

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ ИМПЕРАТОРА АЛЕКСАНДРА I»  
(ФГБОУ ВО ПГУПС)  
ПЕТРОЗАВОДСКИЙ ФИЛИАЛ ПГУПС

ОДОБРЕНО

на заседании цикловой комиссии *ЕН*  
протокол № *8* от *28 апреля 2017*

Председатель цикловой комиссии:

*Масанцева Г.А.* ( *С.П.* )

УТВЕРЖДАЮ

Начальник УМО

*А.В. Калько*

А.В. Калько

*«28»*

*04*

2017 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**по организации и проведению практических занятий**

**по учебной дисциплине «Информатика»**

Специальность:

23.02.06	Техническая эксплуатация подвижного состава железных дорог
23.02.01	Организация перевозок и управление на транспорте (по видам)
13.02.07	Электроснабжение (по отраслям)
08.02.10	Строительство железных дорог, путь и путевое хозяйство

2017 г.

## Пояснительная записка

Выполнение практических занятий способствует повышению уровня полученных знаний, систематизации полученных знаний, активизации мышления студентов, что является активным дополнением к теоретическому объяснению материала.

Учебная дисциплина «Информатика» имеет ряд особенностей:

- компьютерные технологии выступают и как предмет изучения, и как средство активизации познавательной деятельности на занятиях;
- эта учебная дисциплина содействует углублению междисциплинарных связей за счет интеграции информационной и предметной подготовки;
- высокая мотивация студентов, уверенных в значимости знаний, полученных на занятиях по учебной дисциплине «Информатика», в своей будущей профессии.

Методические указания по организации и проведению практических занятий разработаны в соответствии с рабочей программой учебной дисциплины «Информатика» и предназначены для выполнения практических занятий обучающимися.

Освоение содержания учебной дисциплины обеспечивает достижение студентами следующих результатов:

**Предметные результаты** освоения основной образовательной программы должны отражать:

- 1) сформированность представлений о роли информации и связанных с ней процессов в окружающем мире;
- 2) владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- 3) владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц;
- 4) владение стандартными приемами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ; использование готовых прикладных компьютерных программ по выбранной специализации;
- 5) сформированность представлений о компьютерно-математических моделях и необходимости анализа соответствия модели и моделируемого объекта (процесса); о способах хранения и простейшей обработке данных; понятия о базах данных и средствах доступа к ним, умений работать с ними;
- 6) владение компьютерными средствами представления и анализа данных;
- 7) сформированность базовых навыков и умений по соблюдению требований техники безопасности, гигиены и ресурсосбережения при работе

со средствами информатизации; понимания основ правовых аспектов использования компьютерных программ и работы в Интернете.

С методической точки зрения методическая разработка отличается доступностью изложения, наличием примеров, большим количеством визуального материала, что способствует лучшему усвоению материала.

Уровень изложенного материала соответствует современным достижениям в области компьютерной техники и информационных технологий.

Методическая разработка может быть эффективно использована при изучении дисциплины «Информатика», а также при выполнении практических и самостоятельных работ студентами колледжа.

Рабочей программой предусмотрено выполнение обучающимися практических занятий, включая, как обязательный компонент практические задания с использованием персонального компьютера.

Распределение результатов освоения учебного материала в ходе выполнения заданий на практических занятиях происходит в соответствии с таблицей 1.

Таблица 1 – Распределение результатов освоения учебного материала

Раздел, тема	Контрольно-оценочные мероприятия	Результаты
		предметные
Тема 2. Информация и информационные процессы	Практическое занятие № 1-7	1) сформированность представлений о роли информации и связанных с ней процессов в окружающем мире; 7) сформированность базовых навыков и умений по соблюдению требований техники безопасности, гигиены и ресурсосбережения при работе со средствами информатизации; понимания основ правовых аспектов использования компьютерных программ и работы в Интернете.
Тема 3. Моделирование и формализация.	Практическое занятие № 8-12	5) сформированность представлений о компьютерно-математических моделях и необходимости анализа соответствия модели и моделируемого объекта (процесса); о способах хранения и простейшей обработке данных; понятия о базах данных и средствах доступа к ним, умений работать с ними; 6) владение компьютерными средствами представления и анализа данных;
Тема 4. Основы алгоритмизации и программирования	Практическое занятие №13-25	2) владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов; 3) владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц; 4) владение стандартными приемами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ; использование готовых прикладных компьютерных программ по выбранной специализации;

--	--	--

## КРИТЕРИИ ОЦЕНИВАНИЯ ПРАКТИЧЕСКИХ РАБОТ

При оценке освоенных предметных результатов при выполнении практических работ применяется пятибалльная шкала оценивания.

Оценивание практических занятий производится в соответствии со следующими нормативными актами:

- Положение о текущем контроле успеваемости и промежуточной аттестации обучающихся;
- Положение о планировании, организации и проведении лабораторных работ и практических занятий.

### **Перечень практических занятий по учебной дисциплине**

1. Переводы чисел в различных системах счисления.
2. Выполнение арифметических операций в различных системах счисления.
3. Проверка истинности логических высказываний.
4. Построение таблиц истинности логических формул.
5. Дискретное (цифровое) представление текстовой информации.
6. Дискретное (цифровое) представление графической информации.
7. Дискретное (цифровое) представление звуковой и видеoinформации. Решение задач на определение скорости передачи данных.
8. Создание и редактирование графических объектов средствами программ компьютерной графики.
9. Правила ввода и редактирования текста. Правила форматирования текста. Правила создания, редактирования списков и таблиц. Связь и внедрение объектов.
10. Моделирование интегрированных документов. Применение редактора формул и встроенного графического редактора в текстовом процессоре.
11. Использование различных возможностей динамических (электронных) таблиц для выполнения учебных заданий из различных предметных областей.
12. Организация работы с данными в БД. Формирование запросов.
13. Среда программирования.
14. Программирование линейного алгоритма.
15. Графический режим. Моделирование геометрических и физических задач.
16. Вывод на монитор фигуры, состоящей из дуг и отрезков.
17. Создание структуры Условного перехода в конструкторе блок-схем MS Visio. Условный оператор.
18. Восстановление области по заданному логическому выражению.
19. Вычисление сложной функции с помощью Оператора выбора.
20. Табулирование функций с помощью цикла с предусловием.
21. Аналитический расчет результатов выполнения циклических программ.
22. Построение графика функции.
23. Результат исполнения циклической программы в графическом режиме.
24. Ввод и вывод элементов массива. Формирование элементов массива.

25. Формирование массива с помощью генератора случайных чисел и поиск наибольшего или наименьшего элемента.

Выполнение практических работ рассчитано на 50 часов.

## Практическое занятие №1

### Переводы чисел в различных системах счисления.

**Цель:** составить сводную таблицу переводов целых чисел.

**Система счисления** — это способ записи чисел с помощью заданного набора специальных знаков (цифр).

Существуют позиционные и непозиционные системы счисления.

В **непозиционных** системах вес цифры (т.е. тот вклад, который она вносит в значение числа) не зависит от ее позиции в записи числа. Так, в римской системе счисления в числе XXXII (тридцать два) вес цифры X в любой позиции равен просто десяти.

В **позиционных** системах счисления вес каждой цифры изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число.

#### Как порождаются целые числа в позиционных системах счисления?

В каждой системе счисления цифры упорядочены в соответствии с их значениями: 1 больше 0, 2 больше 1 и т.д.

**Продвижением** цифры называют замену её следующей по величине.

Продвинуть цифру 1 значит заменить её на 2, продвинуть цифру 2 значит заменить её на 3 и т.д. **Продвижение старшей цифры** (например, цифры 9 в десятичной системе) означает замену её на 0. В двоичной системе, использующей только две цифры – 0 и 1, продвижение 0 означает замену его на 1, а продвижение 1 – замену её на 0.

Целые числа в любой системе счисления порождаются с помощью **Правил счета:**

Для образования целого числа, следующего за любым данным целым числом, нужно *продвинуть* самую правую цифру числа; если какая-либо цифра после продвижения стала нулем, то нужно продвинуть цифру, стоящую слева от неё.

Применяя это правило, запишем первые десять целых чисел

- в двоичной системе: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001;
- в троичной системе: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100;
- в пятеричной системе: 0, 1, 2, 3, 4, 10, 11, 12, 13, 14;
- восьмеричной системе: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11.

Рассмотрим только те системы счисления, которые применяются в компьютерах — десятичную, двоичную, восьмеричную и шестнадцатеричную.

Задание 1.

Заполним следующую таблицу:

Осно вание	Запись чисел в системе счисления																			
	<b>10</b>																			
<b>2</b>																				
<b>8</b>																				
<b>16</b>																				

Проверим:

Осно вание	Запись чисел в системе счисления																			
	<b>10</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
<b>2</b>	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011
<b>8</b>	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22	23
<b>16</b>	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13

### Сводная таблица переводов целых чисел

№ п./п	Перевод	№ п./п	Перевод
1	$10 \rightarrow 2$ $\begin{array}{r} 46 \overline{) 2} \\ 0 \end{array} \begin{array}{l} 2 \\ 23 \\ 1 \end{array} \begin{array}{l} 2 \\ 11 \\ 1 \end{array} \begin{array}{l} 2 \\ 5 \\ 1 \end{array} \begin{array}{l} 2 \\ 2 \\ 0 \end{array} \begin{array}{l} 2 \\ 2 \\ 1 \end{array}$	5	$2 \rightarrow 10$ $5 \ 4 \ 3 \ 2 \ 1 \ 0$ $1 \ 0 \ 1 \ 1 \ 1 \ 0_2 = 2^5 + 2^3 + 2^2 + 2^1 = 46_{10}$ Ответ: $46_{10}$
	Ответ: $101110_2$	6	$2 \rightarrow 16$ $101110_2 = 10 \ 1110_2 = 2E_{16}$ Ответ: $2E_{16}$
	Ответ: $101110_2$	7	$8 \rightarrow 2$ $56_8 = 101 \ 110_2$ Ответ: $101110_2$
2	$10 \rightarrow 8$ $\begin{array}{r} 46 \overline{) 8} \\ 6 \end{array} \begin{array}{l} 8 \\ 5 \end{array}$	8	$8 \rightarrow 10$ $1 \ 0$ $5 \ 6_8 = 5 \cdot 8^1 + 6 \cdot 8^0 = 40 + 6 = 46_{10}$ Ответ: $46_{10}$
	Ответ: $56_8$	9	$8 \rightarrow 16$ $5 \ 6_8 = 101 \ 110_2 = 10 \ 1110_2 = 2E_{16}$ Ответ: $2E_{16}$
3	$10 \rightarrow 16$ $\begin{array}{r} 46 \overline{) 16} \\ 14 \end{array} \begin{array}{l} 16 \\ 2 \end{array}$	10	$16 \rightarrow 2$ $2E_{16} = 0010 \ 1110_2 = 101110_2$ Ответ: $101110_2$
	Ответ: $2E_{16}$	11	$16 \rightarrow 8$ $2E_{16} = 10 \ 1110_2 = 101 \ 110_2 = 56_8$ Ответ: $56_8$
4	$2 \rightarrow 8$ $101110_2 = 101 \ 110_2 = 56_8$	12	$16 \rightarrow 10$ $1 \ 0$ $2 \ E_{16} = 2 \cdot 16^1 + E \cdot 16^0 = 32 + 14 = 46_{10}$ Ответ: $46_{10}$
Ответ: $56_8$			

#### Задание 2.

- |                                |                                 |                                |
|--------------------------------|---------------------------------|--------------------------------|
| 1. $389_{10} \xrightarrow{2}$  | 5. $1111011_2 \xrightarrow{8}$  | 9. $63_8 \xrightarrow{16}$     |
| 2. $101111_2 \xrightarrow{10}$ | 6. $1111011_2 \xrightarrow{16}$ | 10. $3B_{16} \xrightarrow{2}$  |
| 3. $38_{10} \xrightarrow{8}$   | 7. $52_8 \xrightarrow{2}$       | 11. $3B_{16} \xrightarrow{8}$  |
| 4. $51_{10} \xrightarrow{16}$  | 8. $63_8 \xrightarrow{10}$      | 12. $3B_{16} \xrightarrow{10}$ |



## Практическое занятие №2. Выполнение арифметических операций в различных системах счисления.

**Цель:** рассмотреть основные арифметические операции в позиционных системах счисления.

### Как производятся арифметические операции в позиционных системах счисления?

Рассмотрим основные арифметические операции: сложение, вычитание, умножение и деление. Правила выполнения этих операций в десятичной системе хорошо известны — это сложение, вычитание, умножение столбиком и деление углом. Эти правила применимы и ко всем другим позиционным системам счисления. Только таблицами сложения и умножения надо пользоваться *особыми* для каждой системы.

