^1 + 4 \cdot 8^0 \\ = 64 + 32 + 4 = 100 \end{array}$$

**Шестнадцатеричная система счисления:**  
Основание (количество цифр): 16  
Алфавит: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

$10 \rightarrow 16$

$$\begin{array}{r} 107 \ 16 \\ 96 \ 16 \\ \hline 11 \ 0 \ 0 \end{array}$$

$16 \rightarrow 10$

$$\begin{array}{r} 2 \ 1 \ 0 \text{ разряды} \\ 1C5_{16} = 1 \cdot 16^2 + 12 \cdot 16^1 + 5 \cdot 16^0 \\ = 256 + 192 + 5 = 453 \end{array}$$

В табл.1 приведены эквиваленты десятичных чисел в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Таблица 1

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

**Задание №1.0.**  
Рассмотрим выполнение данного задания.

- Перевести десятичное число 371 в двоичную, восьмеричную и шестнадцатеричную системы счисления.
- Перевести число 111000.11 из двоичной системы счисления в десятичную.
- Перевести восьмеричное и шестнадцатеричное число 32 в десятичное.

Порядок выполнения:  
**Перевод из десятичной системы счисления в двоичную:**  
Будем использовать метод последовательного деления на основание.

- Отведем первый столбец (столбец А) для деления на основание системы счисления
- Для этого в ячейку А1 введем значение 371. В ячейке А2 должно находиться значение 185,5 которое получается в результате деления исходного числа на 2, поэтому в ячейку А2 введем



МЫ ОТВЕДЕМ ПЕРВЫЙ СТОЛБЕЦ (СТОЛБЕЦ А) ДЛЯ ДЕЛЕНИЯ НА ОСНОВАНИЕ СИСТЕМЫ СЧИСЛЕНИЯ 2. ДЛЯ ЭТОГО В ЯЧЕЙКУ А1 ВВЕДЕМ ЗНАЧЕНИЕ 12736. В ЯЧЕЙКЕ А2 ДОЛЖНО НАХОДИТЬСЯ ЗНАЧЕНИЕ 6368,0 КОТОРОЕ ПОЛУЧАЕТСЯ В РЕЗУЛЬТАТЕ ДЕЛЕНИЯ ИСХОДНОГО ЧИСЛА НА 2, ПОЭТОМУ В ЯЧЕЙКУ А2 ВВЕДЕМ. ВЫДЕЛИМ ЯЧЕЙКУ С1, УСТАНОВИМ КУРСОР В НИЖНЕМ ПРАВОМ УГЛУ ЯЧЕЙКИ, КАК ТОЛЬКО КУРСОР ПРИМЕТ ИЗОБРАЖЕНИЕ КРЕСТИКА, НАЖАВ НА ЛЕВУЮ КНОПКУ МЫШКИ, ПЕРЕМЕСТИМ СОДЕРЖИМОЕ ЯЧЕЙКИ С1 НА ЯЧЕЙКИ С2:С16. В РЕЗУЛЬТАТЕ ПОЛУЧАЕТСЯ ИСКОМОЕ ЧИСЛО В ДВОИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ, ПРИЧЕМ РЕЗУЛЬТАТ ЯЧЕЙКИ С16 ЯВЛЯЕТСЯ СТАРШИМ РАЗРЯДОМ ДВОИЧНОГО ЧИСЛА, А С1 – МЛАДШИМ РАЗРЯДОМ

	А	В	С
1	12736	12 736	0
2	6368,0	6 368	0
3	3184,0	3 184	0
4	1592,0	1 592	0
5	796,0	796	0
6	398,0	398	0
7	199,0	199	1
8	99,5	99	1
9	49,8	49	1
10	24,9	24	0
11	12,4	12	0
12	6,2	6	0
13	3,1	3	1
14	1,6	1	1
15	0,8	0	0
16	0,4	0	0

В ПЕРВОМ ЗАДАНИИ Я ПЕРЕВЕЛ ДЕСЯТИЧНОЕ ЧИСЛО 1564 В ДВОИЧНУЮ, ВОСЬМЕРИЧНУЮ И ШЕСТНАДЦАТЕРИЧНУЮ СИСТЕМЫ СЧИСЛЕНИЯ. С ПОМОЩЬЮ ФОРМУЛ В MS EXCEL И СДЕЛАЛ ВСЕ В ВИДЕ ТАБЛИЦЫ

	A	B	C
1	12736	12 736	0
2	6368,0	6 368	0
3	3184,0	3 184	0
4	1592,0	1 592	0
5	796,0	796	0
6	398,0	398	0
7	199,0	199	1
8	99,5	99	1
9	49,8	49	1
10	24,9	24	0
11	12,4	12	0
12	6,2	6	0
13	3,1	3	1
14	1,6	1	1
15	0,8	0	0
16	0,4	0	0

0	1	1	0	0	0	1	1	1	0	0	0	0	0	0
3			0			7			0			0		

0	0	1	1	0	0	0	1	1	1	0	0	0	0	0
3			1				12			0				

ВО ВТОРОМ ЗАДАНИИ Я ПЕРЕВЕЛ ЧИСЛО 111011000.11 ИЗ ДВОИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДЕСЯТИЧНУЮ. С ПОМОЩЬЮ ФОРМУЛ:  $=1 * \text{СТЕПЕНЬ}(2;8) + 1 * \text{СТЕПЕНЬ}(2;7) + 1 * \text{СТЕПЕНЬ}(2;6) + 0 * \text{СТЕПЕНЬ}(2;5) + 1 * \text{СТЕПЕНЬ}(2;4) + 1 * \text{СТЕПЕНЬ}(2;3) + 0 * \text{СТЕПЕНЬ}(2;2) + 0 * \text{СТЕПЕНЬ}(2;1) + 0 * \text{СТЕПЕНЬ}(2;0) + 1 * \text{СТЕПЕНЬ}(2;-1) + 1 * \text{СТЕПЕНЬ}(2;-2)$

$=1 * \text{СТЕПЕНЬ}(2;8) + 1 * \text{СТЕПЕНЬ}(2;7) + 1 * \text{СТЕПЕНЬ}(2;6) + 0 * \text{СТЕПЕНЬ}(2;5) + 1 * \text{СТЕПЕНЬ}(2;4) + 1 * \text{СТЕПЕНЬ}(2;3) + 0 * \text{СТЕПЕНЬ}(2;2) + 0 * \text{СТЕПЕНЬ}(2;1) + 0 * \text{СТЕПЕНЬ}(2;0) + 1 * \text{СТЕПЕНЬ}(2;-1) + 1 * \text{СТЕПЕНЬ}(2;-2)$

**111011000.11**

**472,75**

В ТРЕТЬЕМ ЗАДАНИИ Я ПЕРЕВЕЛ ВОСЬМЕРИЧНОЕ И ШЕСТНАДЦАТЕРИЧНОЕ ЧИСЛО 732577 В ДЕСЯТИЧНОЕ С ПОМОЩЬЮ ФОРМУЛЫ:  $=7*СТЕПЕНЬ(8;5)+3*СТЕПЕНЬ(8;4)+2*СТЕПЕНЬ(8;3)+5*СТЕПЕНЬ(8;2)+7*СТЕПЕНЬ(8;1)+7*СТЕПЕНЬ(8;0)$

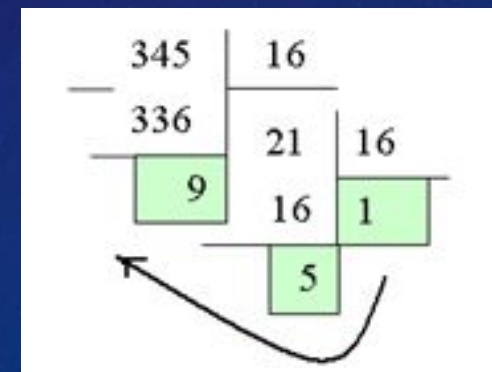
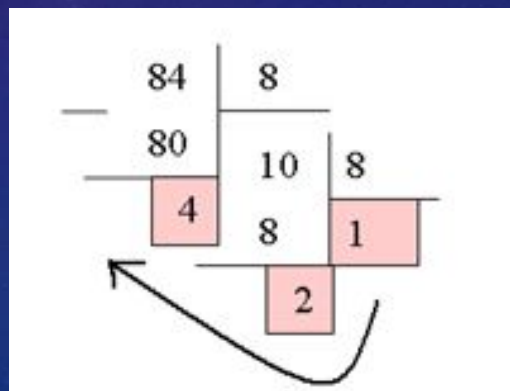
$=7*СТЕПЕНЬ(8;5)+3*СТЕПЕНЬ(8;4)+2*СТЕПЕНЬ(8;3)+5*СТЕПЕНЬ(8;2)+7*СТЕПЕНЬ(8;1)+7*СТЕПЕНЬ(8;0)$	
732577	732577
243071	7546231

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4 ТЕМА: СИСТЕМА СЧИСЛЕНИЯ

- В результате выполнения лабораторной работы №4 я научился решать системы счисления переводить из одной в другую. Все системы счисления делятся на два больших класса – непозиционные и позиционные.
- В непозиционной системе счисления значения символов не зависят от положения в числе. Для образования таких систем используют, в основном, операции сложения и вычитания. Например, система с одним символом-палочкой встречалась у многих народов. Для изображения числа в этой системе нужно записать определенное множество палочек, равное данному числу. Эта система не эффективна, т.к. запись числа получается длинной. Другим примером непозиционной системы счисления является римская система.
- Основными числами в римской системе счисления являются 1, 5, 10, 50, 100, 500, 1000, которые обозначаются следующими знаками: I, V, X, L, C, D, M, а другие получаются путем сложения и вычитания базисных: если цифра слева меньше, чем цифра справа, то левая цифра вычитается из правой. Так, например, число 146 в римской системе счисления имеет вид CXLVI (C – 100, XL – 40, VI – 6), здесь «сорок» получается посредством вычитания из «пятидесяти» числа «десять», «шесть» - посредством сложения «пяти» и «единицы».
- В позиционной системе с некоторым основанием  $p$  используются  $p$  различных между собой символов от 0 до  $p-1$ . Так последовательность символов

ПУСТЬ ИСХОДНОЕ ЧИСЛО  $AP$  СОДЕРЖИТ  $M$  РАЗРЯДОВ. ДЛЯ ТОГО ЧТОБЫ ЦЕЛОЕ ЧИСЛО  $AP$  ПЕРЕВЕСТИ В ДЕСЯТИЧНУЮ СИСТЕМУ СЧИСЛЕНИЯ, ДОСТАТОЧНО ЦИФРУ СТАРШЕГО РАЗРЯДА ЧИСЛА  $AP$  УМНОЖИТЬ НА ОСНОВАНИЕ СИСТЕМЫ СЧИСЛЕНИЯ  $P$ , ВОЗВЕДЕННОЕ В СТЕПЕНЬ СООТВЕТСТВУЮЩУЮ НОМЕРУ ПОЗИЦИИ ЦИФРЫ ДАННОГО РАЗРЯДА ЗА МИНУСОМ ЕДИНИЦЫ ( $M-1$ ). К ПОЛУЧЕННОМУ ПРОИЗВЕДЕНИЮ ПРИБАВИТЬ ЦИФРУ СЛЕДУЮЩЕГО (БОЛЕЕ МЛАДШЕГО) РАЗРЯДА ЧИСЛА  $AP$ , УМНОЖЕННУЮ НА ОСНОВАНИЕ  $P$ , ВОЗВЕДЕННОЕ В СТЕПЕНЬ  $M-2$ .

- Перевести число  $X=9A07,C8_{16}$  в десятичную систему счисления, пользуясь правилами десятичной арифметики.
- Для перевода воспользуемся формулой (1.1) и запишем число  $9A07,C8_{16}$  в виде:
- $9A07,C8_{16} = 9 \cdot 16^3 + 10 \cdot 16^2 + 0 \cdot 16^1 + 7 \cdot 16^0 + 12 \cdot 16^{-1} + 8 \cdot 16^{-2}$
- Выполним все необходимые действия в десятичной системе и получим ответ:
- $X = 36864 + 2560 + 0 + 7 + 0,75 + 0,03125$
- Ответ:  $9A07,C8_{16} = 39431,78125_{10}$



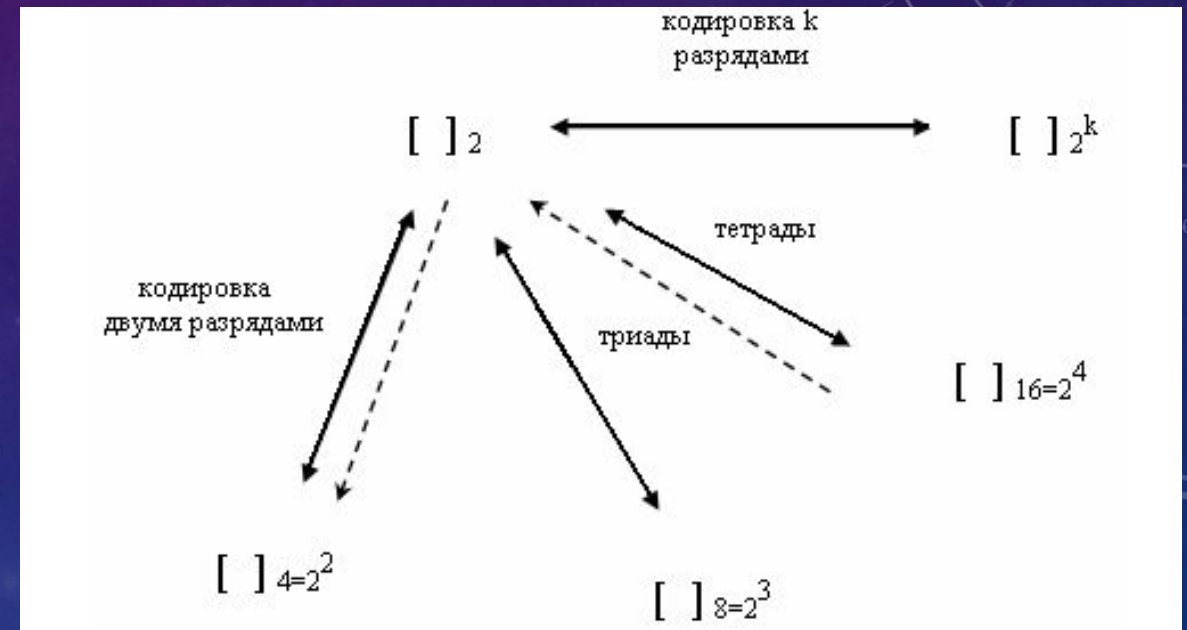
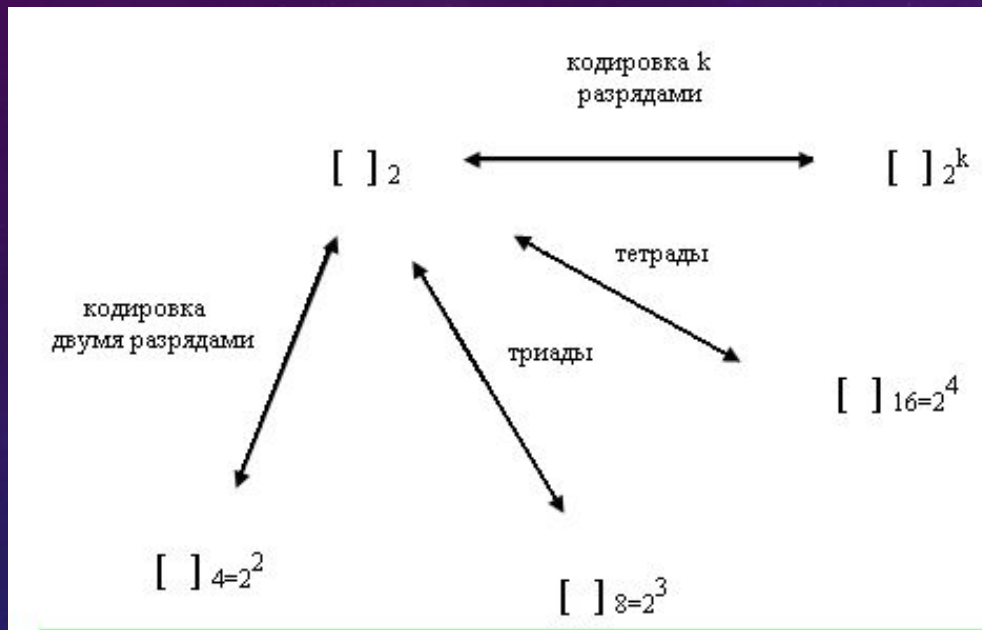
2, 8, 16-РИЧНЫЕ СИСТЕМЫ СЧИСЛЕНИЯ, НАМ ПОНАДОБЯТСЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ

2 с.с.	4 с.с.
00	0
01	1
10	2
11	3

2 с.с.	8 с.с.
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

2 с.с.	16 с.с.
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

ПЕРЕВОД ЧИСЕЛ В СИСТЕМАХ СЧИСЛЕНИЯ С ОСНОВАНИЯМИ КРАТНЫМИ ДВУМ В ЭТОМ ПАРАГРАФЕ МЫ ОСВОИМ БЫСТРЫЙ ПЕРЕВОД ЧИСЕЛ, ОСНОВАНИЕ КОТОРЫХ КРАТНО ДВУМ  $2 = 2^1$ ,  $4 = 2^2$ ,  $8 = 2^3$ ,  $16 = 2^4$  И Т.Д. В СЛУЧАЕ, КОГДА ПРОИСХОДИТ ПЕРЕВОД ЧИСЕЛ С ОДНИМ КРАТНЫМ ОСНОВАНИЕМ В ДРУГОЕ КРАТНОЕ ОСНОВАНИЕ, ПРОЦЕСС ПЕРЕВОДА ЗАМЕНЯЕТСЯ КОДИРОВКОЙ.





## 8 ВАРИАНТ

- В ходе выполнения 1-ого задания записать в десятичной системе следующие числа  $2FB_{16} = \text{---}_{10}$  я выполнил все расчеты в тетради
- В 2 задании нужно перевести десятичное число в заданную систему счисления:  $63_{10} = \text{---}_5$

~~1-ое задание~~ задание 1. 474

1)  $2FB_{16} = 763_{10}$

$$2 \cdot 16^2 + 15 \cdot 16^1 + 11 \cdot 16^0 = 512 + 240 + 11 = 763$$

2)  $63 = 223_5$

$$\begin{array}{r|l} 63 & 5 \\ \hline 50 & 12 \quad 3 \\ \hline 13 & 2 \quad 3 \\ \hline 3 & 1 \quad 2 \end{array}$$

В 3 ЗАДАНИИ НУЖНО ПРЕОБРАЗОВАТЬ ЧИСЛА В УКАЗАННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ:

$$112_3 = \text{---}_{14}$$

3)  $112_3 = \text{---}_{14}$

$$112_3 = 1 \cdot 3^2 + 1 \cdot 3^1 + 2 \cdot 3^0 = 9 + 3 + 2 = 14_{10}$$

14		14
-14		0
<hr/>		
0		1

В 4 ЗАДАНИИ НУЖНО ПРЕОБРАЗОВАТЬ ЧИСЛА В УКАЗАННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ:  $10111_2 =_{-10}$   
 $10111_2 =_{-16}$

Handwritten work on grid paper showing the conversion of the binary number 10111 to decimal and hexadecimal.

4)  $10111_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$   
 $16 + 0 + 4 + 2 + 1 = 23_{10}$

23	16
16	1
<hr/>	
7	

$10111_2 = 17_{16}$

В 5 ЗАДАНИИ НУЖНО ПРЕОБРАЗОВАТЬ ДЕСЯТИЧНЫЕ ЧИСЛА В ДВОИЧНЫЕ И ВОСЬМЕРИЧНЫЕ:

$$109 =_{-2} =_{-8}$$

Handwritten conversion of 109 to binary using the division-by-2 method. The division steps are as follows:

5	109	2
-	108	54
	1	2
		27
		13
		6
		3
		1

The resulting binary number is  $109 = 1101101_2$ .

Handwritten conversion of 109 to octal using the division-by-8 method. The division steps are as follows:

109	8	
-	104	13
	5	1

The resulting octal number is  $109 = 155_8$ .

В 8 задании нужно перевести шестнадцатеричные числа в четверичные:

$$1FFE_{16} =_{10}$$

8)  $1FFE_{16} = 1 \cdot 16^3 + 15 \cdot 16^2 + 15 \cdot 16^1 + 14 \cdot 16^0 =$   
 $4096 + 3840 + 240 + 14 = 8190_{10}$

$1FFE_{16} = 1333332_4$

8190 / 4  
8188 / 2047 / 4  
2 2044 511 / 4  
3 508 127 / 4  
3 724 151 / 4  
3 282 3 / 4  
3 24 3 / 4  
3 12 3 / 4  
3 6 3 / 4  
3 3 3 / 4  
3 1 3 / 4

**АЛГОРИТМИЗАЦИЯ** – ЭТО ПРЕДСТАВЛЕНИЕ НЕФОРМАЛЬНОГО, НЕТОЧНОГО ИЛИ НЕПОЛНОГО ОПИСАНИЯ

ИЗВЕСТНОГО МЕТОДА РЕШЕНИЯ ЗАДАЧИ В ВИДЕ ЧЕТКОГО АЛГОРИТМА.

**АЛГОРИТМ** – ЭТО ОПИСАНИЕ ТОЧНО ОПРЕДЕЛЕННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ОПЕРАЦИЙ, НАПРАВЛЕННОЙ НА ДОСТИЖЕНИЕ ЗАДАННОЙ ЦЕЛИ ИЛИ

- **Решения поставленной задачи** следующим свойствам:
- **Массовость** - алгоритм должен позволять решать не одну конкретную задачу, а множество задач этого класса.
- **Определенность** – алгоритм должен быть однозначным, исключая произвольность толкования любого из предписаний и заданного порядка исполнения.
- **Результативность** (конечность) – прекращение процесса через определенное число шагов с выдачей искомым результатов или сообщения о невозможности продолжения вычислительного процесса.
- Как правило, алгоритмы разрабатываются укрупненными блоками и только для некоторого типа задач необходима детализация этих укрупненных блоков. Для таких ситуаций используется свойство дискретности.
- **Дискретность** – расчленение сложных операций на более легкие.
- Существуют основные способы записи алгоритмов:
  - словесный
  - формульно-словесный
  - с помощью блок-схем
- **Словесный** – содержание этапов вычислений задается на естественном языке в произвольной форме с требуемой детализацией.
- Рассмотрим пример словесной записи алгоритма для решения следующей задачи.

ФОРМУЛЬНО-СЛОВЕСНЫМ ЗАДАНИЕ ИЛИ ПУКЦИОМ С ИСПОЛЬЗОВАНИЕМ МАТЕМАТИЧЕСКИХ СИМВОЛОВ И ВЫРАЖЕНИЙ В СОЧЕТАНИИ СО СЛОВЕСНЫМИ ПОЯСНЕНИЯМИ.

РАССМОТРИМ ПРИМЕР ФОРМУЛЬНО-СЛОВЕСНОЙ ЗАПИСИ АЛГОРИТМА ВЫЧИСЛЕНИЯ ПЛОЩАДИ ТРЕУГОЛЬНИКА ПО ТРЕМ СТОРОНАМ.

**БЛОК-СХЕМА** - ЭТО ГРАФИЧЕСКОЕ ИЗОБРАЖЕНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ АЛГОРИТМА, В КОТОРОМ КАЖДЫЙ ЭТАП ПРОЦЕССА ПЕРЕРАБОТКИ ДАННЫХ ПРЕДСТАВЛЯЕТСЯ В ВИДЕ ГЕОМЕТРИЧЕСКИХ ФИГУР (БЛОКОВ), ИМЕЮЩИХ ОПРЕДЕЛЕННУЮ КОНФИГУРАЦИЮ В ЗАВИСИМОСТИ ОТ ХАРАКТЕРА ВЫПОЛНЯЕМЫХ ОПЕРАЦИЙ.

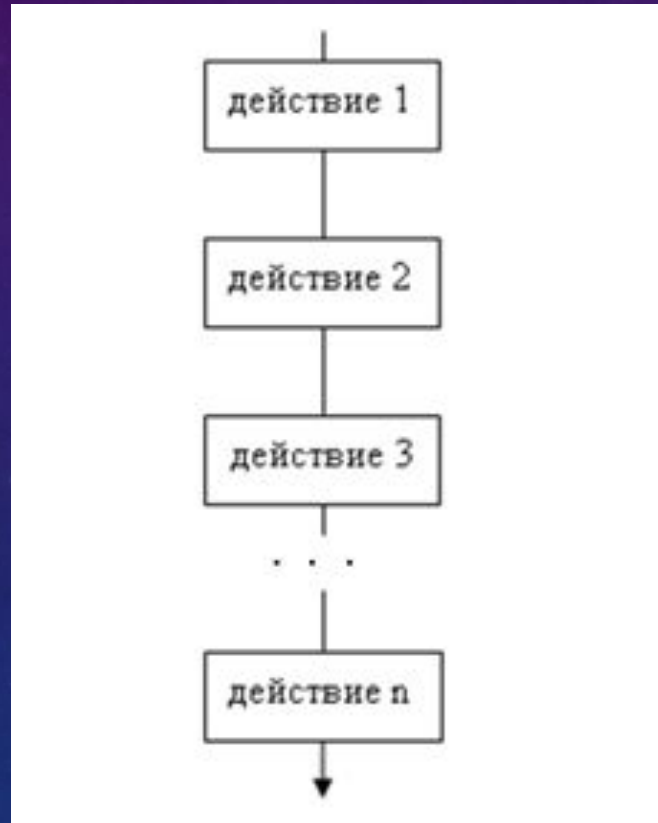
ОСНОВНЫЕ СТРУКТУРЫ АЛГОРИТМОВ:

- ПОСЛЕДОВАТЕЛЬНОСТЬ (ЛИНЕЙНАЯ СТРУКТУРА)
- ЦИКЛ С ПРЕДУСЛОВИЕМ
- ЦИКЛ С ПОСТУСЛОВИЕМ
- ВЕТВЛЕНИЕ
- СОКРАЩЕННОЕ ВЕТВЛЕНИЕ
- МНОЖЕСТВЕННЫЙ ВЫБОР



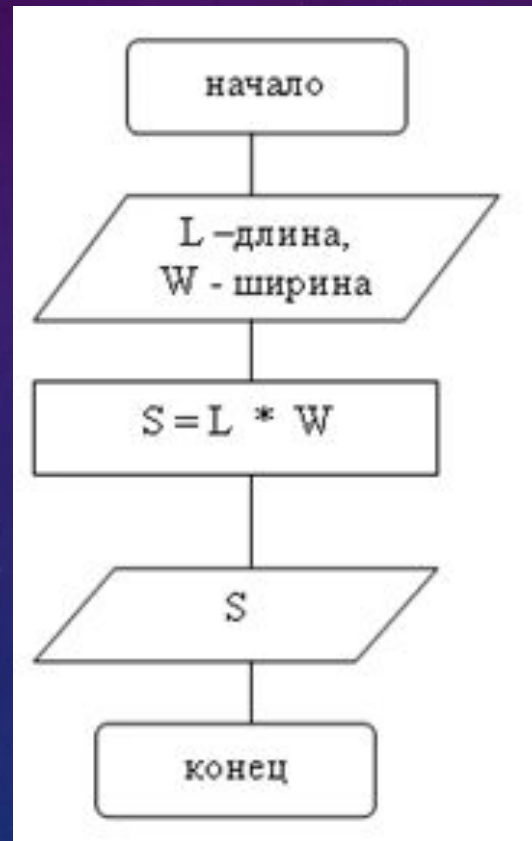
ЛИНЕЙНАЯ СТРУКТУРА ПРЕДСТАВЛЯЕТСЯ ПРОСТО ПОСЛЕДОВАТЕЛЬНОСТЬЮ ЗАПИСЬЮ ОПЕРАТОРОВ АЛГОРИТМЫ С  
ЛИНЕЙНОЙ СТРУКТУРОЙ ЯВЛЯЮТСЯ

ПРОСТЕЙШИМИ И ИСПОЛЬЗУЮТСЯ, КАК ПРАВИЛО, ДЛЯ РЕАЛИЗАЦИИ ПРОСТЫХ ВЫЧИСЛЕНИЙ.





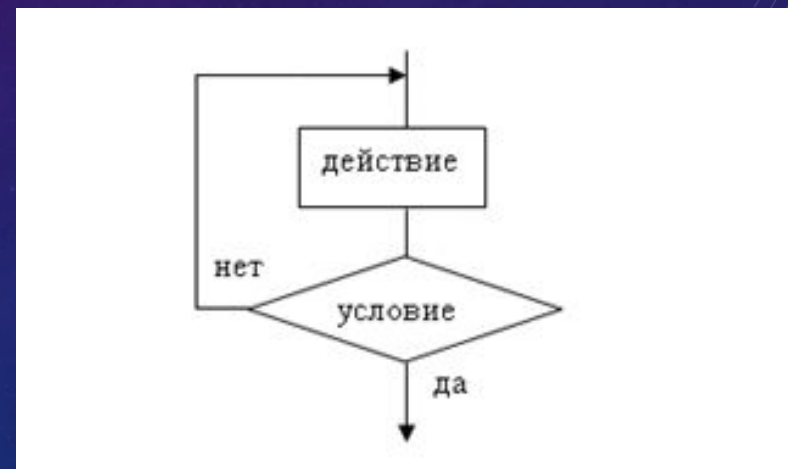
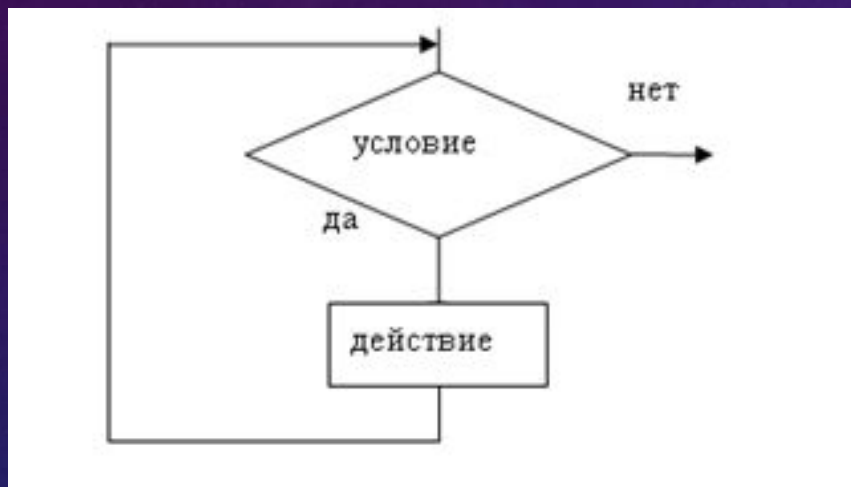
- В данном алгоритме выполняется последовательность действий: сначала вводятся исходные данные  $L$  и  $W$ , а затем выполняется вычисление площади параллелограмма  $S = L * W$  и последним действием является вывод выходных данных  $S$ . Графическое представление алгоритма изображается в виде композиции из геометрических фигур (основные блоки представлены на рис.2). Каждая геометрическая фигура соответствует тому или иному шагу алгоритмического процесса.



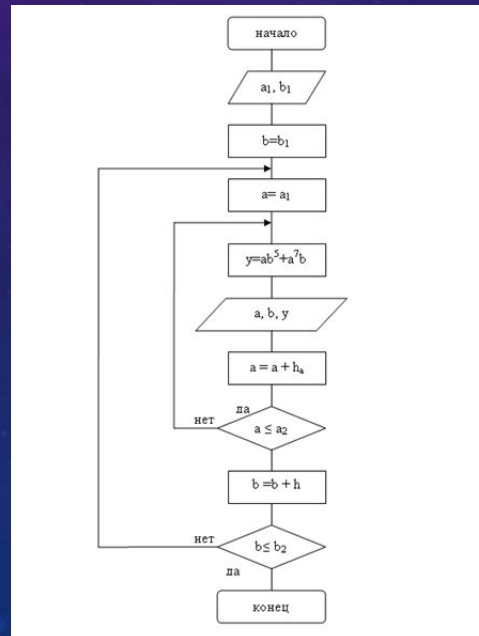
- **Структура алгоритма «цикл с предусловием»**

- В этой структуре «действие» будет повторяться до тех пор, пока будет выполняться условие (по ветке «+»). Поэтому в «действии» должно изменяться значение переменных, от которых зависит «условие», в противном случае произойдет заикливание (рис.5). Проверка «условия» производится до начала выполнения «действия» и может случиться так, что «действие» не будет выполнено ни разу.

- Рассмотрим структуру алгоритма «цикл с предусловием» на следующем примере.



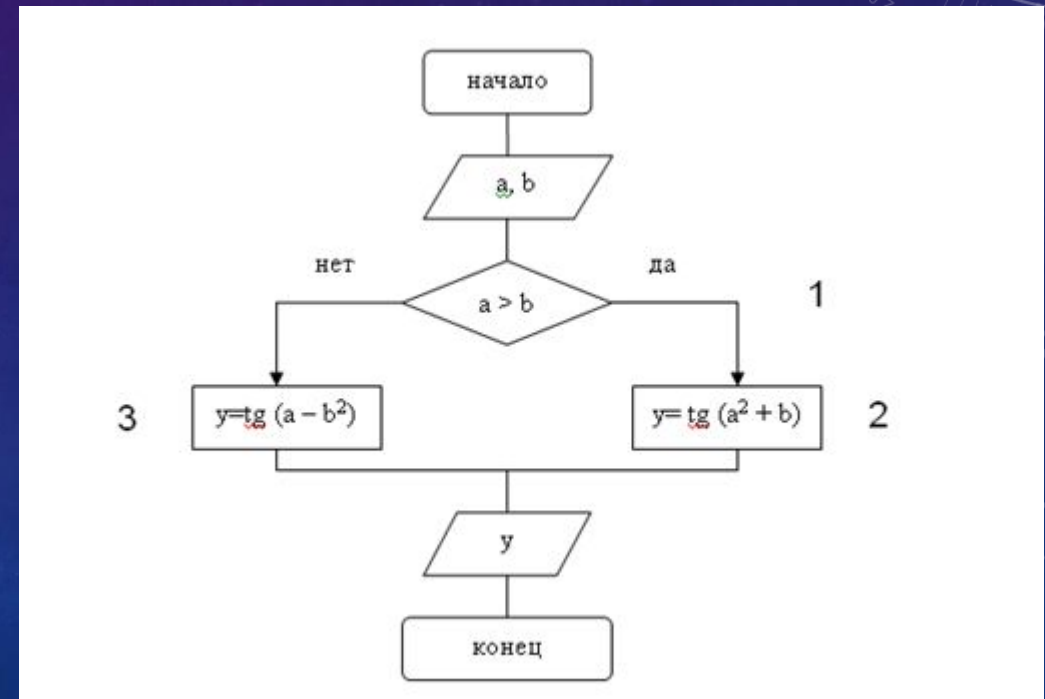
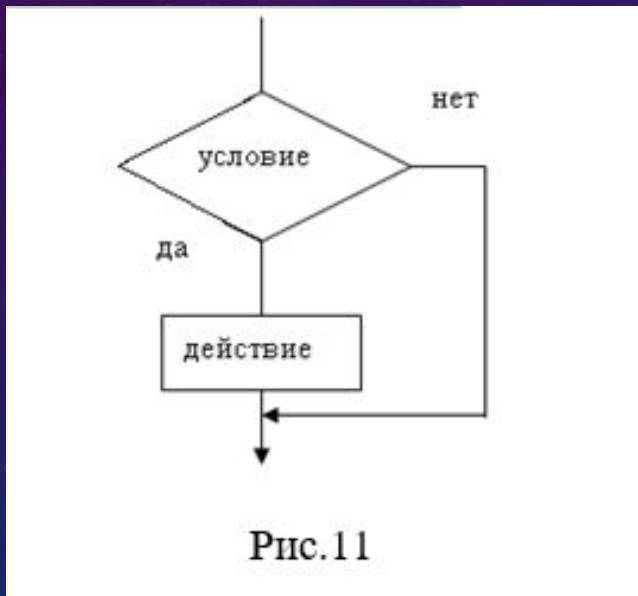
- Структура алгоритма «цикл с постусловием»
- Цикл с постусловием (рис.7) всегда выполняется хотя бы один раз, «действие» перестает выполняться, как только будет выполнено «условие» (по ветке «+»).
- Пояснения к блок-схеме (рис.8):
  - Аргументам  $a$  и  $b$  функции  $y$  присваиваются начальные значения диапазона изменения аргументов  $a_1$  и  $b_1$ ;
  - Вычисляется значение функции  $y$ ;
  - Вывод на экран аргументов  $a$ ,  $b$  и функции  $y$ ;
  - Аргументам функции  $a$  и  $b$  прибавляются их приращения  $h_a$  и  $h_b$ ;
  - Проверяются условия окончания циклов  $a \leq a_2$  и  $b \leq b_2$ . Действия выполняются до тех пор, пока не будут достигнуты конечные значения диапазонов изменений аргументов  $a$  и  $b$ .



- Структура алгоритма ветвление

- Ветвление используется для выбора одного из двух направлений дальнейшего хода решения задачи (рис.9).

- Сокращенное ветвление – это частный случай ветвления, когда отсутствует действие, выполняемое при не выполнении «условия» (по ветке « - »). Структура сокращенного ветвления представлена на рисунке 11.



- Структура алгоритма сокращенное ветвление

- Пояснения к блок-схеме (рис.12):

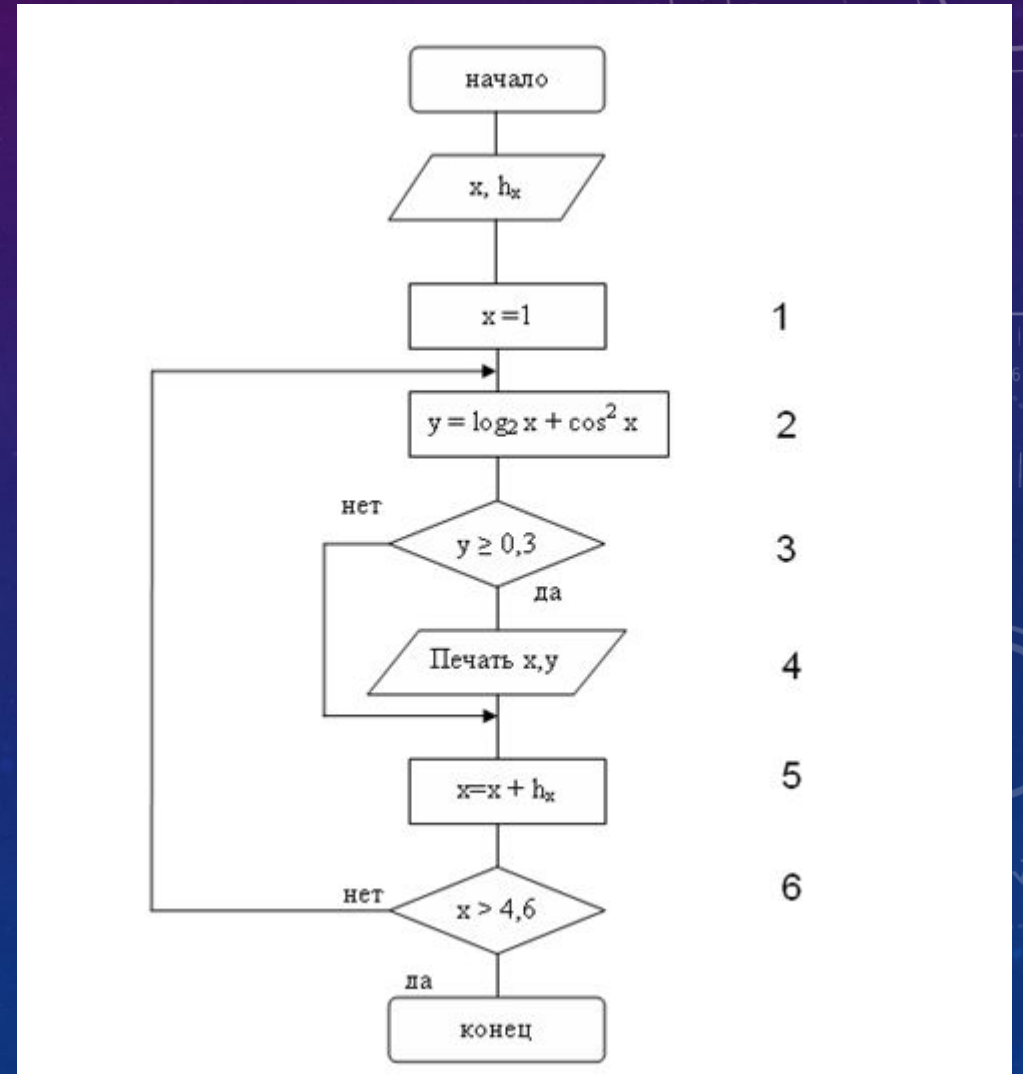
- Аргументу  $x$  функции  $y$  присваивается начальное значение диапазона изменения аргумента, а затем вычисляется значение функции  $y$ ;

- Проверяется условие значения функции ( $y \geq 0,3$ );

- Вывод на экран аргумента  $x$  и функции  $y$ ;

- Аргументу функции  $x$  прибавляется его приращение  $h_x$ ;

- Проверяется условие конца интервала аргумента ( $x > 4,6$ ).