#### Сложение

Таблицы сложения легко составить, используя Правило Счета.

**Сложение в двоичной системе**

+	0	1
0	0	1
1	1	10

**Сложение в восьмеричной системе**

+	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**

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

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

**Пример 1.** Сложим числа 15 и 6 в различных системах счисления.

Десятичная:  $15_{10} + 6_{10}$

Двоичная:  $1111_2 + 110_2$

Восьмеричная:  $17_8 + 6_8$

$$\begin{array}{r} 1 \\ + 15 \\ + 6 \\ \hline 21 \\ \hline \end{array}$$

$5+6=11=10+1$   
 $1+1=2$

$$\begin{array}{r} 111 \\ + 1111 \\ + 0110 \\ \hline 10101 \\ \hline \end{array}$$

$1+0=1$   
 $1+1=2=2+0$   
 $1+1+1=3=2+1$   
 $1+1=2=2+0$

$$\begin{array}{r} 1 \\ + 17 \\ + 6 \\ \hline 25 \\ \hline \end{array}$$

$7+6=13=8+5$   
 $1+1=2$

**Шестнадцатеричная:**  $F_{16}+6_{16}$

**Ответ:**  $15+6 = 21_{10} = 10101_2 = 25_8 = 15_{16}$ .

$$\begin{array}{r} 1 \\ + F \\ + 6 \\ \hline 15 \\ \hline \end{array}$$

$15+6=21=16+5$

**Проверка.** Преобразуем полученные суммы к десятичному виду:

$$10101_2 = 2^4 + 2^2 + 2^0 = 16+4+1=21,$$

$$25_8 = 2 \cdot 8^1 + 5 \cdot 8^0 = 16 + 5 = 21,$$

$$15_{16} = 1 \cdot 16^1 + 5 \cdot 16^0 = 16+5 = 21.$$

**Пример 2.** Сложим числа 15, 7 и 3.

**Десятичная:**  $15_{10}+7_{10}+3_{10}$  **Двоичная:**  $1111_2+111_2+11_2$  **Восьмеричная:**  $17_8+7_8+3_8$

$$\begin{array}{r} 1 \\ + 15 \\ + 7 \\ + 3 \\ \hline 25 \\ \hline \end{array}$$

$5+7+3=15=10+5$   
 $1+1=2$

$$\begin{array}{r} 11+11 \\ + 1111 \\ + 111 \\ + 11 \\ \hline 11001 \\ \hline \end{array}$$

$1+1+1=3=2+1$   
 $1+1+1+1=4=2 \cdot 2+0$   
 $1+1=2=2+0$   
 $1+1+1=3=2+1$

$$\begin{array}{r} 2 \\ + 17 \\ + 7 \\ + 3 \\ \hline 31 \\ \hline \end{array}$$

$7+7+3=17=2 \cdot 8+1$   
 $2+1=3$

**Шестнадцатеричная:**  $F_{16}+7_{16}+3_{16}$

**Ответ:**  $5+7+3 = 25_{10} = 11001_2 = 31_8 = 19_{16}$ .

$$\begin{array}{r} F \\ + 7 \\ + 3 \\ \hline 19 \\ \hline \end{array}$$

$15+7+3=25=16+9$

**Проверка:**

$$11001_2 = 2^4 + 2^3 + 2^0 = 16+8+1=25,$$

$$31_8 = 3 \cdot 8^1 + 1 \cdot 8^0 = 24 + 1 = 25,$$

$$19_{16} = 1 \cdot 16^1 + 9 \cdot 16^0 = 16+9 = 25.$$

**Пример 3.** Сложим числа 141,5 и 59,75.

**Десятичная:**  $141,5_{10} + 59,75_{10}$  **Двоичная:**  $10001101,1_2 + 111011,11_2$

$$\begin{array}{r} 111 \\ + 141,50 \\ + 59,75 \\ \hline 201,25 \\ \hline \end{array}$$

$0+5=5$   
 $5+7=12=10+2$   
 $1+9+1=11=10+1$   
 $4+5+1=10=10+0$   
 $1+1=2$

$$\begin{array}{r} 1111111 \\ + 10001101,1 \\ + 111011,11 \\ \hline 11001001,01 \\ \hline \end{array}$$

$1+0=1$   
 $1+1=2=2+0$   
 $1+1=2=2+0$   
 $1+1+1=3=2+1$   
 $1+1=2=2+0$   
 $1+1=2=2+0$   
 $1+1=2=2+0$   
 $1+1=2=2+0$

**Восьмеричная:**  $215,4_8 + 73,6_8$

**Шестнадцатеричная:**  $8D,8_{16} + 3B,C_{16}$

$$\begin{array}{r} 111 \\ + 215,4 \\ + 73,6 \\ \hline 311,2 \\ \hline \end{array}$$

$4+6=10=8+2$   
 $5+3+1=9=8+1$   
 $1+7+1=9=8+1$   
 $2+1=3$

$$\begin{array}{r} 11 \\ + 8D,8 \\ + 3B,C \\ \hline C9,4 \\ \hline \end{array}$$

$8+12=20=16+4$   
 $13+11+1=25=16+9$   
 $8+3+1=12=C_{16}$

**Ответ:**  $141,5 + 59,75 = 201,25_{10} = 11001001,01_2 = 311,2_8 = C9,4_{16}$

**Проверка.** Преобразуем полученные суммы к десятичному виду:

$$11001001,01_2 = 2^7 + 2^6 + 2^3 + 2^0 + 2^{-2} = 201,25$$

$$311,2_8 = 3 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 + 2 \cdot 8^{-1} = 201,25$$

$$C9,4_{16} = 12 \cdot 16^1 + 9 \cdot 16^0 + 4 \cdot 16^{-1} = 201,25$$

**Вычитание**

**Пример 4.** Вычтем единицу из чисел  $10_2$ ,  $10_8$  и  $10_{16}$

$\begin{array}{r} 1 \\ - 10 \\ \hline 1 \\ \hline \underline{2-1=1} \end{array}$	$\begin{array}{r} 1 \\ - 10 \\ \hline 7 \\ \hline \underline{8-1=7} \end{array}$	$\begin{array}{r} 1 \\ - 10 \\ \hline F \\ \hline \underline{16-1=15=F_{16}} \end{array}$
		Заемы

**Пример 5.** Вычтем единицу из чисел  $100_2$ ,  $100_8$  и  $100_{16}$ .

$\begin{array}{r} 1 \\ - 100 \\ \hline 11 \\ \hline \underline{2-1=1} \\ \underline{1-0=1} \end{array}$	$\begin{array}{r} 1 \\ - 100 \\ \hline 77 \\ \hline \underline{8-1=7} \\ \underline{7-0=7} \end{array}$	$\begin{array}{r} 1 \\ - 100 \\ \hline FF \\ \hline \underline{16-1=15=F_{16}} \\ \underline{1+1=2} \end{array}$
		Заемы

**Пример 6.** Вычтем число  $59,75$  из числа  $201,25$ .

**Десятичная:**  $201,25_{10} - 59,75_{10}$       **Двоичная:**  $11001001,01_2 - 11011,11_2$

$\begin{array}{r} 1 \quad 1 \\ - 201,25 \\ \hline 59,75 \\ \hline 141,50 \\ \hline \underline{5-5=0} \\ \underline{10+2-7-5} \\ \underline{10-9=1} \\ \underline{9-5=4} \\ \underline{2-1=1} \end{array}$	$\begin{array}{r} 1 \quad 1 \quad 1 \\ - 11001001,01 \\ \hline 00111011,11 \\ \hline 10001101,10 \\ \hline \underline{1-0=1} \\ \underline{0-0=0} \\ \underline{1-1=0} \\ \underline{1-1=0} \\ \underline{2-1=1} \end{array}$	$\begin{array}{r} 1 \quad 1 \quad 1 \\ - 11011,11 \\ \hline 1001101,10 \\ \hline \underline{1-1=0} \\ \underline{2-1=1} \\ \underline{1-1=0} \\ \underline{1-0=1} \end{array}$
		Заемы

**Восьмеричная:**  $311,2_8 - 73,6_8$

**Шестнадцатеричная:**  $C9,4_{16} - 3B,C_{16}$

$\begin{array}{r} 1 \quad 1 \quad 1 \\ - 311,2 \\ \hline 73,6 \\ \hline 215,4 \\ \hline \underline{8+2-6=4} \\ \underline{8-3=5} \\ \underline{8-7=1} \end{array}$	$\begin{array}{r} 1 \quad 1 \\ - C9,4 \\ \hline 3B,C \\ \hline 8D,8 \\ \hline \underline{16+4-12=8} \\ \underline{16+8-11=13=D_{16}} \\ \underline{12-1-3=8} \end{array}$
--	---

**Ответ:**  $201,25_{10} - 59,75_{10} = 141,5_{10} = 10001101,1_2 = 215,4_8 = 8D,8_{16}$ .

**Проверка.** Преобразуем полученные разности к десятичному виду:

$$10001101,1_2 = 2^7 + 2^3 + 2^2 + 2^0 + 2^{-1} = 141,5;$$

$$215,4_8 = 2 \cdot 8^2 + 1 \cdot 8^1 + 5 \cdot 8^0 + 4 \cdot 8^{-1} = 141,5;$$

$$8D,8_{16} = 8 \cdot 16^1 + D \cdot 16^0 + 8 \cdot 16^{-1} = 141,5.$$

**Умножение**

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо

заимствовать из соответствующих рассматриваемой системе таблиц умножения и сложения.

**Умножение в двоичной системе**

*	0	1
0	0	0
1	0	1

**Умножение в восьмеричной системе**

*	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

Ввиду чрезвычайной простоты таблицы умножения в двоичной системе, умножение сводится лишь к сдвигам множимого и сложениям.

**Пример 7.** Перемножим числа 5 и 6.

Десятичная:  $5_{10} \cdot 6_{10}$

$$\begin{array}{r} 5 \\ \times 6 \\ \hline 30 \end{array}$$

Двоичная:  $101_2 \cdot 110_2$

$$\begin{array}{r} 101 \\ \times 110 \\ \hline 101 \\ 101 \\ \hline 11110 \end{array}$$

Восьмеричная:  $5_8 \cdot 6_8$

$$\begin{array}{r} 5 \\ \times 6 \\ \hline 36 \end{array}$$

**Ответ:**  $5 \cdot 6 = 30_{10} = 11110_2 = 36_8$ .

**Проверка.** Преобразуем полученные произведения к десятичному виду:

$$11110_2 = 2^4 + 2^3 + 2^2 + 2^1 = 30;$$

$$36_8 = 3 \cdot 8^1 + 6 \cdot 8^0 = 30.$$

**Пример 8.** Перемножим числа 115 и 51.

Десятичная:  $115_{10} \cdot 51_{10}$

$$\begin{array}{r} 115 \\ \times 51 \\ \hline 115 \\ 575 \\ \hline 5865 \end{array}$$

Двоичная:  $1110011_2 \cdot 110011_2$

$$\begin{array}{r} 1110011 \\ \times 110011 \\ \hline 1110011 \\ 1110011 \\ \hline 1110011 \\ 1110011 \\ \hline 1011011101001 \end{array}$$

Восьмеричная:  $163_8 \cdot 63_8$

$$\begin{array}{r} 163 \\ \times 63 \\ \hline 531 \\ 1262 \\ \hline 13351 \end{array}$$

**Ответ:**  $115 \cdot 51 = 5865_{10} = 1011011101001_2 = 13351_8$ .

**Проверка.** Преобразуем полученные произведения к десятичному виду:

$$1011011101001_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^6 + 2^5 + 2^3 + 2^0 = 5865;$$

$$13351_8 = 1 \cdot 8^4 + 3 \cdot 8^3 + 3 \cdot 8^2 + 5 \cdot 8^1 + 1 \cdot 8^0 = 5865.$$

**Деление**

Деление в любой позиционной системе счисления производится по тем же правилам, как и деление углом в десятичной системе. В двоичной системе деление выполняется особенно просто, ведь очередная цифра частного может быть только нулем или единицей.

**Пример 9.** Разделим число 30 на число 6.

Десятичная:  $30_{10} : 6_{10}$

Двоичная:  $11110_2 : 110_2$

Восьмеричная:  $36_8 : 6_8$

$$\begin{array}{r} - 30 \mid 6 \\ \underline{30} \mid 5 \\ 0 \end{array}$$

$$\begin{array}{r} - 11110 \mid 110 \\ \underline{110} \mid 101 \\ 110 \\ \underline{110} \\ 0 \end{array}$$

$$\begin{array}{r} - 36 \mid 6 \\ \underline{36} \mid 5 \\ 0 \end{array}$$

Ответ:  $30 : 6 = 5_{10} = 101_2 = 5_8$ .