- **Структура алгоритма множественный выбор**

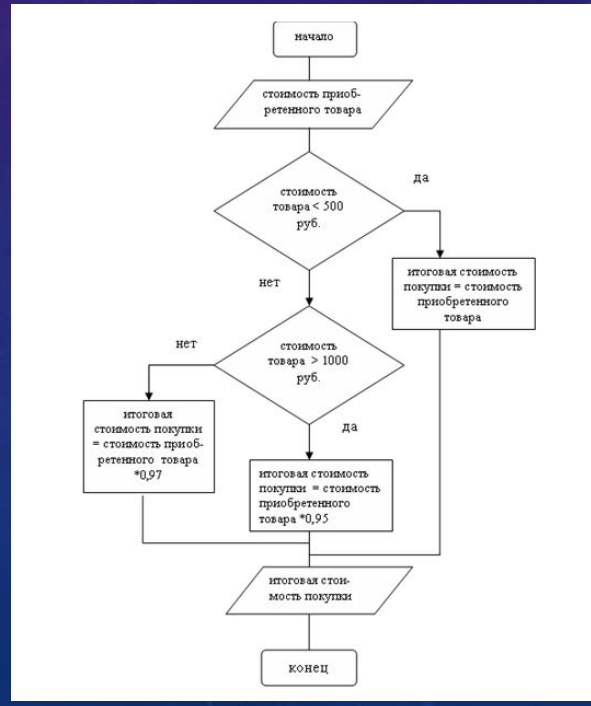
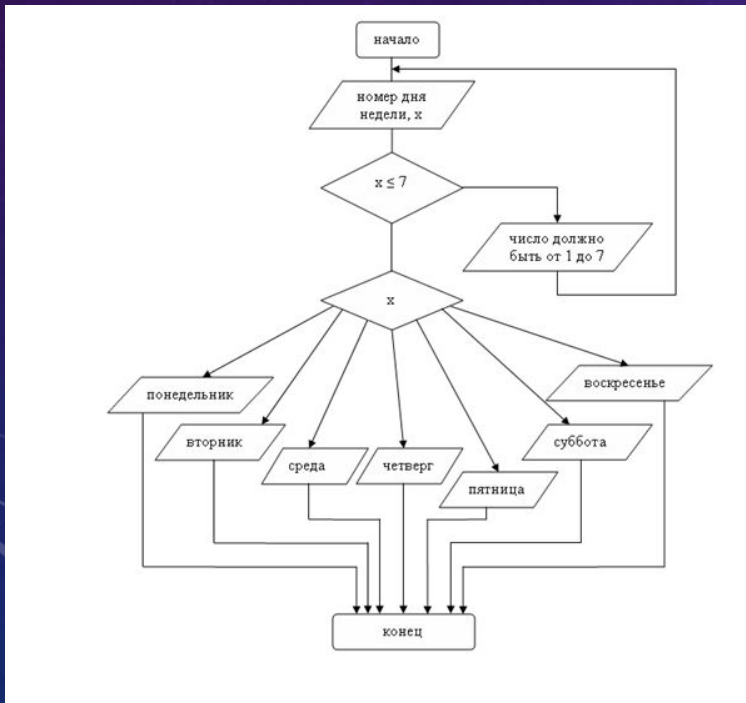
- Множественный выбор - обобщение структур ветвлений. Структура множественного выбора полезна в том случае, когда требуется выбрать одну из нескольких альтернатив. В зависимости от «условия» выполняется одно из действий 1, 2, ... .m. После чего происходит переход к выполнению следующей управляющей структуры (рис.13).

- Пояснения к блок-схеме (рис.14):

- Пользователем вводится номер дня недели.

- Проверяется условие номера дня недели ( $x \leq 7$ ).

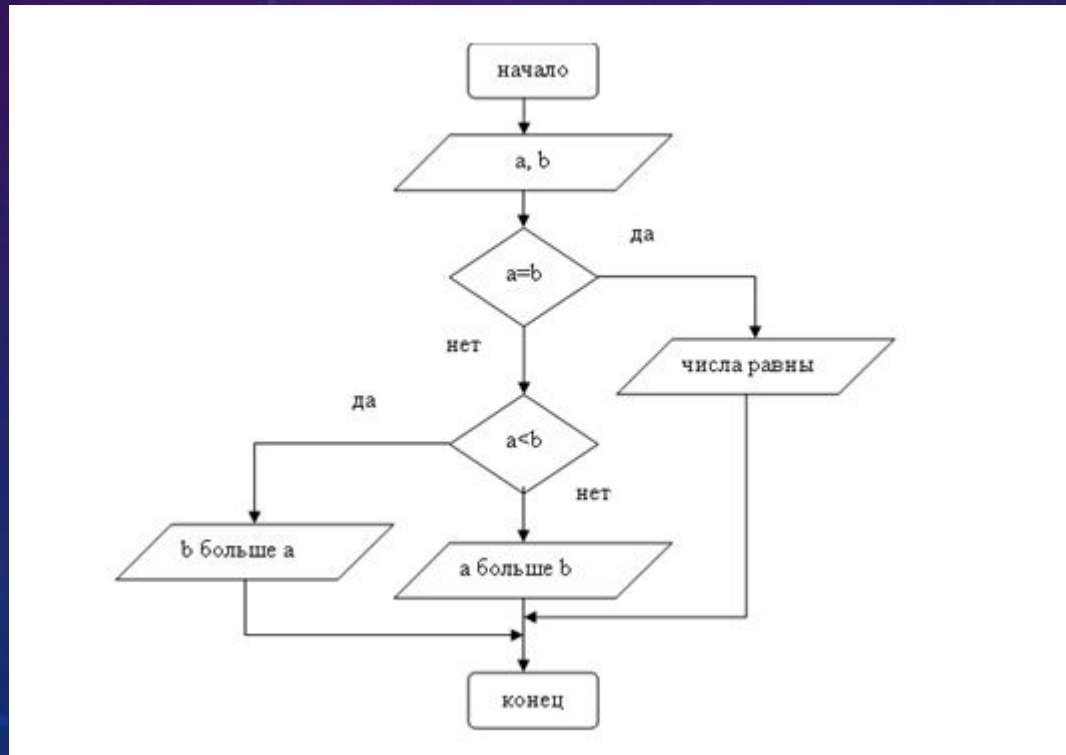
- В зависимости от номера дня недели выводится название дня недели, например, если  $x=3$ , то день недели – среда.



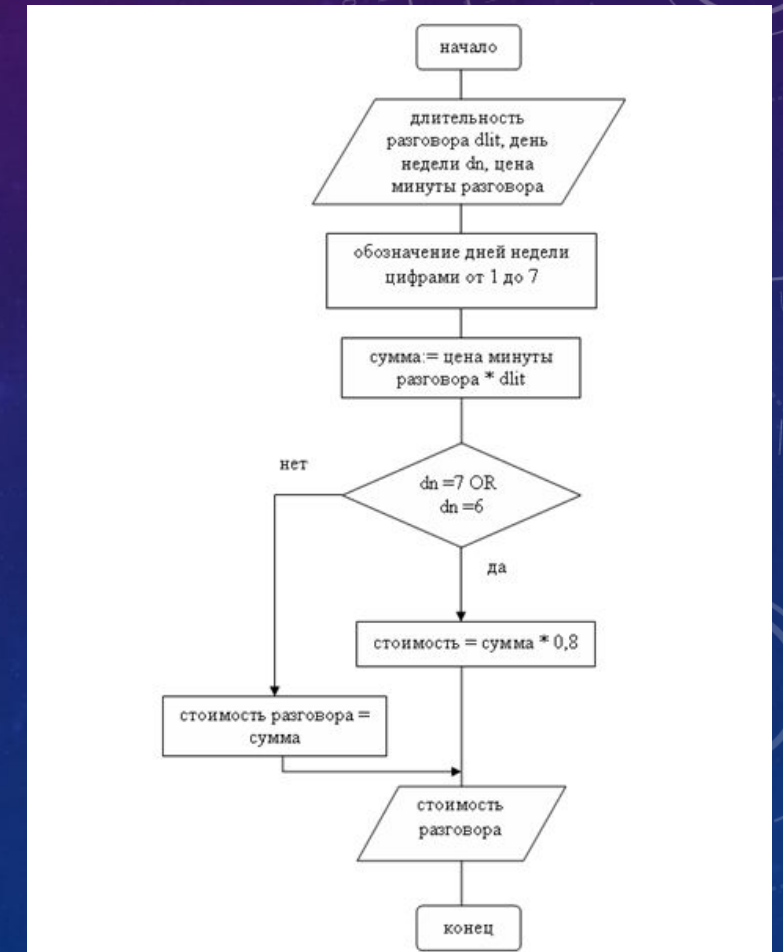
- **Смешанные структуры алгоритмов.**

- При решении задач определенного круга задач использование только одной стандартной структуры не достаточно, поэтому несколько стандартных структур в совокупности может быть использовано для составления алгоритмов в виде блок-схем. Следовательно, в результате получаются структуры алгоритмов, состоящие из различных стандартных структур алгоритмов. Пояснения к блок-схеме (рис.1):

- Проверяются условия стоимость приобретенного товара  $< 500$  рублей и  $> 1000$  рублей;
- Если условия выполняются, то вычисляются итоговые стоимости покупки с учетом 5% и 3% скидок;
- Вывод на экран итоговой стоимости покупки с учетом или без учета скидок

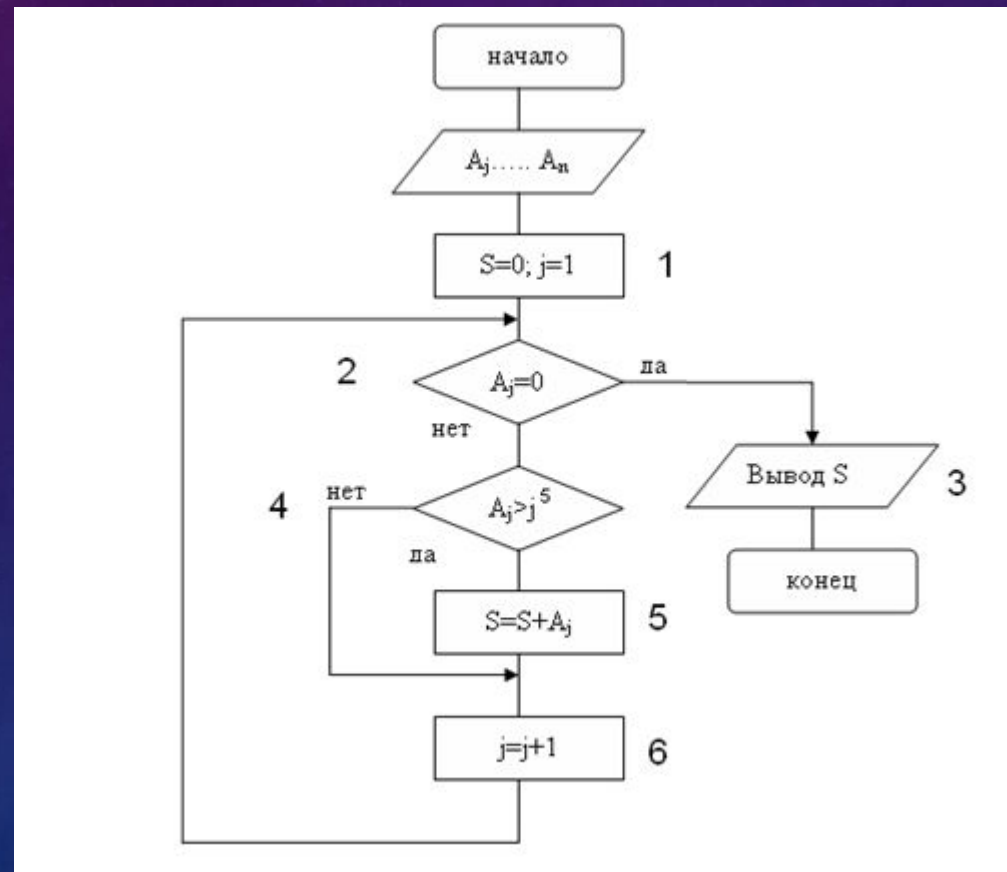


- Составить блок-схему алгоритма сравнения 2-х чисел, с указанием большего или меньшего числа в виде сообщения, например 56 больше 32.
- Последовательность действий при решении данной задачи описана блок-схемой алгоритма, представленной на рис.2.
- Пояснения к блок-схеме (рис.2):
- Ввод пользователем двух чисел а и b;
- Проверяются условия  $a = b$  и  $a < b$ ;
- В зависимости от того, какое условие выполняется, на экран выводятся сообщения.

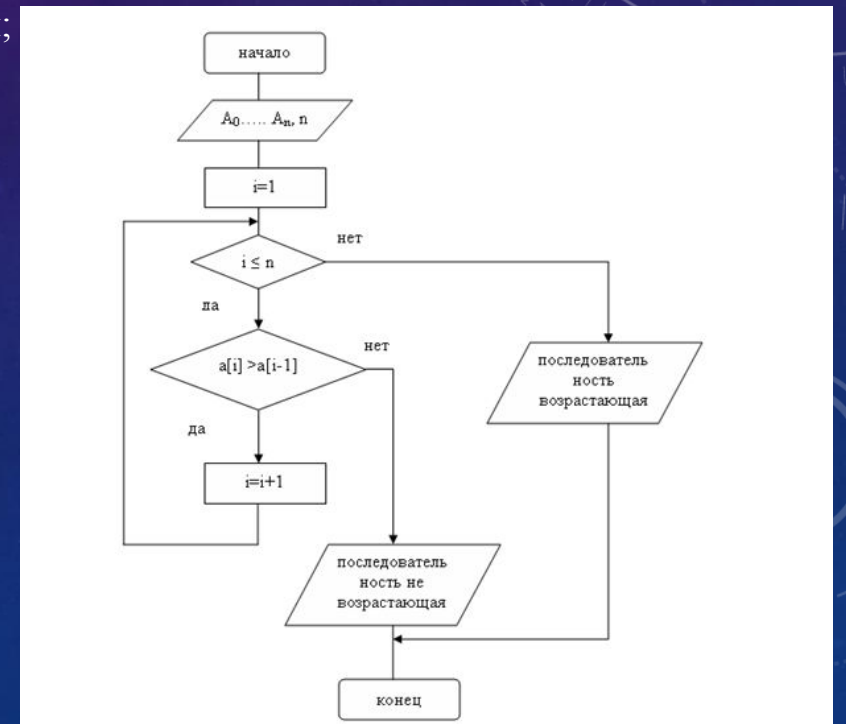




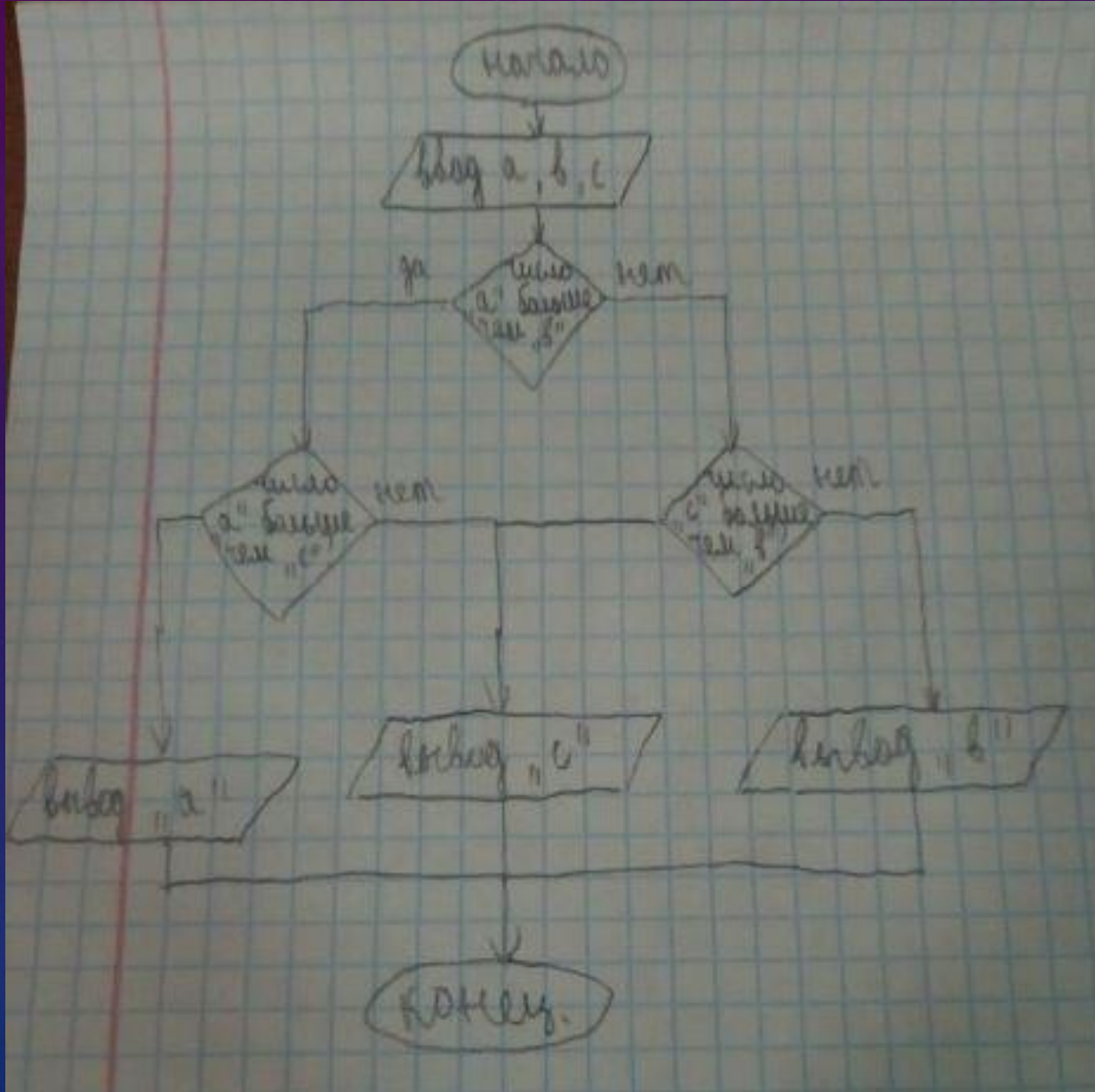
- Составить блок-схему алгоритма вычисления стоимости разговора по телефону с учетом 20% скидки, предоставляемой по субботам и воскресеньям (в данном примере цена минуты разговора фиксированная).
- Последовательность действий при решении данной задачи описана блок-схемой алгоритма, представленной на рис.3.
- Пояснения к блок-схеме (рис.3):
- Проверяется условие, является ли день недели субботой или воскресеньем ( $dn = 7$  OR  $dn = 6$ ).
- В зависимости от выполнения или не выполнения условия вывод на экран стоимости разговора по телефону с учетом и без учета скидки.



- Дана последовательность целых чисел  $A_1, A_2, \dots, A_n$ . Определить является ли данная последовательность возрастающей.
- Последовательность действий при решении данной задачи описана блок-схемой алгоритма, представленной на рис.6.
- Пояснения к блок-схеме (рис.6):
- Сравниваем текущий член последовательности с предыдущим.
- Вводится вспомогательная переменная  $S$ , которая будет суммировать значения  $\max$  и  $\min$  членов последовательности;
- Находится минимальное значение, путем перебора всех членов последовательности;
- Находится максимальное значение, путем перебора всех членов последовательности;
- При нахождении минимального и максимального значений, они складываются;
- Выводится на экран значение переменной  $S$ .

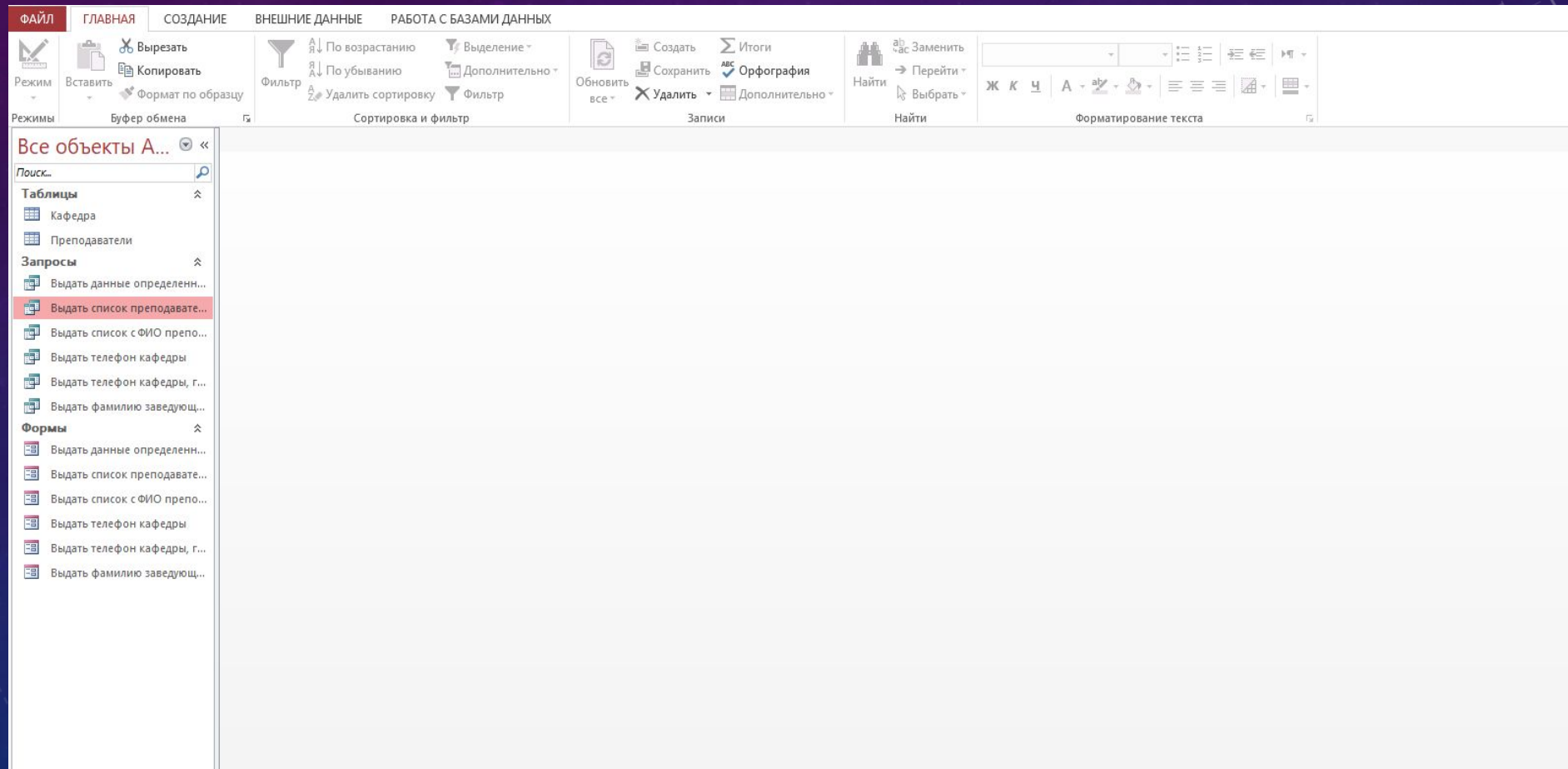


- В последнем задании нужно составить блок схему к задаче:
- Дана последовательность чисел, длина которой заранее неизвестна. Найти минимальное значение элементов последовательности.



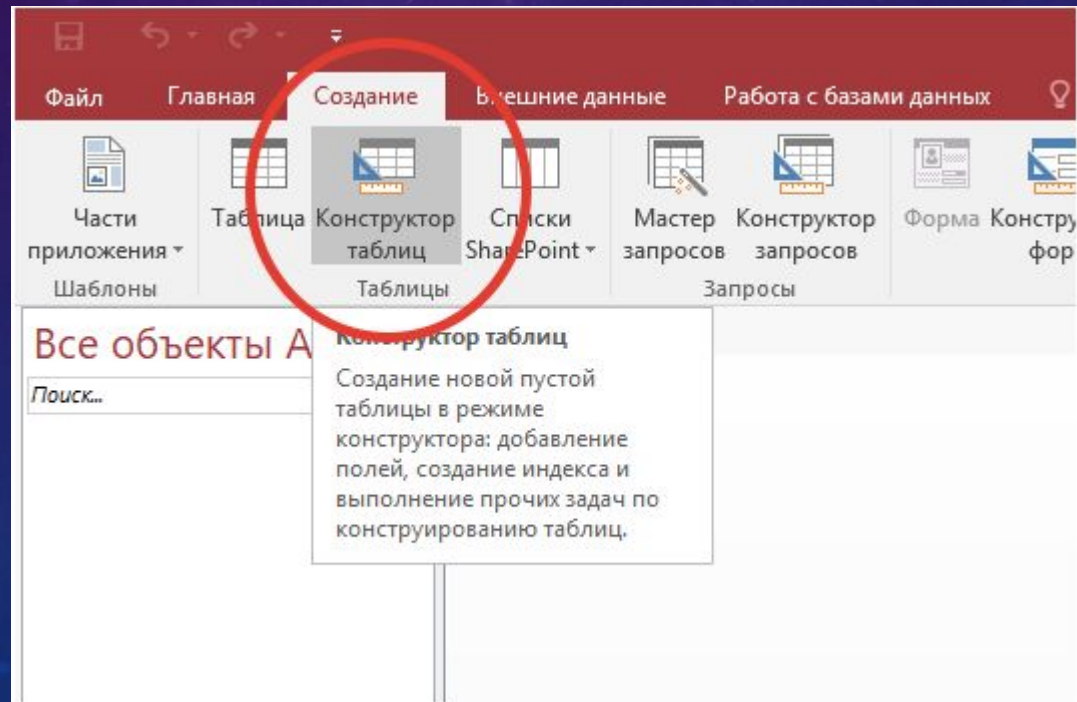
## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5 ТЕМА: РАБОТА С БАЗАМИ ДАННЫХ

- В ходе выполнения лабораторной работы №5 я научился создавать таблицы, работать с ними узнал что такое вкладки, научился изменять таблицы работая в конструкторе, научился работать в мастере конструктора и в мастере форм.

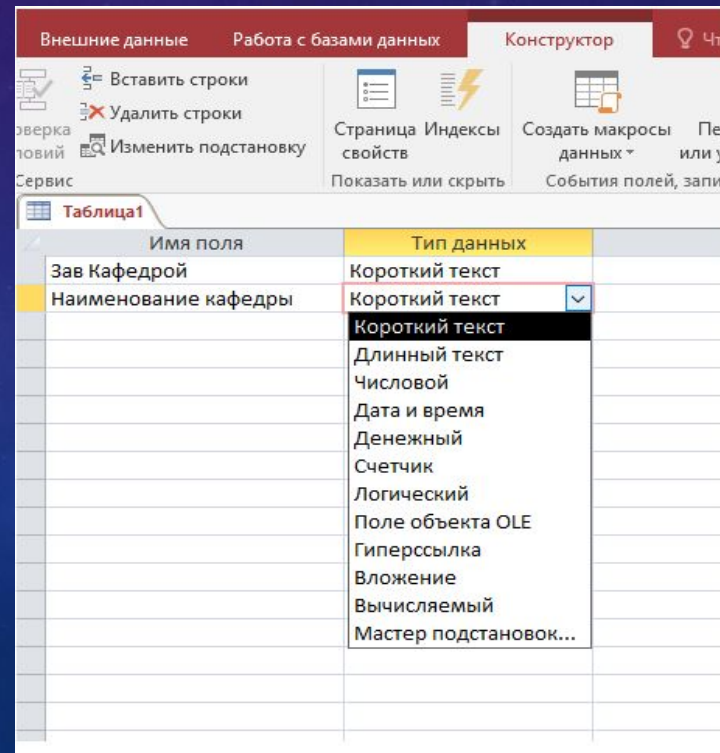


- Основным структурным компонентом базы данных является таблица. В таблицах хранятся вводимые нами данные. Внешне каждая таблица Access похожа на таблицы, с которыми мы привыкли работать на бумаге, — она состоит из столбцов, называемых *полями*, и строк, называемых *записями*. Каждая запись таблицы содержит всю необходимую информацию об отдельном элементе базы данных. Например, запись о преподавателе может содержать фамилию, имя, отчество, дату рождения, должность и т.п. При разработке структуры таблицы прежде всего необходимо определить названия полей, из которых она должна состоять, типы полей и их размеры. Каждому полю таблицы присваивается уникальное имя, которое не может содержать более 64 символов. Имя желательно делать таким, чтобы функция поля узнавалась по его имени. Далее надо решить, данные какого типа будут содержаться в каждом поле. В Access можно выбирать любые из основных типов данных. Один из этих типов данных должен быть присвоен каждому полю. Значение типа поля может быть задано только в режиме конструктора.
- В Access существует четыре способа создания пустой таблицы:
- использование мастера баз данных для создания всей базы данных, содержащей все требуемые отчеты, таблицы и формы, за одну операцию. Мастер баз данных создает новую базу данных, его нельзя использовать для добавления новых таблиц, форм, отчетов в уже существующую базу данных;
- мастер таблиц позволяет выбрать поля для данной таблицы из множества определенных ранее таблиц, таких, как деловые контакты, список личного имущества или рецепты;
- ввод данных непосредственно в пустую таблицу в режиме таблицы. При сохранении новой таблицы в Access данные анализируются и каждому полю присваивается необходимый тип данных и формат;
- определение всех параметров макета таблицы в режиме конструктора.
- Независимо от метода, примененного для создания таблицы, всегда имеется возможность использовать режим конструктора для дальнейшего изменения макета таблицы, например для добавления новых полей, установки значений по умолчанию или для создания масок ввода. Однако только четвертый метод позволяет сразу задать ту структуру таблицы, которая вам нужна, и поэтому далее рассмотрим именно этот метод.