**Пример 10.** Разделим число 5865 на число 115.

Десятичная:  $5865_{10} : 115_{10}$  Двоичная:  $1011011101001_2 : 1110011_2$

$$\begin{array}{r} - 5865 \mid 115 \\ \underline{575} \mid 51 \\ - 115 \\ \underline{115} \\ 0 \end{array}$$

$$\begin{array}{r} - 1011011101001 \mid 1110011 \\ \underline{1110011} \mid 110011 \\ - 1000100 \\ - 1110011 \\ \underline{10101100} \\ - 1110011 \\ \underline{1110011} \\ 1110011 \\ \underline{1110011} \\ 0 \end{array}$$

Восьмеричная:  $13351_8 : 163_8$

$$\begin{array}{r} - 13351 \mid 163 \\ \underline{1262} \mid 63 \\ - 531 \\ \underline{531} \\ 0 \end{array}$$

Ответ:  $5865 : 115 = 51_{10} = 110011_2 = 63_8$ .

**Проверка.** Преобразуем полученные частные к десятичному виду:  
 $110011_2 = 2^5 + 2^4 + 2^1 + 2^0 = 51$ ;  $63_8 = 6 \cdot 8^1 + 3 \cdot 8^0 = 51$ .

**Пример 11.** Разделим число 35 на число 14.

$$\begin{array}{r} - 35 \mid 14 \\ \underline{28} \mid 2,5 \\ - 70 \\ \underline{70} \\ 0 \end{array}$$

$$\begin{array}{r} - 100011 \mid 1110 \\ \underline{1110} \mid 10,1 \\ - 1110 \\ \underline{1110} \\ 0 \end{array}$$

Десятичная:  $35_{10} : 14_{10}$

Двоичная:  $100011_2 : 1110_2$

Восьмеричная:  $43_8 : 16_8$

$$\begin{array}{r} - 43 \mid 16 \\ \underline{34} \mid 2,4 \\ - 70 \\ \underline{70} \\ 0 \end{array}$$

Ответ:  $35 : 14 = 2,5_{10} = 10,1_2 = 2,4_8$ .

**Проверка.** Преобразуем полученные частные к десятичному виду:

$$10,1_2 = 2^1 + 2^{-1} = 2,5;$$

$$2,4_8 = 2 \cdot 8^0 + 4 \cdot 8^{-1} = 2,5.$$

### Практическое занятие №3.

#### Проверка истинности логических высказываний.

**Цель:** научиться проверять истинность логических высказываний.

Давайте в качестве разминки решим несколько задач по математической логике из демо-версий ЕГЭ прошлых лет.

Для какого числа  $X$  истинно высказывание

**A9.**  $((X > 3) \vee (X < 3)) \rightarrow (X < 1)$

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

Проверим все 4 возможных варианта значения  $X$  и выберем из них тот вариант, когда значением выражения будет истина. Вызывает ученика к доске.

Решение:

**$X=1$**

Выражение будет иметь вид:  $((1 > 3) \vee (1 < 3)) \rightarrow (1 < 1)$ . Определим истинность каждого высказывания.  $(0 \vee 1) \rightarrow 0$ , откуда  $1 \rightarrow 0 = 0$

**$X=2$**

Выражение будет иметь вид:  $((2 > 3) \vee (2 < 3)) \rightarrow (2 < 1)$ . Определим истинность каждого высказывания.  $(0 \vee 1) \rightarrow 0$ , откуда  $1 \rightarrow 0 = 0$

**$X=3$**

Выражение будет иметь вид:  $((3 > 3) \vee (3 < 3)) \rightarrow (3 < 1)$ . Определим истинность каждого высказывания.  $(0 \vee 0) \rightarrow 0$ , откуда  $0 \rightarrow 0 = 1$

**$X=4$**

Выражение будет иметь вид:  $((4 > 3) \vee (4 < 3)) \rightarrow (4 < 1)$ . Определим истинность каждого высказывания.  $(1 \vee 0) \rightarrow 0$ , откуда  $1 \rightarrow 0 = 0$

Ответ: верный ответ № 3.

**A11.** Символом  $F$  обозначено одно из указанных ниже логических выражений от трех аргументов:  $X, Y, Z$ .

Дан фрагмент таблицы истинности выражения  $F$ :

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

Какое выражение соответствует  $F$ ?

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

Составим таблицы истинности для каждого высказывания, и сравним результат с  $F$ .

X	Y	Z	$\neg X$	$\neg Y$	$\neg Z$	$\neg X \vee Y \vee \neg Z$	$X \wedge Y \wedge \neg Z$	$\neg X \wedge \neg Y \wedge Z$	$X \vee \neg Y \vee Z$	F
0	1	0	1	0	1	1	0	0	0	0
1	1	0	0	0	1	1	1	0	1	1
1	0	1	0	1	0	0	0	0	1	0

Ответ: верный ответ 2

**В2.** Каково наибольшее целое число  $X$ , при котором истинно высказывание  $(90 < X \cdot X) \rightarrow (X < (X - 1))$  ?

Вспомним таблицу истинности импликации. Импликация истинна в трёх случаях:

$0 \rightarrow 0$ ;

$0 \rightarrow 1$ ;

$1 \rightarrow 1$ ;

Рассуждаем, т.к. вторая часть выражения  $X < (X - 1)$  всегда ложна, остается только первый случай. ( $0 \rightarrow 0$ ) ; Т.о наибольшее целое число  $X$ , при котором  $90 < X \cdot X$  ложно равно **9**

**А10.** Какое логическое выражение равносильно выражению  $\neg (A \wedge B) \wedge \neg C$ ?

1)  $\neg A \vee B \vee \neg C$

2)  $(\neg A \vee \neg B) \wedge \neg C$

3)  $(\neg A \vee \neg B) \wedge C$

4)  $\neg A \wedge \neg B \wedge \neg C$

### Рассмотрим табличный способ решения логических задач.

При использовании этого способа условия, которые содержит задача, и результаты рассуждений фиксируются с помощью специально составленных таблиц.

**Пример 1.** В симфонический оркестр приняли на работу трёх музыкантов: Брауна, Смита и Вессона, умеющих играть на скрипке, флейте, альте, кларнете, гобое и трубе.

Известно, что:

1. Смит самый высокий;
2. играющий на скрипке меньше ростом играющего на флейте;
3. играющие на скрипке и флейте и Браун любят пиццу;
4. когда между альтистом и трубачом возникает ссора, Смит мирит их;
5. Браун не умеет играть ни на трубе, ни на гобое.

На каких инструментах играет каждый из музыкантов, если каждый владеет двумя инструментами?

**Решение.** Составим таблицу и отразим в ней условия задачи, заполнив соответствующие клетки цифрами 0 и 1 в зависимости от того, ложно или истинно соответствующее высказывание.

Так как музыкантов трое, инструментов шесть и каждый владеет только двумя инструментами, получается, что каждый музыкант играет на инструментах, которыми остальные не владеют.

Из условия 4 следует, что Смит не играет ни на альте, ни на трубе, а из условий 3 и 5, что Браун не умеет играть на скрипке, флейте, трубе и гобое. Следовательно, инструменты Брауна — альт и кларнет. Занесем это в таблицу, а оставшиеся клетки столбцов "альт" и "кларнет" заполним нулями:

	скрипка	флейта	альт	кларнет	гобой	труба
Браун	0	0	1	1	0	0
Смит			0	0		0
Вессон			0	0		

Из таблицы видно, что на трубе может играть только Вессон.

Из условий 1 и 2 следует, что Смит не скрипач. Так как на скрипке не играет ни Браун, ни Смит, то скрипачом является Вессон. Оба инструмента, на которых играет Вессон, теперь определены, поэтому остальные клетки строки "Вессон" можно заполнить нулями:

	скрипка	флейта	альт	кларнет	гобой	труба
Браун	0	0	1	1	0	0
Смит	0		0	0		0
Вессон	1	0	0	0	0	1

Из таблицы видно, что играть на флейте и на гобое может только Смит.

	скрипка	флейта	альт	кларнет	гобой	труба
Браун	0	0	1	1	0	0
Смит	0	1	0	0	1	0
Вессон	1	0	0	0	0	1

**Ответ:** Браун играет на альте и кларнете, Смит — на флейте и гобое, Вессон — на скрипке и трубе.

**Пример 2.** Три одноклассника — Влад, Тимур и Юра, встретились спустя 10 лет после окончания школы. Выяснилось, что один из них стал врачом, другой физиком, а третий юристом. Один полюбил туризм, другой бег, страсть третьего — регби.

Юра сказал, что на туризм ему не хватает времени, хотя его сестра — единственный врач в семье, заядлый турист. Врач сказал, что он разделяет увлечение коллеги.

Забавно, но у двоих из друзей в названиях их профессий и увлечений не встречается ни одна буква их имен.

Определите, кто чем любит заниматься в свободное время и у кого какая профессия.

**Решение.** Здесь исходные данные разбиваются на тройки (имя — профессия — увлечение).

Из слов Юры ясно, что он не увлекается туризмом и он не врач. Из слов врача следует, что он турист.

Имя	Юра		
Профессия		врач	
Увлечение		туризм	

Буква "а", присутствующая в слове "врач", указывает на то, что Влад тоже не врач, следовательно врач — Тимур. В его имени есть буквы "т" и "р", встречающиеся в слове "туризм", следовательно второй из друзей, в названиях профессии и увлечения которого не встречается ни одна буква его имени — Юра. Юра не юрист и не регбист, так как в его имени содержатся буквы "ю" и "р".

Следовательно, окончательно имеем:

Имя	Юра	Тимур	Влад
Профессия	физик	врач	юрист



Увлечение	бег	туризм	регби
-----------	-----	--------	-------

**Ответ.** Влад — юрист и регбист, Тимур — врач и турист, Юра — физик и бегун.

**Пример 3.** Три дочери писательницы Дорис Кей — Джуди, Айрис и Линда, тоже очень талантливы. Они приобрели известность в разных видах искусств — пении, балете и кино. Все они живут в разных городах, поэтому Дорис часто звонит им в Париж, Рим и Чикаго.

Известно, что:

1. Джуди живет не в Париже, а Линда — не в Риме;
2. парижанка не снимается в кино;
3. та, кто живет в Риме, певица;
4. Линда равнодушна к балету.

Где живет Айрис, и какова ее профессия?

**Решение.** Составим таблицу и отразим в ней условия 1 и 4, заполнив клетки цифрами 0 и 1 в зависимости от того, ложно или истинно соответствующее высказывание:

Париж	Рим	Чикаго		Пение	Балет	Кино
0			Джуди			
			Айрис			
	0		Линда		0	

Далее рассуждаем следующим образом. Так как Линда живет не в Риме, то, согласно условию 3, она не певица. В клетку, соответствующую строке "Линда" и столбцу "Пение", ставим 0.

Из таблицы сразу видно, что Линда киноактриса, а Джуди и Айрис не снимаются в кино.

Париж	Рим	Чикаго		Пение	Балет	Кино
0			Джуди			0
			Айрис			0
	0		Линда	0	0	1

Согласно условию 2, парижанка не снимается в кино, следовательно, Линда живет не в Париже. Но она живет и не в Риме. Следовательно, Линда живет в Чикаго. Так как Линда и Джуди живут не в Париже, там живет Айрис. Джуди живет в Риме и, согласно условию 3, является певицей. А так как Линда киноактриса, то Айрис балерина.

В результате постепенного заполнения получаем следующую таблицу:

Париж	Рим	Чикаго		Пение	Балет	Кино
0	0	1	Джуди	1	0	0
1	0	0	Айрис	0	1	0
0	0	1	Линда	0	0	1

**Ответ.** Айрис балерина. Она живет в Париже.

**Задание:** Решите самостоятельно.

Студенты II курса Дмитрий, Андрей, Кирилл и Максим занимались факультативными занятиями по информатике, экологии, философии и социологии.

Известно, что каждый студент занимался только одним факультативом, и никакой факультатив не посещали два студента. Кирилл и Максим никогда не посещали факультатив информатики. Дмитрий вместе со студентом, посещавшим факультатив социологии, ходили в гости к любителю философии. Кирилл никогда не посещал факультативных занятий по философии, а Дмитрий на последнем факультативе изучал влияние кислотных дождей на окружающую среду.

Кто из ребят занимался философией?

### Практическое занятие №4.

#### Построение таблиц истинности логических формул.

**Цель:** научиться составлять таблицы истинности.

**Таблица истинности логической формулы выражает соответствие между всевозможными наборами значений переменных и значениями формулы.**

Для формулы, которая содержит две переменные, таких наборов значений переменных всего четыре: (0,0), (0,1), (1,0), (1,1).

Если формула содержит три переменные, то возможных наборов значений переменных восемь:

(0,0,0), (0,0,1), (0,1,0), (0,1,1),

(1,0,0), (1,0,1), (1,1,0), (1,1,1).

Количество наборов для формулы с четырьмя переменными равно шестнадцати и т.д.

Удобной формой записи при нахождении значений формулы является таблица, содержащая кроме значений переменных и значений формулы также и значения промежуточных формул.

#### Примеры.

1. Составим таблицу истинности для формулы  $x \cdot y \vee \overline{x \vee y} \vee x$ , которая содержит две переменные  $x$  и  $y$ . В первых двух столбцах таблицы запишем четыре возможных пары значений этих переменных, в последующих столбцах — значения промежуточных формул и в последнем столбце — значение формулы. В результате получим таблицу:

Переменные		Промежуточные логические формулы					Формула
$x$	$y$	$\overline{x}$	$\overline{x \cdot y}$	$x \vee y$	$\overline{x \vee y}$	$x \cdot y \vee \overline{x \vee y} \vee x$	

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

Из таблицы видно, что при всех наборах значений переменных  $x$  и  $y$  формула  $x \cdot y \vee x \vee y \vee x$  принимает значение 1, то есть является *тождественно истинной*.

2. Таблица истинности для формулы  $\overline{x \vee y} \cdot (x \cdot \overline{y})$ :

Переменные		Промежуточные логические формулы				Формула
$x$	$y$	$x \vee y$	$\overline{x \vee y}$	$\overline{y}$	$x \cdot \overline{y}$	$\overline{x \vee y} \cdot (x \cdot \overline{y})$
0	0	0	1	1	0	0
0	1	1	0	0	0	0
1	0	1	0	1	1	0
1	1	1	0	0	0	0

Из таблицы видно, что при всех наборах значений переменных  $x$  и  $y$  формула  $\overline{x \vee y} \cdot (x \cdot \overline{y})$  принимает значение 0, то есть является *тождественно ложной*.

3. Таблица истинности для формулы  $\overline{\overline{x \vee y} \vee \overline{x \cdot z}}$ :

Переменные			Промежуточные логические формулы					Формула
$x$	$y$	$z$	$\overline{y}$	$x \vee \overline{y}$	$\overline{\overline{x \vee y}}$	$\overline{x}$	$\overline{x \cdot z}$	$\overline{\overline{x \vee y} \vee \overline{x \cdot z}}$
0	0	0	1	1	0	1	0	0
0	0	1	1	1	0	1	1	1
0	1	0	0	0	1	1	0	1
0	1	1	0	0	1	1	1	1
1	0	0	1	1	0	0	0	0
1	0	1	1	1	0	0	0	0
1	1	0	0	1	0	0	0	0
1	1	1	0	1	0	0	0	0

Из таблицы видно, что формула  $\overline{\overline{x \vee y} \vee \overline{x \cdot z}}$  в некоторых случаях принимает значение 1, а в некоторых — 0, то есть является *выполнимой*.

**Выполнить задание:** Составить таблицу истинности:

1 $\overline{x \wedge (y \wedge \bar{z} \vee x \wedge \bar{z})}$	2 $\overline{x \wedge \bar{y} \wedge (x \wedge y \wedge \bar{z} \vee x \wedge y)}$
3 $\overline{x \wedge \bar{y} \wedge x \wedge \bar{y} \wedge \bar{z} \vee x \wedge \bar{z}}$	4 $\overline{x \vee \bar{y} \wedge z \vee x \vee y \wedge z}$
5 $(x \wedge \overline{y \wedge z \vee \bar{y} \wedge z}) \wedge (\overline{x \wedge \bar{y} \vee y \wedge \bar{z}})$	6 $\overline{x \wedge y \wedge (x \wedge \bar{y} \wedge z \vee \bar{x} \wedge y)}$
7 $\overline{x \wedge z \wedge (x \wedge y \wedge z \vee \bar{x} \wedge \bar{y})}$	8 $\overline{x \wedge \bar{y} \vee z \wedge x \wedge y \vee z}$
9 $x \wedge y \wedge z \vee \bar{x} \wedge \bar{y} \wedge z$	10 $x \wedge y \wedge z \vee x \wedge y \wedge \bar{z} \vee x \wedge \bar{y} \vee \bar{x}$
11 $x \wedge \bar{y} \wedge \bar{z} \vee x \wedge y \wedge \bar{z} \vee \bar{x} \wedge \bar{z} \vee x \wedge$	12 $\overline{x \vee y \vee x \wedge y \vee y \wedge z}$

### Практическое занятие №5.

#### Дисктерное (цифровое) представление текстовой информации.

**Цель:** научиться рассчитывать информационный объем текстового файла.

В компьютере каждый символ кодируется уникальным кодом.

Принято интернациональное соглашение о присвоении каждому символу своего уникального кода. В качестве международного стандарта принята кодовая таблица ASCII (American Standard Code for Information Interchange)

В этой таблице представлены коды от 0 до 127 (буквы английского алфавита, знаки математических операций, служебные символы и т.д.), причем коды от 0 до 32 отведены не символам, а функциональным клавишам. Запишите название этой кодовой таблицы и диапазон кодируемых символов.

Коды с 128 по 255 выделены для национальных стандартов каждой страны. Этого достаточно для большинства развитых стран.

Для России были введены несколько различных стандартов кодовой таблицы (коды с 128 по 255).

Вот некоторые из них. Рассмотрим и запишем их названия:

**КОИ8-Р, CP1251, CP866, Mac, ISO.**

Понятие кодировки Unicode

В мире существует примерно 6800 различных языков. Если прочитать текст, напечатанный в Японии на компьютере в России или США, то понять его будет нельзя. Чтобы буквы любой страны можно было читать на любом компьютере, для их кодировки стали использовать два байта (16 бит).

Сколько символов можно закодировать двумя байтами? (Для слабоуспевающих учащихся можно предложить им воспользоваться инженерным калькулятором).

Ответ:  $65536=2^{16}$

Такая кодировка называется Unicode и обозначается как UCS-2. Этот код включает в себя все существующие алфавиты мира, а также множество математических, музыкальных, химических символов и многое другое. Существует кодировка и UCS-4, где для кодирования используют 4 байта, то есть можно кодировать более 4 млрд. символов.

Расчет количества текстовой информации

Так как каждый символ кодируется 1 байтом, то информационный объем текста можно узнать, умножив количество символов в тексте на 1 байт.

**Проверим это на практике.** Включите монитор, создайте текстовый документ в редакторе Блокнот и напечатайте в нём пословицу: “Ученье – атаман, а неученье – комар”. Сколько в ней символов?

Ответ: 36

Сохраните и закройте файл. Определите его объем в байтах. Каков он?

Ответ: 36 байт.

**Разбор и решение задач**

а) Лазерный принтер Canon LBP печатает со скоростью в среднем 6,3 Кбит в секунду. Сколько времени понадобится для распечатки 8-ми страничного документа, если известно, что на одной странице в среднем по 45 строк, в строке 70 символов (1 символ – 1 байт)

Решение:

1) Находим количество информации, содержащейся на 1 странице:

$$45 * 70 * 8 \text{ бит} = 25200 \text{ бит}$$

2) Находим количество информации на 8 страницах:

$$25200 * 8 = 201600 \text{ бит}$$

3) Приводим к единым единицам измерения. Для этого Мбиты переводим в биты:

$$6,3 * 1024 = 6451,2 \text{ бит/сек.}$$

4) Находим время печати:  $201600 : 6451,2 \approx 31$  секунда.

б) Текстовое сообщение, содержащее 1048576 символов общепринятой кодировки, необходимо разместить на дискете ёмкостью 1,44Мб. Какая часть дискеты будет занята?

Дано:

$K=1048576$  символов;

$i=8$  бит/символ

Решение:

$$V=K*i=1048576*8=8388608\text{бит}=1048576\text{байт}=1024 \text{ Кб}=1\text{Мб,}$$

что составляет  $1\text{Мб}*100\%/1,44\text{Мб}=69\%$  объёма дискеты

Ответ: 69% объёма дискеты будет занято переданным сообщением

в) Информация в кодировке Unicode передается со скоростью 128 знаков в секунду в течение 32 минут. Какую часть дискеты ёмкостью 1,44Мб займёт переданная информация?

Дано:

$v=128$  символов/сек;

$t=32$  минуты= $1920$ сек;

$i=16$  бит/символ

Решение:

$$K=v*t=245760\text{символов}$$

$$V=K*i=245760*16=3932160\text{бит}=491520\text{байт}=480 \text{ Кб}=0,469\text{Мб,}$$

что составляет  $0,469\text{Мб}*100\%/1,44\text{Мб}=33\%$  объёма дискеты

Ответ: 33% объёма дискеты будет занято переданным сообщением

## Практическое занятие № 6.

### Дисктерное (цифровое) представление графической информации.

**Цель:** научиться рассчитывать информационный объем графического файла.

#### Цифровое представление графической информации.

Существует два метода представления графической информации растровый и векторный. Растровое изображение – мозаика из точек – пикселей, каждый пиксель окрашен в определенный цвет.

Количество информации, которое необходимо для хранения  $N$  цветов пикселей называется битовой глубиной ( $i$ ).

$$N=2^{i(\text{бит})} \rightarrow I = \log_2 N$$

1. Черно-белое изображение на экране монитора размером  $20 \times 10$  занимает 3 страницы видеопамати. Вычислить размер файла:

$N=2$ $R \times St = 20 \times 10$ $P=3$ <hr style="border: 0.5px solid black;"/> $I=?$	$I = R \times St \times P \times i$ $i = \log_2 2 = 1 \text{ бит}$ $I = 20 \times 10 \times 3 \times 1 \text{ бит} = 600 \text{ бит}$
--	---

2. На экране дисплея необходимо отображать 4096 цветов. Поместится ли файл, занимающий 3 страницы видеопамати при разрешающей способности монитора  $320 \times 240$ , на дискете. Ответ обосновать.

$N=4096$ $R \times St = 320 \times 240$ $M_d = 1,4 \text{ Мбайт}$ $P=3$ <hr style="border: 0.5px solid black;"/> $I > M?$ $I < M?$	$I = R \times St \times P \times i$ $N = 2^i; 4096 = 2^{12}; i = 12 \text{ бит}$ или $i = \log_2 4096 = \log_2 2^{12} \text{ бит} = 12 \text{ бит}$ $I = 320 \times 240 \times 3 \times 12 \text{ бит} = 3 \times 320 \times 240 \times 12 \text{ бит} \approx 0,33 \text{ Мб}$  $I < M \rightarrow \text{поместится}$
--	---

**Реши задачи самостоятельно и сверься с решениями.**

а) Документ содержит точечную черно-белую фотографию  $10 \times 15$  см. Каждый квадратный сантиметр содержит 600 точек, каждая точка описывается 4 битами. Каков общий информационный объем документа в килобайтах?

**Решение.** Вычислим общее количество точек, содержащихся в фотографии. Обратите внимание, что 600 точек содержит не линейный сантиметр, а квадратный. Таким образом общее число точек будет  $10 \times 15 \times 600 = 9000$  точек. Поскольку точка описывается 4 битами, то общее число бит  $9000 \times 4 = 36000$  бит.

Переведем биты в байты и получим  $36000 : 8 = 4500$  байт  
 Переведем байты в килобайты  $4500 : 1024 = 4,39$  килобайт.

б) Графическое 16 цветное изображение имеет размер 256 пикселей на 200 пикселей. Какое место в памяти оно занимает?

**Решение:** Для представления 16 цветного изображения требуется  $\log_2 16 = 4$  бита, следовательно, цвет пикселя кодируется 4 битами. Размер изображения 256 на 200, значит количество информации в картинке  $256 * 200 * 4 = 204\,800$  байт = 200 Кб. Изображение занимает в памяти 200 килобайт

$b$	$K=2^b$
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
16	65536
24	16777216

в) Какое максимально возможное количество цветов в изображении, если цвет кодируется 24 битами?

**Решение:** Для того чтобы вычислить максимальное количество цветов, достаточно возвести два в степень 24.

$2^{24}=16\ 777\ 216$ . 24 битами можно представить 16 777 216 цветов.

г) Глаз человека способен различать порядка 4 тысяч цветов, сколько бит достаточно для представления такого количества?