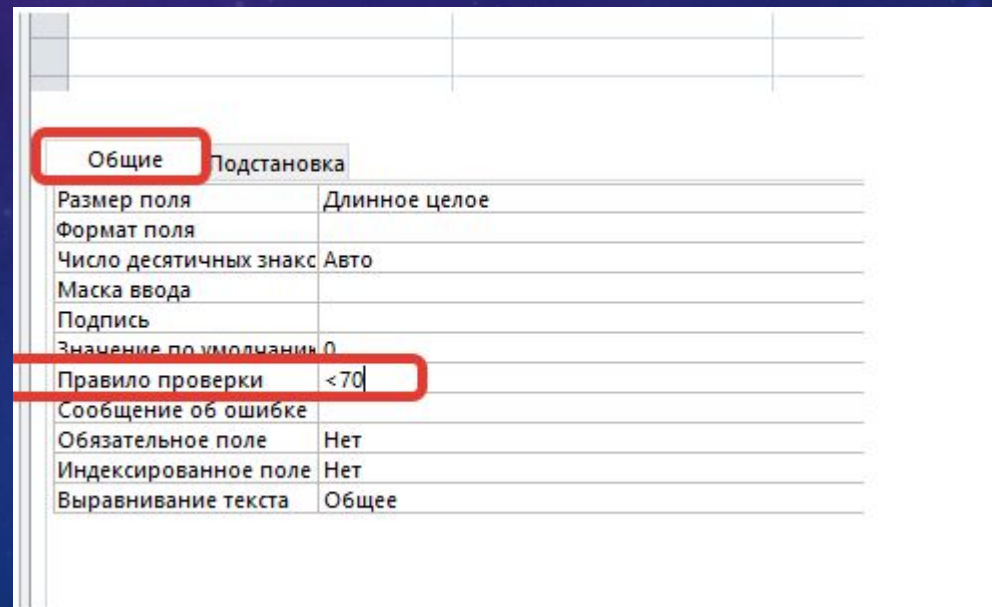
- Особенности ввода следующие: при нажатии клавиши <Del> ячейка очищается; если ввод данных в ячейку прервать, нажав клавишу <Esc>, то восстанавливается старое значение, а если нажать клавиши <Enter> или <Tab>, то в ячейку заносится новое значение. Можно редактировать текущее значение, меняя в нем отдельные символы. Для этого надо в ячейке таблицы дважды щелкнуть мышью или нажать клавишу <F2>. Однако есть одна важная особенность при вводе данных — для некоторых типов данных (числовой, денежный, дата/время, логический) Access автоматически проверяет правильность их ввода. Например, если вы введете букву в ячейку с числовым типом, то Access выдаст сообщение о неправильно введенном значении и не позволит перейти к другой ячейке, пока не будут введены правильные данные.
- Для всех типов полей (кроме типов *Счетчик* и поле объекта OLE) можно самостоятельно задавать ограничения для вводимых данных. Для этого



- В режиме конструктора надо выбрать вкладку *Общие*, перевести курсор в поле с именем «Правило проверки» ввести ограничение на данные. Например, стаж работы вряд ли может быть более 70 лет.: ограничение будет следующее: <70. Ограничение можно вводить и на текстовое поле Обычно в этом случае задаются слова, которые могут присутствовать в данном поле *Примечание*. Вводить ограничения можно не только вручную, но и с помощью *Построителя выражений*. При появлении курсора в строке параметр! *Условия на значение* справа от нее появится -кнопка с тремя точками. При щелчке по этой кнопке появляется окно построителя выражений. Обычно им пользуются для построения сложных выражений, но разобраться 1 его работе довольно просто при помощи справки, расположенной в этом же окне.



- Можно использовать еще один удобный инструмент при вводе данных — параметр *Значение по умолчанию* (находится также на вкладке *Общие*). Здесь можно задать данные, которые Access будет вводить по умолчанию при заполнении таблицы. Это удобно использовать, когда большинство значений данного поля одинаковы и лишь некоторые отличаются. Например, должность большинства преподавателей — доцент. Если ввести это слово в строку параметра *Значение по умолчанию*, то все ячейки поля «Должность» примут значение *Доцент* и надо будет лишь изменить значения ячеек для преподавателей с другими должностями. Данные можно вводить и копированием их из .одной ячейки в другую стандартными средствами Windows.
- Для каждого типа данных (кроме типа данных *Счетчик*) предусмотрено





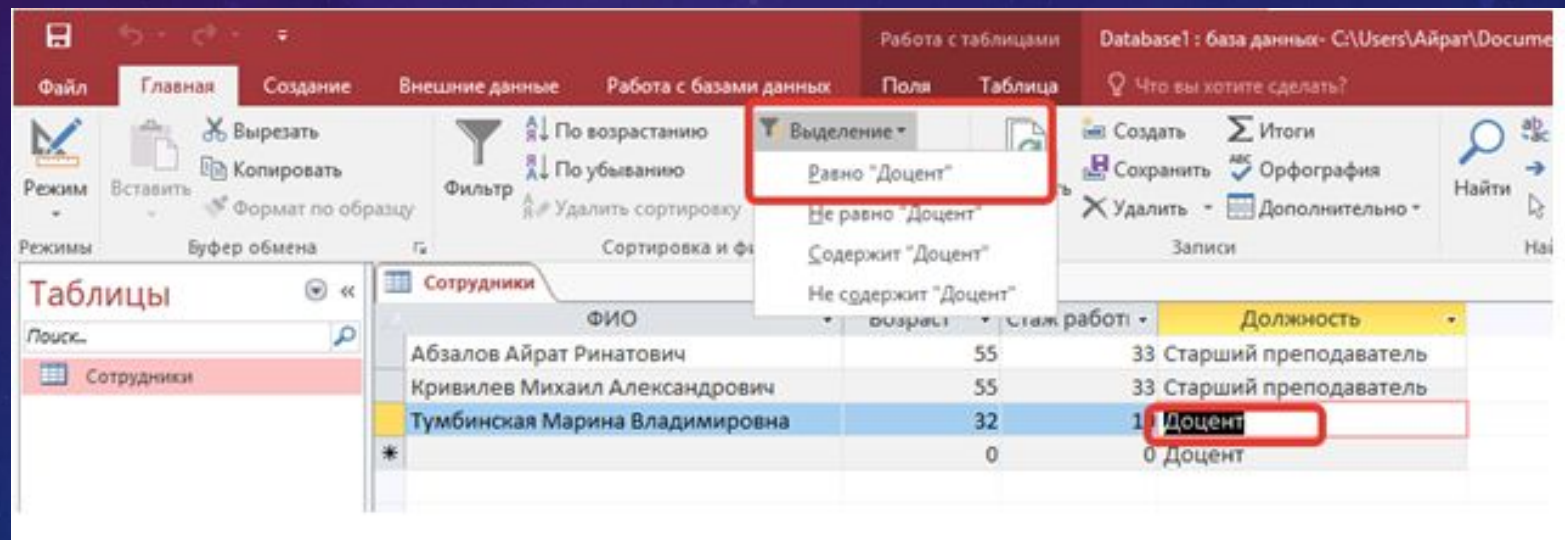
- Пустое (нулевое) значение. Различают два типа пустых значений: пустые (Null) значения и пустые строки. Если пустое поле имеет пустое (Null) значение, то это означает, что данные для него существуют, но пока не известны. Если же введена пустая строка (два знака прямых кавычек ("")), то это означает, что данные не существуют вовсе. Access позволяет обрабатывать такие пустые значения.
- **Редактирование данных**
- Редактирование данных производится очень просто — курсор переводится в нужную ячейку, старые данные удаляются (клавишами <Del> или <Backspace>) и вводятся новые данные. Удалять можно не только данные в ячейках, но и целиком строки, предварительно их выделив. Но если таблица большая, то редактируемые данные надо сначала найти. Это делается следующим образом: переводят курсор в любую ячейку поля, начиная с которой будет вестись поиск, затем открывают пункт меню **Правка** и выполняют команду **Найти**. В появившемся окне вводят образец искомых данных и щелкают по кнопке <Найти>. Если значение найдено, курсор перейдет в эту ячейку. Иногда требуется большое количество одинаковых данных заменить на другое значение. Для этого надо открыть пункт меню **Правка** и выполнить команду **Заменить**. В появившемся окне ввести образцы того, что надо найти и на что заменить. Затем надо решить, будете ли вы просматривать каждое заменяемое значение или нет. Если будете, то надо щелкнуть по кнопке <Заменить>, а если нет — один раз щелкнуть по кнопке <Заменить все>.

- Для удобства просмотра можно сортировать записи в таблице в определенной последовательности, например, в таблице *Преподаватели* записи можно отсортировать в порядке убывания стажа преподавателей. Кнопки сортировки на панели инструментов (или команды меню **Записи** ▢ **Сортировка** ▢ **Сортировка по возрастанию (по убыванию)**) позволяют сортировать столбцы по возрастанию или по убыванию. Прежде чем щелкнуть по кнопке сортировки, следует выбрать поля, используемые для сортировки. Для выбора поля достаточно поместить курсор в любую его запись. После этого щелкните по кнопке сортировки — и данные отобразятся в отсортированном порядке. В режиме таблицы можно выделить сразу два или несколько соседних столбцов, а затем выполнить по ним сортировку. По умолчанию в Access 97 сортировка записей начинается с крайнего, левого выделенного столбца. При этом записи таблицы будут отсортированы сначала по крайнему левому выделенному столбцу, затем (для одинаковых значений в первом сортируемом столбце) — по второму и т.д. Если вы хотите восстановить порядок отображения записей используйте команду **Записи** ▢ **Удалить фильтр**.
- Можно использовать еще один удобный инструмент при вводе данных — параметр *Значение по умолчанию* (находится также на вкладке *Общие*). Здесь можно задать данные, которые Access будет вводить по умолчанию при заполнении таблицы. Это удобно использовать, когда большинство значений данного поля одинаковы и лишь некоторые отличаются. Например, должность большинства преподавателей — доцент. Если ввести это слово в строку параметра *Значение по умолчанию*, то все ячейки поля «Должность» примут значение *Доцент* и надо будет лишь изменить значения ячеек для преподавателей с другими должностями. Данные можно вводить и копированием их из .одной ячейки в другую стандартными средствами Windows.

- **Отбор данных с помощью фильтра**

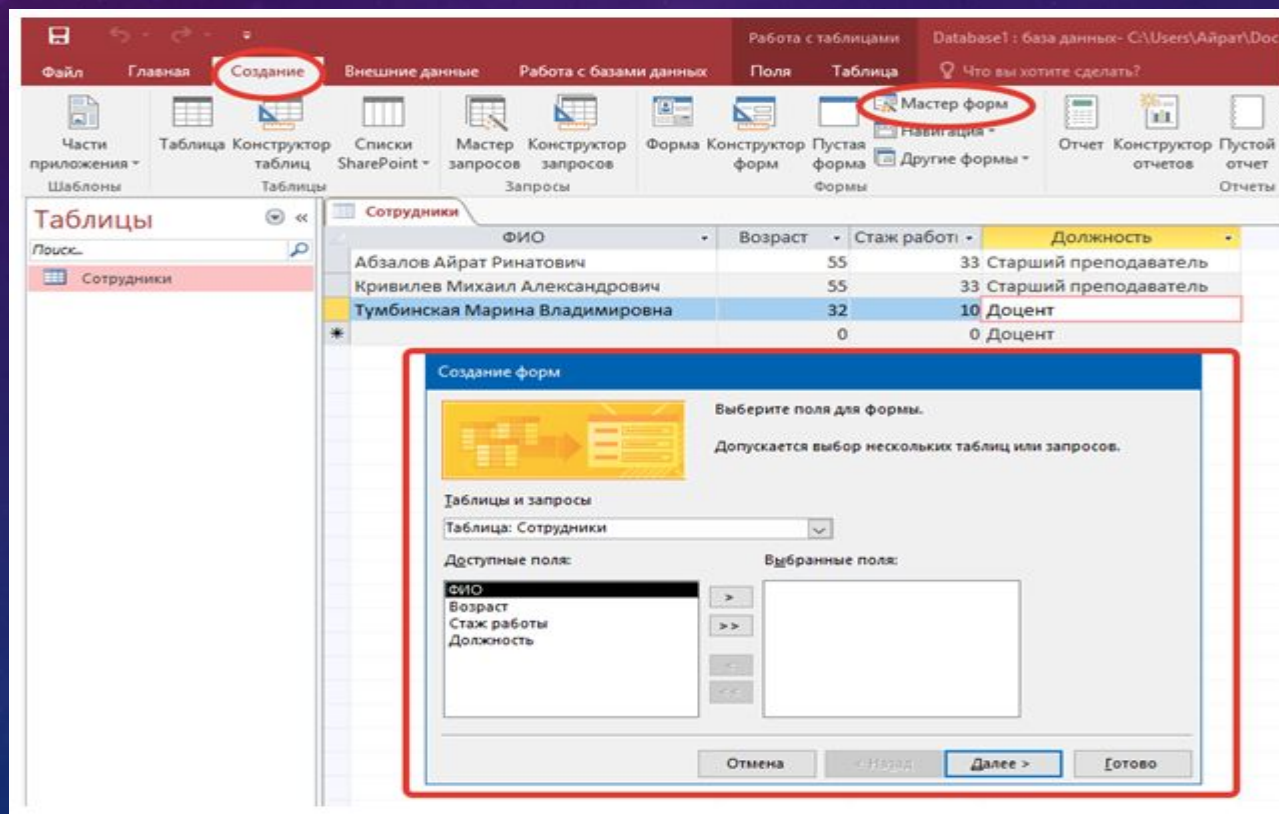
- Фильтр — это набор условий, применяемых для отбора подмножества записей. В Access 97 существуют фильтры четырех типов: фильтр по выделенному фрагменту, обычный фильтр, расширенный фильтр и фильтр по вводу.

- *Фильтр по выделенному фрагменту* — это способ быстрого отбора записей по выделенному образцу. Например, вам нужно просмотреть в таблице записи только о доцентах. Выделите слово *Доцент* в любой из записей. Щелкните по кнопке <Фильтр по выделенному> (или выполните команду **Записи** □ **Фильтр** □ **Фильтр по выделенному**), и Access выберет только те записи, для которых значение в столбце *Должность* равно *Доцент*.