**Решение:** Для ответа на вопрос задачи нужно решить уравнение  $\log_2 4000 = x$ ; или эквивалентное ему  $2^x = 4000$ . Поскольку  $2^{12} = 4096$ , то достаточно 12 бит по 4 бита на составляющие красного, зеленого и синего цвета. Для представления 4 тысяч цветов достаточно 12 бит.

д) Цветное растровое графическое изображение, палитра которого включает в себя 65 536 цветов, имеет размер 100 x 100 точек (пикселей). Какой объем видеопамати компьютера (в Кбайтах) занимает это изображение в формате BMP?

**Решение:**  $65536 = 2^{16}$ , I = 16 бит на кодирование 1 цвета. Все изображение состоит из  $10 \times 10 = 10\ 000$  точек. Следовательно, количество информации, необходимое для хранения изображения целиком  $16 \times 10\ 000 = 160\ 000$  бит = 20 000 байт = 19,5 Кб.

е) Сколько бит видеопамати занимает информация об одном пикселе на черно-белом экране?

**Решение.** Для черно-белого изображения N=2. Отсюда b=1 пиксель.

ж) Сколько цветов может отображать монитор, если для хранения состояния каждого пикселя отводится 24 бита?

**Решение.** Определяем по таблице N для b=24. N=16777216 цветов.

з) На экране с разрешающей способностью 800x600 высвечивается 16-цветное изображение. Определить объем одной страницы видеопамати в килобайтах.

**Решение.**

1) Определяем по таблице количество бит видеопамати b, необходимых для хранения состояния одного пикселя при N=16. b=4 бита.

2) Определяем необходимый объем видеопамати в байтах:  $4 \times 800 \times 600 / 8 = 240000$  байт

3) Определяем объем памяти одной страницы в килобайтах:  $240000 / 1024 = 234,375$  Кб

и) Объем видеопамати равен 1 Мб. Разрешающая способность дисплея - 800x600. Какое максимальное количество цветов можно использовать при условии, что видеопамать делится на 2 страницы?

**Решение.**

1) Определяем объем памяти, необходимый для хранения 1 экранной страницы:

$1 \times 1024 / 2 = 512$  Кб

2) Определяем объем памяти 1 экранной страницы в битах  $512 \times 1024 \times 8 = 4194304$  бит

3) Определяем количество пикселей на экране:  $800 \times 600 = 240000$  пикселей

4) Определяем количество бит видеопамати, приходящихся на 1 пиксель:

$4194304 / 240000 = 17,48$  бит

5) Определяем по таблице битовую глубину, ближайшее и меньшее 17,48 значение b. b=16

6) Определяем по таблице количество цветов N для b=16. N=65536 цветов.

## Практическое занятие № 7.

### Дисктерное (цифровое) представление звуковой и видеоинформации.

#### Решение задач на определение скорости передачи данных.

**Цель:** научиться рассчитывать информационный объем звукового и видеофайла, скорость передачи данных.

#### Кодирование звуковой информации

С начала 90-х годов персональные компьютеры получили возможность работать со звуковой информацией. Каждый компьютер, имеющий звуковую плату, микрофон и колонки, может записывать, сохранять и воспроизводить звуковую информацию.

Звук представляет собой звуковую волну с непрерывно меняющейся амплитудой и частотой. Чем больше амплитуда, тем он громче для человека, чем больше частота сигнала, тем выше тон. Программное обеспечение компьютера в настоящее время позволяет непрерывный звуковой сигнал преобразовывать в последовательность электрических импульсов, которые можно представить в двоичной форме.

#### Процесс преобразования звуковых волн в двоичный код в памяти компьютера:

Звуковая волна → МИКРОФОН → переменный электрический ток →  
→ АУДИОАДАПТЕР → двоичный код → ПАМЯТЬ ЭВМ

#### Процесс воспроизведения звуковой информации, сохраненной в памяти ЭВМ:

Аудиоадаптер (звуковая плата) – специальное устройство, подключаемое к компьютеру, предназначенное для преобразования электрических колебаний звуковой частоты в числовой двоичный код при вводе звука и для обратного преобразования (из числового кода в электрические колебания) при воспроизведении звука.

В процессе записи звука аудиоадаптер с определенным периодом измеряет амплитуду электрического тока и заносит в регистр двоичный код полученной величины. Затем полученный код из регистра переписывается в оперативную память компьютера. Качество компьютерного звука определяется характеристиками аудиоадаптера: частотой дискретизации и разрядностью.

Частота дискретизации – это количество измерений входного сигнала за 1 секунду. Частота измеряется в герцах (Гц). Одно измерение за одну секунду соответствует частоте 1 Гц. 1000 измерений за 1 секунду – 1 кГц. Характерные частоты дискретизации аудиоадаптеров: 11 кГц, 22 кГц, 44,1 кГц и др.

Разрядность регистра – число бит в регистре аудиоадаптера. Разрядность определяет точность измерения входного сигнала. Чем больше разрядность, тем меньше погрешность каждого отдельного преобразования величины электрического сигнала в число и обратно. Если разрядность равна 8 (16), то при измерении входного сигнала может быть получено  $2^8 = 256$  ( $2^{16} = 65536$ ) различных значений. Очевидно, 16-разрядный аудиоадаптер точнее кодирует и воспроизводит звук, чем 8-разрядный.

Звуковой файл - файл, хранящий звуковую информацию в числовой двоичной форме.

#### Решаем задачи

- Определить информационный объем стерео аудио файла длительностью звучания 1 секунда при частоте дискретизации 48 кГц и разрешении 16 битов.
- Определить информационный объем цифрового моно аудио файла длительностью звучания которого составляет 10 секунда при частоте дискретизации 96 кГц и разрешении 8 битов.
- Определить объем памяти для хранения цифрового стерео аудио файла, время звучания которого составляет две минуты при частоте дискретизации 48 кГц и разрешении 16 битов.



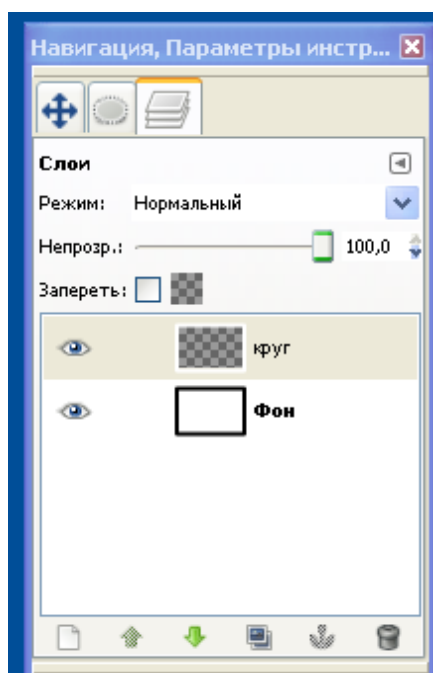
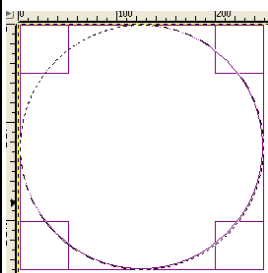
- В распоряжении пользователя имеется память объемом 2,6 Мб. Необходимо записать цифровой аудиофайл длительностью звучания 1 минута. Какой должна быть частота дискретизации и разрядность?
- Объем свободной памяти на диске — 10,5 Мб, разрядность звуковой платы — 16 бит. Какова длительность звучания цифрового моно аудиофайла, записанного с частотой дискретизации 48 кГц?
- Одна минута записи цифрового стерео аудиофайла занимает на диске 1,3 Мб, разрядность звуковой платы - 8 бит. С какой частотой дискретизации записан звук?
- Две минуты записи цифрового моно аудиофайла занимают на диске 5,05 Мб. Частота дискретизации — 48 кГц. Какова разрядность аудиоадаптера?
- Объем свободной памяти на диске — 0,1 Гб, разрядность звуковой платы — 16. Какова длительность звучания цифрового аудиофайла, записанного с частотой дискретизации 44 100 Гц?
- Скорость передачи данных по локальной сети 16 Мбит в секунду. Каков объем информации был передан за 20 минут?
- Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 640x480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?
- Скорость передачи данных через модемное соединение равна 51 200 бит/с. Передача текстового файла через это соединение заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.
- Модем передает данные со скоростью 7680 бит/с. Передача текстового файла заняла 1,5 мин. Определите, сколько страниц содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode, а на одной странице – 400 символов.
- Передача данных через ADSL-соединение заняла 2 минуты. За это время был передан файл размером 3 750 Кбайт. Определите минимальную скорость (бит/с), при которой такая передача возможна.
- Сколько секунд потребуется модему, передающему сообщения со скоростью 32000 бит/с, чтобы передать 16-цветное растровое изображение размером 800 x 600 пикселей, при условии, что в каждом байте закодировано максимально возможное число пикселей?
- Скорость передачи данных через ADSL-соединение равна 512000 бит/с. Через данное соединение передают файл размером 1500 Кб. Определите время передачи файла в секундах.
- Скорость передачи данных через ADSL-соединение равна 1024000 бит/с. Через данное соединение передают файл размером 2500 Кб. Определите время передачи файла в секундах.
- Скорость передачи данных через ADSL-соединение равна 1024000 бит/с. Передача файла через данное соединение заняла 5 секунд. Определите размер файла в килобайтах.
- Скорость передачи данных через ADSL-соединение равна 512000 бит/с. Передача файла через данное соединение заняла 8 секунд. Определите размер файла в килобайтах.

## Практическое занятие № 8.

### Создание и редактирование графических объектов средствами программ компьютерной графики.

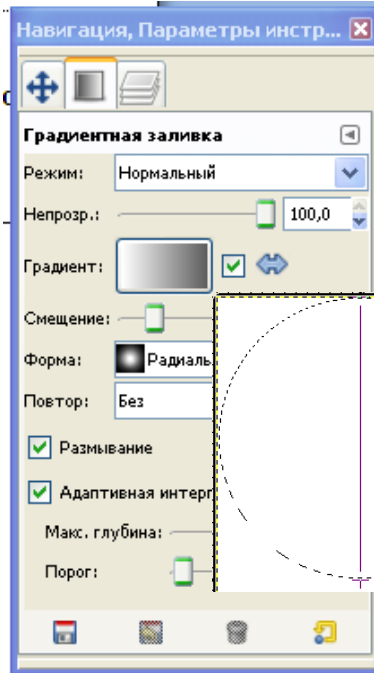
**Цель:** научиться создавать и редактировать графических объектов, научиться создавать слои, настраивать и изменять панели инструментов, строить простые графические рисунки.

Создайте новое изображение с белым цветом фона, задайте разрешение 600x400. Создайте новый слой и задайте ему название «круг». Возьмите инструмент **Эллиптическое выделение** и создайте круг диаметром 250px.



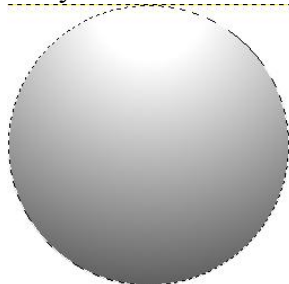
Установите серый цвет переднего плана (515151), а фона – белый. Возьмите инструмент **Градиент**, в его настройках разверните градиент стрелочкой, укажите радиальную форму для

форму для

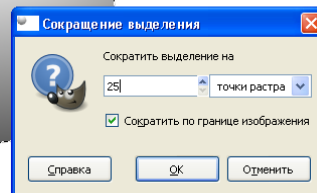
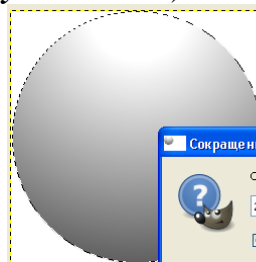


градиента, а смещение равно 20px. Создайте градиент сверху вниз, как на рисунке ниже.

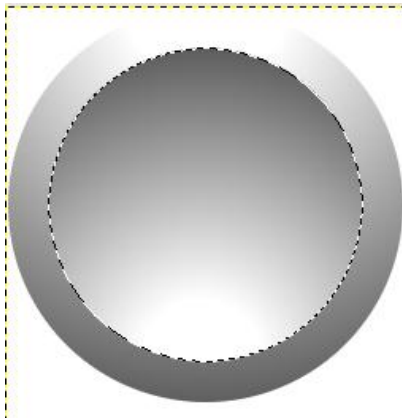
Получиться должно так:



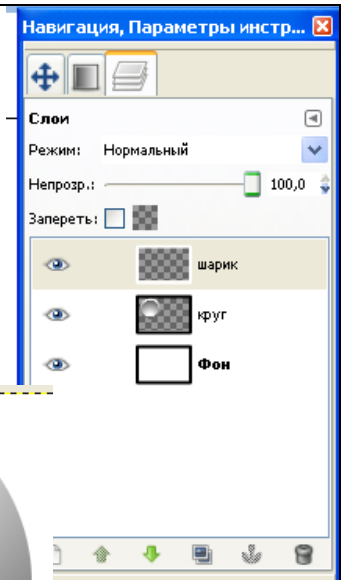
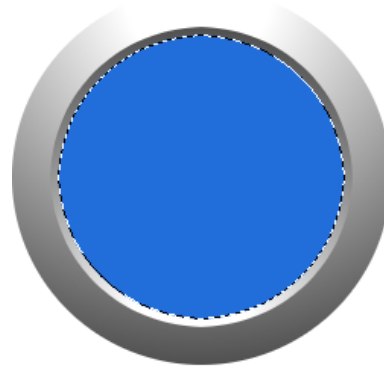
Уменьшите выделение на 25px (**Выделение-уменьшить**).



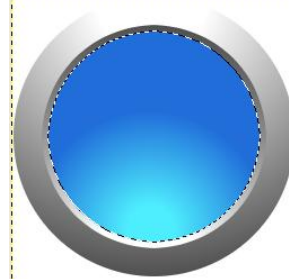
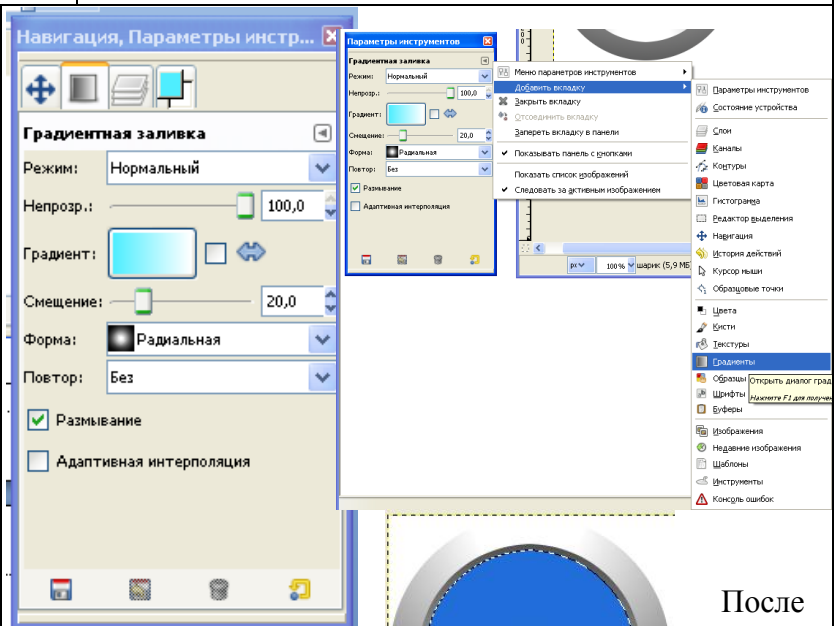
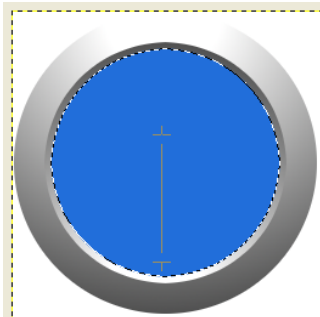
Примените прошлый градиент в обратном направлении (снизу вверх).



Создайте новый слой и назовите его «шарик». Уменьшите выделение еще на 5px, затем заполните выделенное цветом (21beda) инструментом **Плоская заливка (область применения – все выделение)**



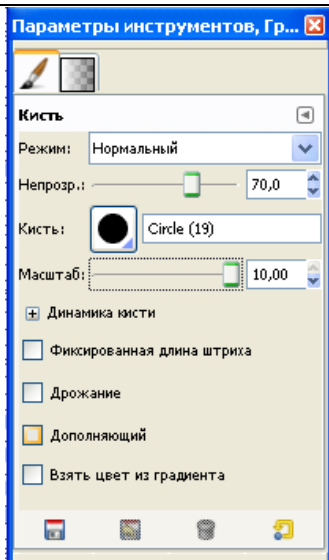
Установите цвет переднего плана (4feeff). Возьмите инструмент **Градиент** и в его вкладке задайте градиент «**Основной в прозрачный**», тип градиента: «радиальный», снимите галочку с разворота градиента, а адаптивную интерполяцию отключите. Примените градиент. Начинаться этот градиент должен немного выше низа выделения, а его длина должна быть примерно 120px.



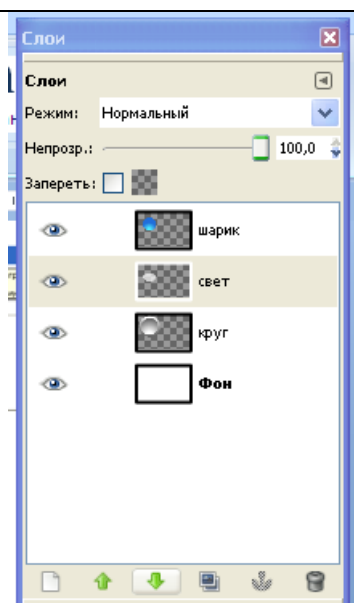
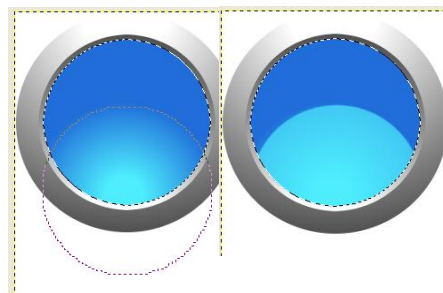
После

применения должно получиться так:

Создайте новый слой «свет». Теперь мы будем использовать **КИСТЬ** (двойной щелчок), чтобы добавить белого цвета снизу. Установите белый цвет переднего плана и возьмите большую кисть с мягкими краями Circle(19), установите непрозрачность кисти 70 %, а масштаб – 10.



Расположите кисть как раньше ниже и кликните мышью один раз.

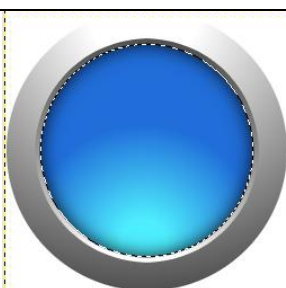
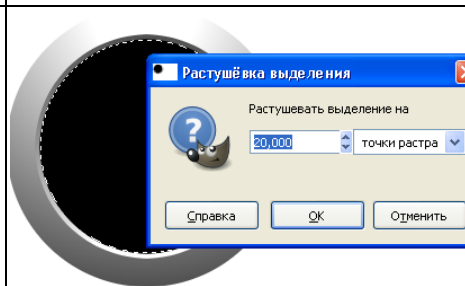


Сместите «свет» под «шарик».



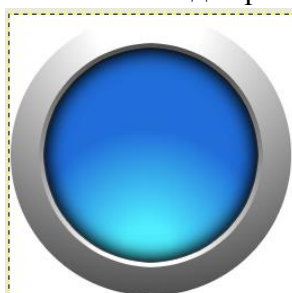
Получился мягкий светлый цвет.

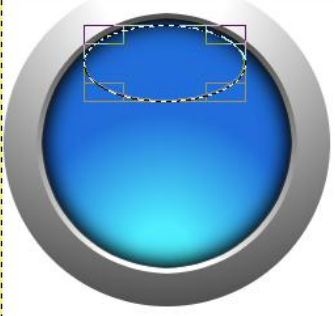

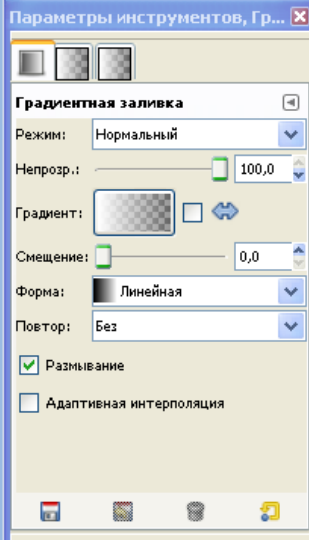

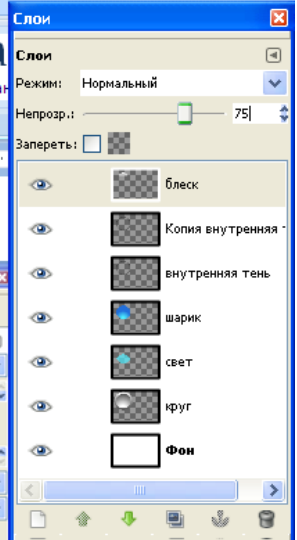

Создадим внутреннюю тень, для этого сделайте новый слой и назовите его «внутренняя тень». Увеличьте область выделения на 1 px. Заполните выделение цветом (000000), используя инструмент «Плоская заливка».



Нажмите Del. Снимите выделение (*Выделение снять*)

Продублируйте слой (*Слой-Создать копию слоя*) и тень станет в два раза четче.



 <p>Создайте новый слой «блеск». Нужно создать</p> <p>Эллиптическое выделение в том месте, где будет блик.</p>	<p>Установите белый цвет переднего плана и возьмите инструмент <b>Градиент</b>. Градиент должен быть «<i>Основной в прозрачный</i>», смещение 0, линейной формы. Примените градиент сверху вниз.</p>  
 <p>Понижьте прозрачность слоя до 75%, чтобы блик не был таким «жестким».</p> 	<p>Добавьте текст на кнопку с помощью инструмента Текст. После добавления текста следует изменить</p>  <p>непрозрачность слоя с текстом до 90%.</p>

## Практическое занятие № 9.

### Правила ввода и редактирования текста. Правила форматирования текста. Правила создания, редактирования списков и таблиц. Связь и внедрение объектов.

**Цель:** изучение информационной технологии набора текста, а также его форматирования, добавления в текст различных символов, оформления и заливки текста в текстовом редакторе MS Word, изучение информационной технологии создания и форматирования абзацев текста в текстовом редакторе MS Word, изучение информационной технологии создания и форматирования таблиц и списков в текстовом редакторе MS Word.

## Ход работы:

### 1. Подготовка к созданию текстового документа.

Загрузите программу MS Word

Изучите кнопки вкладок, подводя к ним курсор.

### 2. Набор текста.

- Наберите два абзаца текста по приведенному образцу, расположенному ниже.
- Установите вид шрифта - Times New Roman, размер шрифта - 14, начертание - курсив).

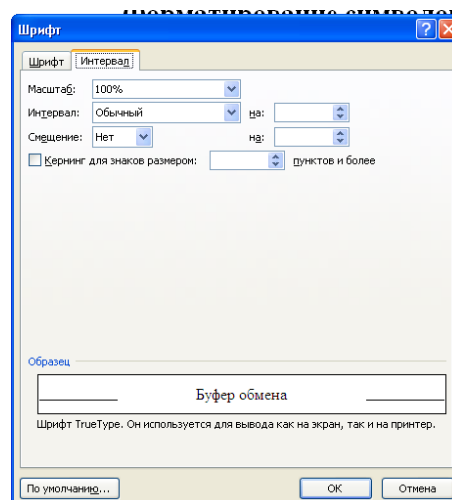
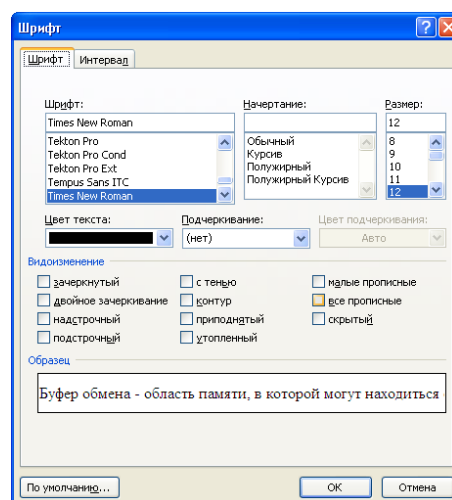
#### Образец текста для набора:

**Буфер обмена** - область памяти, в которой могут находиться объекты при проведении операций обмена между приложениями.

Работа с этой областью сводится к выполнению трех операций – **копирование, вырезание и вставка.**

### 3 Форматирование шрифта.

- Установите в первом абзаце напечатанного текста различные размеры шрифта: первое слово – 24 пт.; второе слово – 18 пт.; третье слово - 12 пт.; четвертое слово – 10 пт.; пятое слово – 8 пт.
- Оформите во втором абзаце в первой строке каждые два слова разным цветом.
- Произведите во втором абзаце следующие преобразования: первые два слова оформите полужирным шрифтом; вторые два слова – курсивом; третьи два слова – подчеркиванием; следующие два слова – курсивом + полужирным шрифтом + подчеркиванием.
- Задайте в первом абзаце разные виды подчеркивания: первое слово - с одинарным подчеркиванием; второе слово - с пунктирным подчеркиванием; второе слово - с двойным подчеркиванием.
- Наберите слово «эффект». Скопируйте его пять раз и наложите следующие видоизменения: первое слово "эффект" - видоизменение "зачеркнутый"; второе слово "эффект" (все буквы слова, кроме первой) - верхний индекс; третье слово "эффект" (все буквы слова, кроме первой) - нижний индекс; четвертое слово "эффект" - малые прописные; пятое слово "ЭФФЕКТ" - все прописные, полужирный, контур. Если вы все сделали правильно, то на экране вы увидите слова "эффект" в следующем варианте:
- На слова «Буфер обмена» установите интервал разряжения на 10 пт.
- Сохраните документ с именем **Буфер обмена.**



При наборе текста Word **автоматически** переносит текст на следующую строку. **Не совершайте типичную ошибку неопытных пользователей** – не нажимайте клавишу **Enter** для перехода на новую строку, так как нажатие **Enter** в Word обозначает **конец одного абзаца** и переход к началу другого. Конец абзаца **всегда** обозначается



специальным символом – ¶ – *маркером конца абзаца*, в котором также содержится информация о форматах абзаца. Если Вы удаляете маркер абзаца, то удаляется и форматирование, а текст в этом абзаце получит форматирование следующего за ним абзаца.