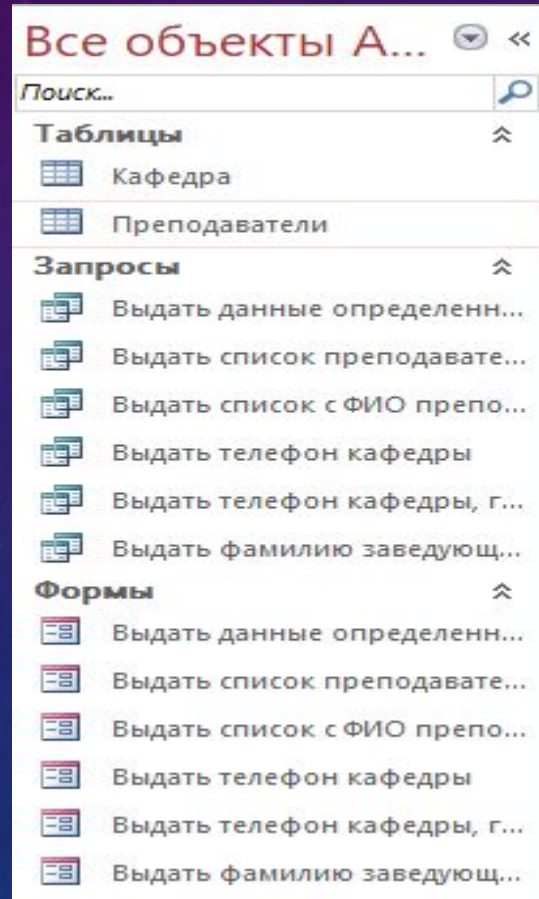


Собратите внимание, по строке состояния окна таблицы присутствует слово *Фильтр*. В дополнение к этому кнопка <Применить фильтр> (третья кнопка фильтрования, на которой изображена воронка) затенена, а это означает, что используется фильтр. При отключении этой кнопка все фильтры будут сняты. Установки фильтра не пропадут: он просто будет отключен. Фильтр по выделенному может собирать вместе критерии выбора при каждом использовании кнопки <Фильтр по выделенному>. Например, вы поместили курсор в столбец Должность в записи, -в которой значение поля равно *Доцент*, и щелкнули по кнопке <Фильтр по выделенному>. Вы увидите только записи о всех доцентах. Если затем вы поместите курсор в столбец <Должность> и выделите слово *Информатика* (конечно, если такая дисциплина присутствует в вашей таблице), а затем щелкните по кнопке <Фильтр по выделенному>, появятся только записи о доцентах, которые преподают информатику. Если вы хотите просмотреть значения, которые не удовлетворяют этому критерию, например просмотреть всех преподавателей, кроме доцентов, читающих информатику, щелкните правой кнопкой (курсor мыши должен находиться внутри таблицы) и выберите команду **Исключить выделенное**. Будут выбраны все записи, кроме недавно выбранных (*инверсный выбор*). Фильтрование данных в Access производится с помощью кнопок <Фильтр по выделенному> или <Изменить фильтр> (команда меню **Записи** □ **Фильтр** □ **Изменить фильтр**). После нажатия второй кнопки от таблицы остается одна запись. Каждое поле становится полем со списком (когда в нем находится курсор), в котором можно выбрать из списка все значения для данного поля. После щелчка по кнопке <Применить фильтр> будут выбраны записи, соответствующие измененному фильтру. Еще более сложные условия фильтрации можно задать командой меню **Записи** □ **Фильтр** □ **Расширенный фильтр...**

- **Ввод и просмотр данных посредством формы**
- Формы обеспечивают наиболее гибкий способ ввода, редактирования, просмотра и удаления данных и фактически являются шаблонами, управляющими отображением информации. Форма позволяет отображать одновременно все поля одной или нескольких записей. Оптимально построенная форма может вмещать несколько десятков полей на одном экране, а если полей намного больше, то для каждой записи можно создать многостраничную форму. Можно создать форму-меню для вызова других форм, таблиц, запросов или отчетов. В форме каждое поле можно разместить в точно заданном месте, выбрать для него цвет или заливку и добавить элементы управления текстом для эффективного ввода данных.



- В моей базе данных имеется две таблицы, шесть запросов и шесть форм



Кафедра		Преподаватели		
Код	Зав кафедр	Наименова	Телефон ка	Щелкните для добавления
	1 Шарнин Л.М.	АСУ	645432	
	2 Новиков С.В.	ИППК	434322	
	3 Даутов О.Ш.	КПВА	897543	
	4 Песошин В.А.	КС	446677	
	5 Солодухо Н.М.	Философии	678594	
	6 Глова В.И.	СИБ	230056	
*	(№)		0	

- У меня есть две таблицы «кафедра» и «преподаватели» в кафедре у меня указано шесть человек, а в таблице преподаватели у меня указаны 10 преподавателей

Кафедра		Преподаватели				
Код	Наименов	ФИО	Адрес	Дата рожде	Табельный	Щелкните для добавления
	1 ИППК	Бердникова Э.Р.	Попова 23-56	03.09.1975	12345	
	2 АСУ	Медведев В.И.	Декабристов 60-54	16.10.1978	13323	
	3 СИБ	Аникин И.В.	Мусина 6-59	16.03.1935	21344	
	4 АСУ	Яхина З.Т.	Ак.Павлова 45-66	12.04.1940	23545	
	5 Философии	Евстратов А.Р.	Амирхана 40-66	24.07.1957	42544	
	6 Философии	Солодухо Н.М.	Декабристов 55-33	04.07.1970	54656	
	7 КС	Хафизхова А.Ш.	Ак.Губкина 12-111	09.07.1970	55556	
	8 КПВА	Воронова В.В.	Гладилова 12-5	08.12.1970	54544	
	9 СИБ	Александрова Л.	Космонавтов 70-11	03.11.1963	78394	
	10 СИБ	Петровский В.И.	Челюскина 85-62	14.05.1950	45214	
*	(№)				0	

- В базе данных у меня есть шесть запросов «выдать данные определенного препода» , «выдать список преподав», «Выдать список с ФИО преподав определенной кафедры», «Выдать телефон кафедры», «Выдать телефон кафедры, где работает определенный препода», «Выдать фамилию заведующего кафедрой СИБ

Преподаватели	Кафедра	Выдать список преподавателей
ФИО	Дата рожде	
Аникин И.В.	16.03.1935	
Яхина З.Т.	12.04.1940	
Евстратов А.Р.	24.07.1957	
Петровский В.И.	14.05.1950	
*		

Преподаватели	Кафедра	Выдать телефон кафедры
Телефон ка	Наименова	
446677	КС	
*	0	

Выдать фамилию заведующего кафедрой СИБ			
Зав кафедры	Наименова	Телефон ка	Код
Глова В.И.	СИБ	230056	6
*		0	(№)

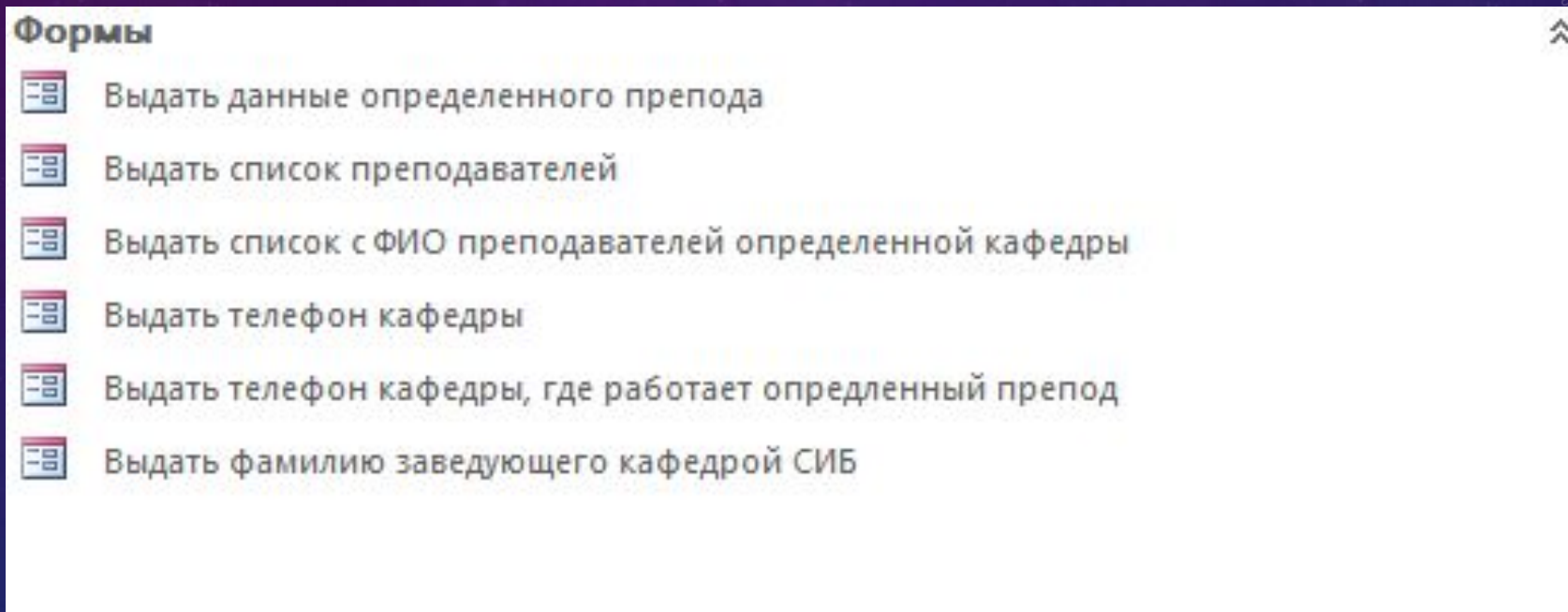
Код	Зав кафедры	Наименова	Телефон ка	Щелкните для добавления
1	Шарнин Л.М.	АСУ	645432	
2	Новиков С.В	ИППК	434322	
3	Даутов О.Ш.	КПВА	897543	
4	Песошин В.А	КС	446677	
5	Солодухо Н.М	Философии	678594	
6	Глова В.И.	СИБ	230056	
*	(№)		0	



- **Запросы**

- Запросы являются мощным средством обработки данных, хранимых в таблицах Access 97. С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц. Они также используются в качестве источника данных для форм и отчетов. Запросы позволяют вычислять итоговые значения и выводить их в компактном формате, подобном формату электронной таблицы, а также выполнять вычисления над группами записей.
- Запросы можно создавать самостоятельно и с помощью *мастеров*. *Мастера запросов* автоматически выполняют основные действия в зависимости от ответов пользователя на поставленные вопросы. Самостоятельно разработать запросы можно в режиме конструктора.
- В Access можно создавать следующие типы запросов:
  - запрос на выборку;
  - запрос с параметрами;
  - перекрестный запрос;
  - запрос на изменение (запрос на удаление, обновление и добавление записей на создание таблицы);
  - запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы)
- *Запрос на выборку* используется наиболее часто. При его выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или из нескольких таблиц и выводятся в определенном порядке. Например, можно вывести на экран данные о фамилиях доцентов, стаж которых более 15 лет (на основе таблицы *Преподаватели*).

У меня в базе данных имеется шесть форм: Выдать данные определенного препода, выдать список преподаов, выдать список с ФИО преподаов определенной кафедры, выдать телефон кафедры, выдать телефон кафедры где работает определенный препод, выдать фамилию заведующего кафедрой СИБ



Все мои формы, сделанные в мастере конструкторов.

Выдать данные определенного преподавателя | Выдать список преподавателей | Выдать список с ФИО преподавателей определенной кафедры | **Выдать телефон кафедры**

Телефон кафедры:

Наименование кафедры:

**Выдать фамилию заведующего кафедрой СИБ**

Зав кафедры:

Наименование кафедры:

Телефон кафедры:

Код:

**Выдать список с ФИО преподавателей определенной кафедры**

Наименование кафедры:

Зав кафедры:

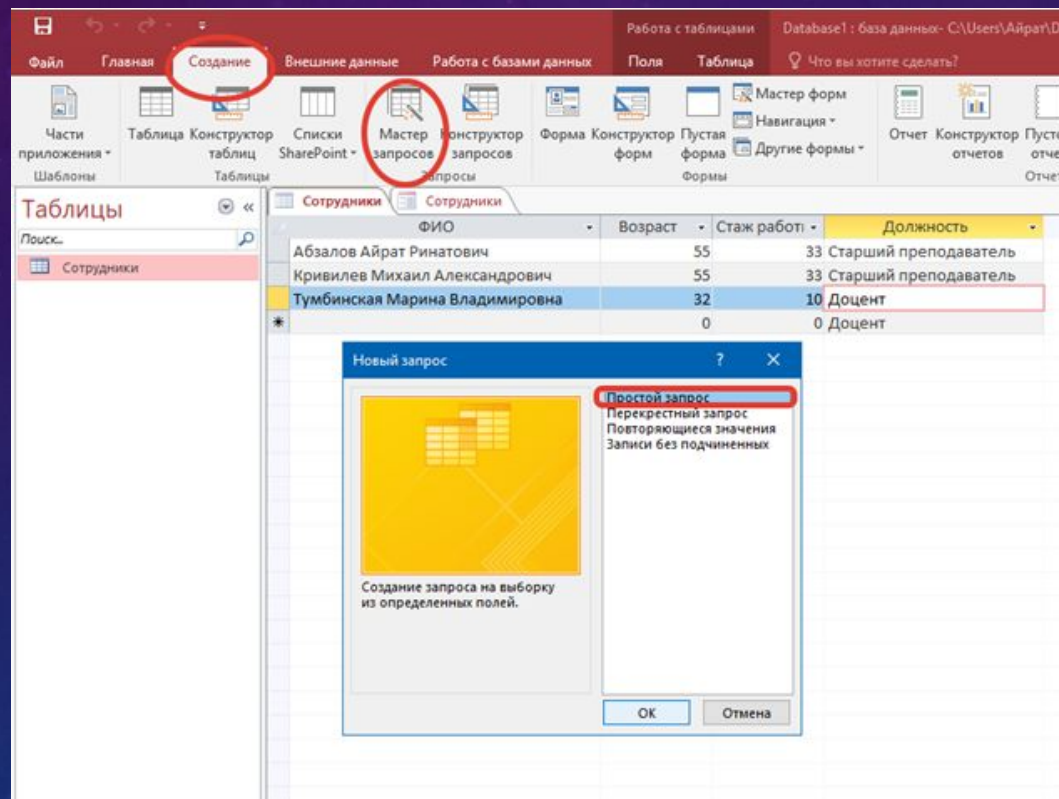
**Выдать список преподавателей**

ФИО:

Дата рождения:

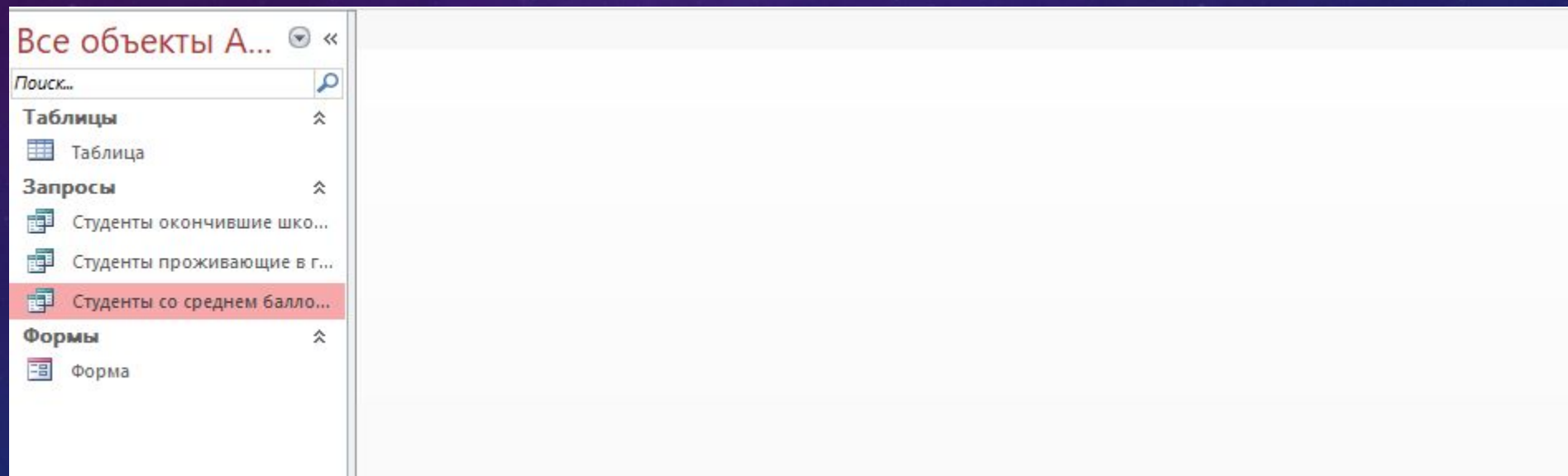
- *Запрос с параметрами* — это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, на основе которых будет выполняться запрос. Например, часто требуются данные о том, какие дисциплины ведут преподаватели. Чтобы не создавать отдельные запросы по каждому преподавателю, можно создать один запрос с параметрами, где в качестве параметра будет использоваться фамилия преподавателя. При каждом вызове этого запроса вам будет предложено ввести фамилию преподавателя, а затем на экран будут выведены все поля, которые вы указали в запросе, например фамилия, имя, отчество преподавателя и читаемая им дисциплина.
- В *перекрестном запросе* отображаются результаты статистических расчетов (такие, как суммы, количество записей, средние значения), выполненных по данным из одного поля таблицы. Эти результаты группируются по двум наборам данных, один из которых расположен в левом столбце таблицы, а второй — в верхней строке. Например, нам надо узнать средний стаж работы ассистентов, доцентов и профессоров на разных кафедрах (на основе таблицы *Преподаватели*). Перекрестный запрос позволит легко решить эту задачу, создан таблицу, в которой заголовками строк будут служить должности, заголовками столбцов - названия кафедр, а в ячейках будут рассчитаны средние значения стажа преподавателей.

- *Запрос на изменение* — это запрос, который за одну операцию вносит изменения в не-сколько записей. Существует четыре типа запросов на изменение: на удаление, обновление и добавление записей, а также на создание таблицы.
- *Запрос на удаление* удаляет группу записей, удовлетворяющих заданным условиям, одной или нескольких таблиц. С помощью запроса на удаление можно удалять только вся запись, а не отдельные поля внутри нее.
- *Запрос на обновление записей* вносит общие изменения в группу



- *Запрос на добавление* добавляет группу записей из одной или нескольких таблиц в по-нец одной или нескольких таблиц. Например, появилось несколько новых преподавателей а также база данных, содержащая сведения о них. Чтобы не вводить все данные вручную их можно добавить в таблицу *Преподаватели*.
- *Запрос на создание таблицы* создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Например, на основе таблицы *Преподаватели* можно создать новую таблицу, содержащую данные только о профессорах.
- Для создания нового запроса надо в окне базы данных выбрать вкладку *Запросы* и щелкнуть по кнопке <Создать>. Откроется окно «Новый запрос».
- В окне вы должны выбрать один из пяти пунктов: Конструктор, Простой запрос. Перекрестный запрос, Повторяющиеся записи, Записи без подчиненных. *Конструктор* позволит вам самостоятельно создать любой тип запроса, но этот режим рекомендуется пользователям, уже имеющим некоторый опыт создания запросов. *Простой запрос* позволит создать с помощью *Мастера* запрос на выборку из определенных полей таблиц или других запросов. (Это наилучший способ создания запроса для начинающих пользователей.)
- Что такое *перекрестный запрос*, рассматривалось выше. При выборе пункта *Повторяющиеся записи* будет создан запрос на поиск повторяющихся записей в простой таблице или в запросе, а при выборе пункта *Записи без подчиненных* — запрос на поиск записей, которым не соответствует ни одна запись в подчиненной таблице. Такой запрос используется для многотабличных баз данных.

- База данных «Абитуриет» в нее входят одна таблица, три запроса, и одна форма







студенты со средним баллом аттестата «4,5» и выше		студенты, окончившие школу до 2020 года		
Средний бал аттестата	Фамилия	Имя	Отчество	
4.7	Саитов	Ренат	Альфетович	
4.85	Федоров	Алексей	Викторович	
4.773	Егоров	Егор	Алексеевич	
5.00	Никонова	Валерия	Ярославовна	
5.00	Миназиева	Алида	Рустамовна	
4.95	Мухаметшина	Диляра	Алмазовна	
*				

- Так же у меня есть три запроса студенты окончившие школу до 2020 года, студенты со средним баллом выше в 4.5, студенты проживающие в Казани

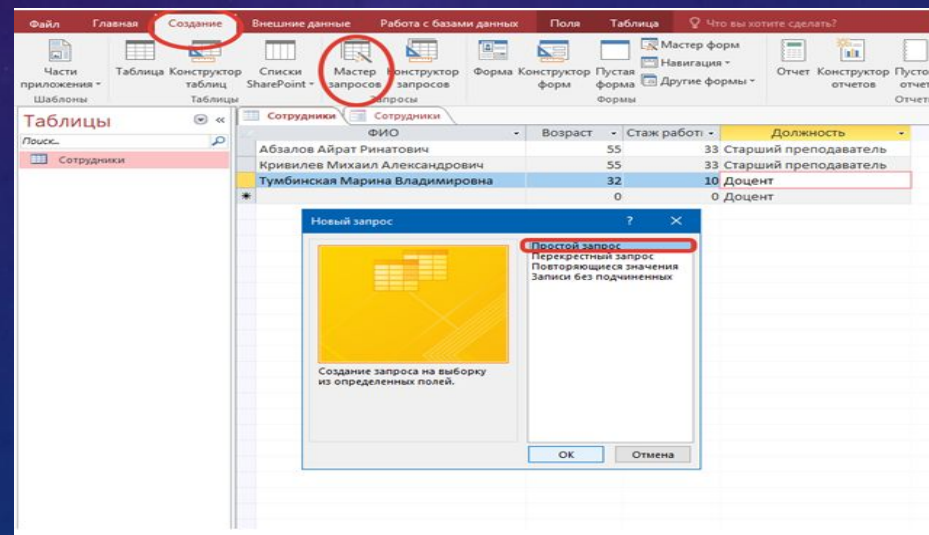
студенты со средним баллом аттестата «4,5» и выше		студенты, окончившие школу до 2020 года		
Место жительства в городе	Фамилия	Имя	Отчество	
Казань	Саитов	Ренат	Альфетович	
Казань	Федоров	Алексей	Викторович	
Казань	Егоров	Егор	Алексеевич	
Казань	Герасимов	Александр	Федорович	
Казань	Горин	Генадий	Ваисильевич	
Казань	Миназиева	Алида	Рустамовна	
Казань	Мухаметшина	Диляра	Алмазовна	
*				

студенты со средним баллом аттестата «4,5» и выше			студенты, окончившие школу до 2020 года
Фамилия	Имя	Отчество	Когда окончил
Саитов	Ренат	Альфетович	2017
Федоров	Алексей	Викторович	2017
Егоров	Егор	Алексеевич	2017
Плеханов	Захар	Ренатович	2017
Герасимов	Александр	Федорович	2017
Горин	Генадий	Ваисильевич	2017
Зеганшина	Аделя	Руслановна	2017
Никонова	Валерия	Ярославовна	2017
Миназиева	Алида	Рустамовна	2017
Мухаметшина	Диляра	Алмазовна	2017
*			

- Так же у меня в базе данных «Абитуриент» есть одна форма, специальная где хранится вся информация про ученика.

Фамилия	<input type="text" value="Саитов"/>		
Имя	<input type="text" value="Ренат"/>	Отчество	<input type="text" value="Альфетович"/>
Год рождения	Месяц	Число	
<input type="text" value="2003"/>	<input type="text" value="Сентябрь"/>	<input type="text" value="9"/>	
Место жительства в городе	<input type="text" value="Казань"/>	Место жительства в республике	<input type="text" value="Татарстан"/>
	Гражданство	<input type="text" value="РФ"/>	
Что окончил	Когда окончил	Средний бал аттестата	
<input type="text" value="МБОУ СОШ №87"/>	<input type="text" value="2017"/>	<input type="text" value="4.7"/>	
<input type="button" value="◀"/>		<input type="button" value="▶"/>	

- У вас может возникнуть вопрос: как создавать запросы с параметрами и запросы на изменение, если при создании запроса они явно не указаны? Следует отметить, что основой для всех этих запросов является запрос на выборку, т.е. сначала необходимо определить набор данных, с которым хотите работать. Затем для созданного запроса на выборку надо перейти в режим конструктора. Задание параметров производится в строке *Условия отбора* для соответствующих полей. Подробнее это будет рассмотрено ниже при выполнении задания. Для доступа к запросам на изменение надо открыть пункт меню **Запрос** — в открывшемся списке вы увидите все виды запросов на изменение.
- При выполнении запроса на выборку Access извлекает записи из таблиц и формирует результирующий набор данных. Он выглядит, как таблица, хотя и не является ею. Результирующий набор данных является *динамическим* (или *виртуальным*) набором записей и не хранится в базе данных.
- После закрытия запроса результирующий набор данных этого запроса прекращает свое существование. Хотя сам по себе динамический набор данных больше не существует, помните, что данные, которые в нем содержались, остаются в базовых таблицах.





- При сохранении запроса остается только структура запроса — перечень таблиц, список полей, порядок сортировки, ограничения на запись, тип запроса и т.д. При сохранении в базе данных запрос, по сравнению с результирующим набором данных, имеет ряд преимуществ:
- на физическом носителе информации (обычно это жесткий диск) требуется меньший объем пространства;
- запрос может использовать обновленные версии любых записей, измененных со времени последнего запуска запроса. При каждом выполнении запрос обращается к базовым таблицам и снова создает результирующий набор данных. Поскольку сам по себе результирующий набор данных не сохраняется, запрос автоматически отображает любые изменения, произошедшие в базовых таблицах с момента последнего запуска этого запроса (даже в реальном времени в многопользовательской среде).
- Для сохранения запроса следует выполнить следующие действия. Выполните команду **Файл** □ **Сохранить** или щелкните по кнопке <Сохранить> на панели инструментов. Если вы впервые сохраняете запрос, введите новое имя запроса в диалоговом окне «Сохранение».
- Пользователь имеет возможность разработать отчет самостоятельно или создать отчет с помощью *мастера*. *Мастер по разработке отчетов* выполняет всю рутинную работу и позволяет быстро разработать отчет. После вызова *Мастера* выводятся диалоговые окна с *приглашением ввести необходимые данные*, и отчет создается на основании ответов пользователя. *Мастер* необходим даже для опытных пользователей, так как позволяет быстро разработать макет, служащий основой создаваемого отчета. После этого можно переключиться в режим конструктора и внести изменения в стандартный макет.

- При работе с *мастером* в зависимости от того, какой отчет вы хотите создать (т.е. как вы отвечаете на вопросы *мастера*), Access предлагает вам различные варианты макетов отчета. Например, если вы создаете простой отчет без группировки данных, то вам предлагается три варианта макета: в столбец, табличный и выровненный. При этом в небольшом окне представляется вид этих макетов. Если вы задаете уровни группировки (т.е. по каким признакам надо сгруппировать данные, например по должности), то вам предлагаются шесть видов макетов. Перечислять их не будем, так как они хорошо проиллюстрированы в окне создания макетов.
- Основное различие между отчетами и формами заключается в их назначении. Если формы задуманы преимущественно для ввода данных, то отчеты — для просмотра данных (на экране либо на бумаге). В формах используются вычисляемые поля (обычно с помощью вычислений на основе полей в текущей записи). В отчетах вычисляемые поля (итоги) формируются на основе общей группы записей, страницы записей или всех записей отчета. Все, что можно сделать с формой (за исключением ввода данных), можно сделать и с отчетом. Действительно, форму можно сохранить в виде отчета, а затем изменить элементы управления формы в окне конструктора отчета.
- Для создания отчета надо открыть вкладку *Отчеты* и щелкнуть по кнопке <Создать>. Откроется окно «Новый отчет», в котором приведены шесть пунктов меню, т.е. шесть способов создания отчета: Конструктора.
- Мастер отчетов, Автоотчет в столбец, Автоотчет ленточный, Мастер диаграмм и Почтовые наклейки. *Конструктор* позволит вам самостоятельно создать отчет, но это не просто даже для опытного пользователя. *Мастер отчетов* автоматически создаст отчет на основе выбранных вами полей таблиц (запросов) и макетов отчетов. Этот способ создания отчетов является наиболее удобным как для начинающих, так и для опытных пользователей. *Автоотчет в столбец* и *Автоотчет ленточный* — самые простейшие способы создания отчетов: достаточно указать только имя таблицы (запроса), на основе которого будет создан отчет, а остальное сделает *мастер отчетов*. *Мастер диаграмм* поможет создать отчет в виде диаграммы, а *Почтовые наклейки* создадут отчет, отформатированный для печати почтовых наклеек

ВЫВОД: ВСЕ МОИ РАБОТЫ ПОЛУЧИЛИСЬ ОЧЕНЬ ИНФОРМАТИВНЫЕ И ПОНЯТНЫЕ, ДЕЛАЯ ИХ, Я ВНИКАЛ В СУТЬ ПРОЦЕССА, А НЕ ПРОСТО ДЕЛАЛ РАБОТУ. РАБОТЫ МНЕ ПОКАЗАЛИСЬ ОЧЕНЬ ИНТЕРЕСНЫЕ.

**СПАСИБО ЗА ВНИМАНИЕ!**