Маркер конца абзаца относится к непечатаемым символам, т.е. при распечатывании документа на принтере этот символ не отображается. На экране можно сделать видимыми все непечатаемые символы, в том числе маркеры абзацев и пробелы, если нажать кнопку **Непечатаемые символы** ¶ на стандартной панели инструментов.

Таким образом, **Абзац** – это любая часть документа, введенная от одного нажатия клавиши **Enter** до другого. Абзац может содержать текст, графику, объекты (например, формулы и диаграммы) или другие элементы. Конец абзаца обозначается маркером абзаца.

*Разные абзацы текста могут быть по-разному отформатированы.*

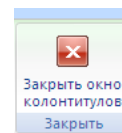
### 1. Форматирование абзацев текста.

- Загрузите программу MS Word
- Наберите один абзац текста по образцу.
- Установите параметры шрифта: вид шрифта – Times New Roman, размер шрифта – 14, начертание – обычный.

### Образец текста для набора:

Микропроцессор – основная микросхема ПК. Все вычисления выполняются в ней. Процессор аппаратно реализуется на большой интегральной схеме (БИС). Большая интегральная схема на самом деле не является большой по размеру и представляет собой, наоборот, маленькую плоскую полупроводниковую пластину размером примерно 20x20 мм, заключенную в плоский корпус с рядами металлических штырьков (контактов). БИС является большой по количеству элементов.

Использование современных высоких технологий позволяет разместить на БИС процессора огромное количество функциональных элементов, размеры которых составляют всего около 0,13 микрон (1 микрон =  $10^{-6}$  м). Например, в процессоре Pentium 4 их около 42 миллионов.

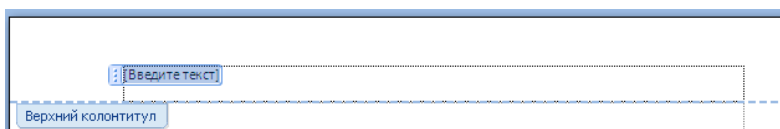


- Скопируйте набранный абзац текста пять раз.
- Выделите первый абзац текста и установите следующие параметры абзаца: первая строка - отступ стандартный; междустрочный интервал – полуторный; выравнивание – по ширине.
- Выделите второй абзац текста и установите следующие параметры абзаца: первая строка – 0,5 см; междустрочный интервал – одинарный; выравнивание – по правому краю.

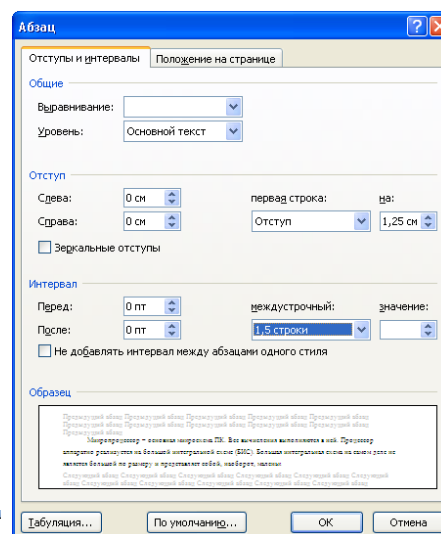
### 2. Задание междуабзацных расстояний.

Выделите весь текст командой и задайте междуабзацное расстояние 24 пт.

### 3. Задание колонтитулов.



введите следующую информацию: в верхний колонтитул – Ф.И.О., дата, время; в нижний колонтитул – Название учебного заведения и номера страниц.



Задайте колонтитулы документа. В колонтитулы

Обратите внимание, что при вводе колонтитулов основной текст приобрел бледный цвет и стал недоступен. Окончание работы с колонтитулами производится кнопкой *Заккрыть* панели *Колонтитулы*.

4. Сохраните набранный документ в свою папку с именем *Процессор*.

**Примечание.** Во время ввода текста Word может автоматически выполнять орфографическую проверку. Каждое слово, отсутствующее в словаре, воспринимается как ошибочное и подчеркивается волнистой красной линией. Чтобы увидеть список возможных правильных слов для исправления, установите указатель мыши на волнистую красную линию, нажмите правую кнопку мыши, а затем выберите правильное слово.

#### **Задание:**

1. Скопируй из Обмена папку «Жесткий диск» в папку Мои документы.
2. Открой файл «Винчестер» и прочитай внимательно текст.
3. Проверь орфографию.
4. Установи параметры страницы: левое поле – 2 см, остальные по 1,5 см.
5. Заголовок: выравнивание по центру, гарнитура Arial, начертание полужирное, размер 14, цвет темно-синий (50%).
6. Интервал между абзацами - 1,5 строки.
7. Абзацы: красная строка 0,5 см, по ширине, Times New Roman, обычное, 12, синий (25%).
8. Картинки из папки «Жесткий диск» расположи по контексту, применяя *Формат - Обтекание текстом – по контуру*.
9. Создай для картинок рамки [цвет – синий (50%), 1,5 пт]
10. Проставь нумерацию страниц.
11. В последнем абзаце найди фрагмент текста, в котором нужно создать список. Создай маркированный список, состоящий из трех абзацев.
12. Выдели абзац, в котором говорится, что такое фрагментация и дефрагментация диска, полужирным курсивом.

#### **Винчестер.**

Накопитель на жёстких магнитных дисках, жёсткий диск, хард, HDD или винчестер, (англ. Hard Magnetic Disk Drive, HMDD) — энергонезависимое, перезаписываемое компьютерное запоминающее устройство.

Является основным накопителем данных практически во всех современных компьютерах. В отличие от «гибкого» диска (дискеты), информация в НЖМД записывается на жёсткие (алюминиевые или стеклянные) пластины, покрытые слоем ферромагнитного материала, чаще всего двуокиси хрома. Считывающие головки в рабочем режиме не касаются поверхности пластин благодаря прослойке набегающего потока воздуха, образуемого у поверхности при быстром вращении. Расстояние между головкой и диском составляет несколько нанометров (в современных дисках 5-10 нм), а отсутствие механического контакта обеспечивает долгий срок службы жёсткого диска. При отсутствии вращения дисков, головки находятся у шпинделя в безопасной зоне, где исключен их контакт с поверхностью дисков.

Название «винчестер» жёсткий диск получил благодаря фирме IBM, которая в 1973 году выпустила жёсткий диск, впервые объединивший в одном неразъёмном корпусе пластины диска и считывающие головки. При его разработке инженеры использовали краткое внутреннее название «30-30», что означало два модуля (в максимальной компоновке) по 30 Мб каждый. Кеннет Хотон, руководитель проекта, по созвучию с обозначением популярного охотничьего ружья «Winchester 30-30» предложил назвать этот диск «винчестером».



В Европе и США название «винчестер» вышло из употребления в 1990-х годах, в российском же компьютерном сленге название «винчестер» сохранилось, сократившись до слова «винт».

Как правило, подобные накопители устроены так, как показано на рис. 1 и 2. Диск вращается, с помощью магнитной головки происходит запись информации или ее считывание.

Информация записывается на диск порциями (кластерами) в виде концентрических колец (треков/дорожек).

Для того чтобы записать на диск порцию информации, требуется время: магнитная головка должна передвинуться на дорожку, где есть свободное место, а свободное место на дорожке — передвинуться (прокрутиться) и оказаться под магнитной головкой.

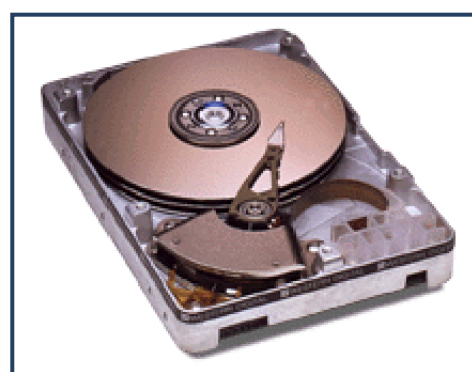
Чтобы считать с диска порцию информации, тоже требуется время: магнитная головка должна передвинуться на дорожку, где находится нужная информация, а место на дорожке с нужной информацией — передвинуться и оказаться под магнитной головкой.

Пока диск пустой, запись информации идет с самой высокой скоростью. После записи первой дорожки (кластеры 1, 2, 3 и т.д.) головка перемещается на соседнюю дорожку и процедура повторяется. Потери времени при этом минимальны.

С ростом количества и размера файлов, хранимых на жестком диске, увеличивается степень фрагментации диска. Фрагментация дисков - это такое состояние, когда файлы оказываются разбитыми на части, распределенные по всему жесткому диску. Дефрагментация диска - операция, состоящая в эффективном размещении физических структур файлов. При дефрагментации кластеры

каждого файла размещаются в диске, что сокращает время файлу.

Во всех современных моделях обязательно есть поддержка S.M.A.R.T. Что это за S.M.A.R.T. - это Self Monitoring Reporting Technology, что технология самопроверки, анализа



одном месте на обращении к

жестких дисков технологии аббревиатура? Analysis and переводится как и отчета. В принципе из названия четко видно, для чего предназначена эта технология.

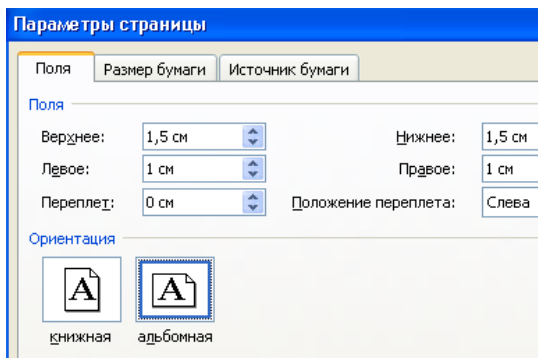
Принцип же работы S.M.A.R.T. заключается в следующем. В процессе работы жесткого диска регулярно измеряется ряд параметров, которые могут оказывать влияние на его состояние и работоспособность (количество bad-секторов, высота полета головки над поверхностью "блина" и т. п.). При этом помимо цифрового обозначения каждой характеристики, которое немного говорит простому пользователю, есть и более понятные градации.

Их всего три. Первая обозначает, что значение данного параметра равно начальному и не изменялось. Вторая свидетельствует о постепенном ухудшении характеристики. Третья

же сигнализирует о критическом состоянии жесткого диска, рекомендуя пользователю сейчас же перейти на работу с другим винчестером.

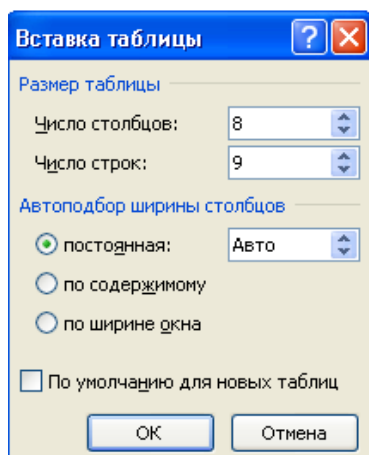
### Задание.





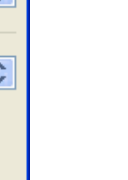
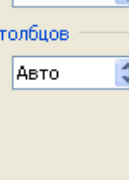
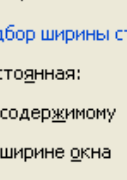
- Установить поля страницы: верхнее и нижнее 1,5 см, левое и правое 1 см, ориентация альбомная.



– альбомная.

- Вставьте таблицу 8 x 9 согласно образцу. Заполните ее данными. Картинки взять из Обмена.



	Россия	Великобритания	США	Австралия и Океания	Франция	Германия	Япония
Часть света	Европа	Америка	Австралия и Океания	Азия	Европа	Азия	Азия
Столица	Москва	Вашингтон	Канберра	Токио	Париж	Берлин	Токио
Государственный язык	русский	английский		японский	французский	немецкий	японский
Флаг							
Численность населения	141,9 млн. чел.	304,2 млн. чел.	20,5 млн. чел.	63,2 млн. чел.	82,6 млн. чел.	127,6 млн. чел.	127,6 млн. чел.
Национальная валюта	рубли	доллары	австралийские доллары	французские франки	немецкие марки	японские йены	японские йены
Площадь	17 075,5 тыс. км <sup>2</sup>	9 400 тыс. км <sup>2</sup>	8 511 тыс. км <sup>2</sup>	637,7 тыс. км <sup>2</sup>	357 тыс. км <sup>2</sup>	377,9 тыс. км <sup>2</sup>	377,9 тыс. км <sup>2</sup>
Плотность населения	8,7 чел/км <sup>2</sup>	31 чел/ км <sup>2</sup>	2,2 чел/км <sup>2</sup>	110 чел/км <sup>2</sup>	231 чел/км <sup>2</sup>	337 чел/км <sup>2</sup>	337 чел/км <sup>2</sup>

## Практическое занятие № 10.

### Моделирование интегрированных документов. Применение редактора формул и встроенного графического редактора в текстовом процессоре.

Цель: научиться моделировать интегрированные документы.

#### ХОД РАБОТЫ

1. Открыть программу Microsoft Word
2. Сохранить файл с именем «Формулы и рисование в документе.docx»
3. Вызвать команду Лента – Разметка страницы, Группа – Параметры страницы, Ориентация – Альбомная, ОК
4. Вызвать команду Лента – Вставка, Группа – Таблица, Число столбцов – 2, строк – 2
5. Выделить таблицу, Вызвать команду Лента – Главная, Группа Абзац, Кнопка – Границы и заливка – Нет границ, там же – Отобразить сетку. В результате останется только сетка (структура) таблицы.
6. Выделить две ячейки первой строки и Вызвать Контекстную ленту – Работа с таблицами, Вкладка – Макет, Кнопка – Объединить ячейки
7. Включить режим непечатаемых символов: Лента – Главная, Группа – Абзац, Кнопка – Отобразить все знаки. Выделить абзац в первой строке и установить след. параметры: Шрифт – Times new Roman размер 12 пт.

Ввести текст:

Объем шара	
Теорема	
	$\frac{4}{3}\pi R^3$
Объем шара радиуса R равен	

Для ввода выражения, содержащего формулу вызвать команду Лента – Вставка, Группа Текст, Кнопка – Объект – Объект – Microsoft Equation 3.0.

В документ Word внедрится окно редактора формул, снабженное панелью Формула. Панель Формула содержит шаблоны всех возможных формул, выражений и знаков. Достаточно правильно выбирать шаблоны и вписывать в них соответствующие символы. Например, для приведенной формулы понадобятся шаблоны дробей и радикалов и Шаблоны верхних и нижних индексов.

8. Во второй строке установить ширину первой ячейки: Контекстная лента – Работа с таблицами, Вкладка – Макет, Группа – Размер ячейки Ширина – 15см и ввести текст (Шрифт – Times new Roman размер 12 пт):

Доказательство
Рассмотрим шар радиуса R с центром в точке O и выберем ось OX произвольным образом (рис. 1). Сечение шара плоскостью, перпендикулярной к оси OX и проходящей через точку M этой оси, является кругом с центром в точке M. Введем обозначения: x – абсцисса точки M, r – радиус круга, полученного в сечении, S(x) – площадь этого круга, зависящая от x. Выразим S(x) через x и R.
$r = \sqrt{OC^2 - OM^2} = \sqrt{R^2 - x^2}$
Так как $S(x) = \pi r^2$ , то
$S(x) = \pi(R^2 - x^2)$
Заметим, что эта формула верна для любого положения точки M на диаметре AB, т.е. для всех x, удовлетворяющих условию $-R \leq x \leq R$ . Применяя основную формулу для вычисления объемов тел при $a=-R$ , $b=R$ , получим
$V = \int_{-R}^R \pi(R^2 - x^2) dx = \pi R^2 \int_{-R}^R dx - \pi \int_{-R}^R x^2 dx = \pi R^2 x \Big _{-R}^R - \pi \frac{x^3}{3} \Big _{-R}^R = \frac{4}{3} \pi R^3$
Теорема доказана. □

для указания пределов интегрирования использовать Шаблоны скобок и затем Шаблоны матриц

9. Форматировать введенный текст (сверять с образцом), установив след. параметры:
  - а. Везде: Интервал между абзацами (после-6пт)

b. Первая строка таблицы: Шрифт -Размер-14пт, Начертание – Полужирный, Интервал разреженный-2пт, Абзац: Выравнивание – влево; для пустого абзаца установить заливку Цвет – темно-синий (применить к абзацу)

c. Вторая строка:

Доказательство- Шрифт – Начертание – Полужирный;

Для латинских символов (Первый – R) Шрифт – Начертание – Полужирный курсив), Далее (для остальных лат симв.) использовать Лента главная – группа Буфер обмена – Кнопка Формат по образцу.

Для всех абзацев, содержащих формулы, Тип выравнивания абзаца – По центру;

Для всех абзацев, содержащих многострочный текст, Тип выравнивания – по ширине, Отступ первой строки– 2,5см.

Для всех остальных абзацев – без изменения

10. Выполнение чертежа

a. Включить Лента – Вставка, Группа – Иллюстрации, Кнопка – Фигуры – Основные фигуры

b. Используя Основные фигуры и Контекстную ленту Средства рисования, расположить на чертеже следующие объекты и сделать над ними преобразования:

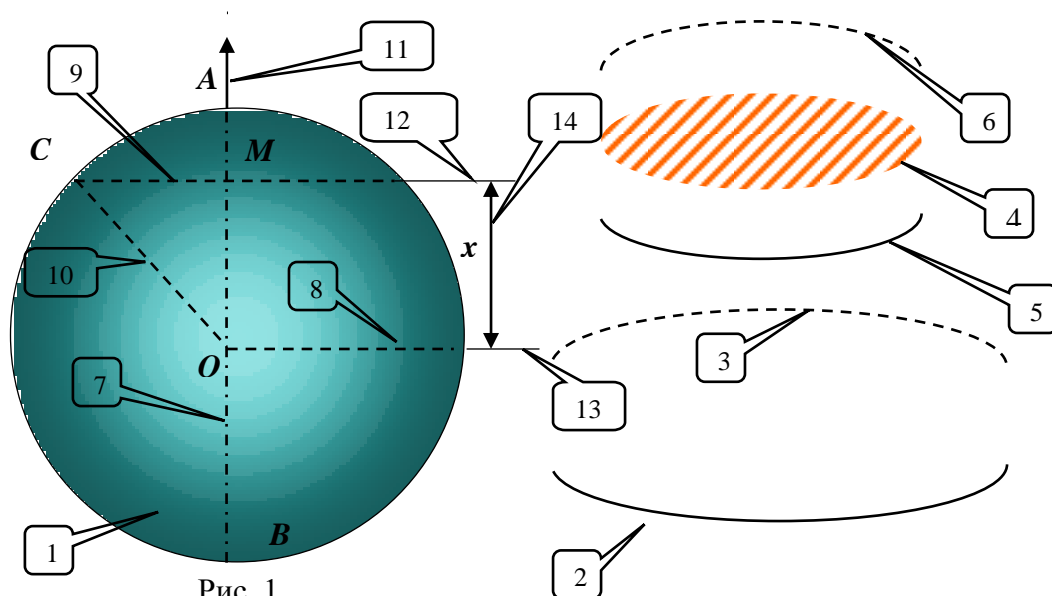





Рис. 1

№ объекта	Наименование объекта	Ширина(см)	Высота(см)	Цвет заливки – Способы заливки	Цвет линии	Тип линии (пт)	Штрих	Преобразование
1	Овал (+Shift)	6	6	Параметры смотри в конце страницы <sup>1</sup>	черный	0,5	сплошная линия	
2	Дуга	6	0,75	-	черный	1	сплошная линия	Копировать <sup>2</sup> , Вставить Повернуть- Отразить сверху-вниз, Поменять штрих (см <sup>3</sup> )
3	Дуга	6	0,75	-	черный	1	штрих	

<sup>1</sup> Заливка фигуры – Градиентная – Другие градиентные заливки – Заливка – Градиентная заливка – Градиентная, тип Путь, Точки градиента :

<sup>2</sup> Цвет Сине-зеленый – Другие цвета – Спектр Красный 51, Зеленый 204, Синий 204, ОК, Положение 0%, яркость 0%, прозрачность -50%

<sup>3</sup> Цвет Темно – Сизый – Другие цвета – Спектр Красный 24, Зеленый 94, Синий 94, ОК, Положение 100%, яркость 0%, прозрачность -0%

4	Овал	4,24	1,27	Параметры смотри в конце страницы <sup>2</sup>	нет линий	-	-	
5	Дуга	4,24	0,6	-	черный	1	сплошная линия	Копировать  , Вставить Повернуть- Отразить сверху-вниз, Поменять штрих (см  )
6	Дуга	4,24	0,6	-	черный	1	штрих	
7	Линия +(Shift)	0	6	-	черный	1	штрих-пунктир	
8	Линия +(Shift)	3	0	-	черный	1	штрих	
9	Линия +(Shift)	4,24	0	-	черный	1	штрих	
10	Линия	2,2	2,21	-	черный	1	штрих	
11	Стрелка+(Shift)	0	0,95	-	черный	1	сплошная линия	
12	Линия +(Shift)	1,59	0	-	черный	0,5	сплошная линия	
13	Линия +(Shift)	0,95	0	-	черный	0,5	сплошная линия	
14	Линия +(Shift)	0	2,22		черный	1	сплошная линия	Стрелка, стиль 

с. Для обозначения букв Лента – Вставка, Группа – Текст Кнопка Надпись Простая Надпись  
Удалить текст, ввести букву ,настроить параметры: Цвет Заливки-Нет заливки, Контур фигуры –Нет Контур, Шрифт- Начертание –Полужирный курсив.

d. Аналогично надпись под рисунком.

11. При нажатой клавише Shift последовательно выделить все объекты и нажать кнопку Лента Средства Рисования – Группа Упорядочить Кнопка Группировать. В результате рисунок, состоящий из 20 объектов станет как 1 объект. Его можно увеличить уменьшить, преобразовать.
12. Сохранить текущие изменения, нажав кнопку Сохранить.

<sup>2</sup> Заливка фигуры Цвета темы – *Оранжевый Акцент 6. 25%*, Заливка Градиентная – Другие градиентные заливки – Заливка – Узорная заливка, Узор-широкая диагональная 2

## Объем шара

### Теорема

Объем шара радиуса  $R$  равен  $\frac{4}{3}\pi R^3$ .

Доказательство

Рассмотрим шар радиуса  $R$  с центром в точке  $O$  и выберем ось  $OX$  произвольным образом (рис. 1). Сечение шара плоскостью, перпендикулярной к оси  $OX$  и проходящей через точку  $M$  этой оси, является кругом с центром в точке  $M$ . Введем обозначения:  $x$  – абсцисса точки  $M$ ,  $r$  – радиус круга, полученного в сечении,  $S(x)$  – площадь этого, круга, зависящая от  $x$ . Выразим  $S(x)$  через  $x$  и  $R$ .

$$r = \sqrt{OC^2 - OM^2} = \sqrt{R^2 - x^2}$$

Так как  $S(x) = \pi R^2$ , то

$$S(x) = \pi(R^2 - x^2)$$

Заметим, что эта формула верна для любого положения точки  $M$  на диаметре  $AB$ , т.е. для всех  $x$ , удовлетворяющих условию  $-R \leq x \leq R$ . Применяя основную формулу для вычисления объемов тел при  $a=-R$ ,  $b=R$ , получим

$$V = \int_{-R}^R \pi(R^2 - x^2) dx = \pi R^2 \int_{-R}^R dx - \pi \int_{-R}^R x^2 dx = \pi R^2 x \Big|_{-R}^R - \pi \frac{x^3}{3} \Big|_{-R}^R = \frac{4}{3} \pi R^3$$

Теорема доказана.

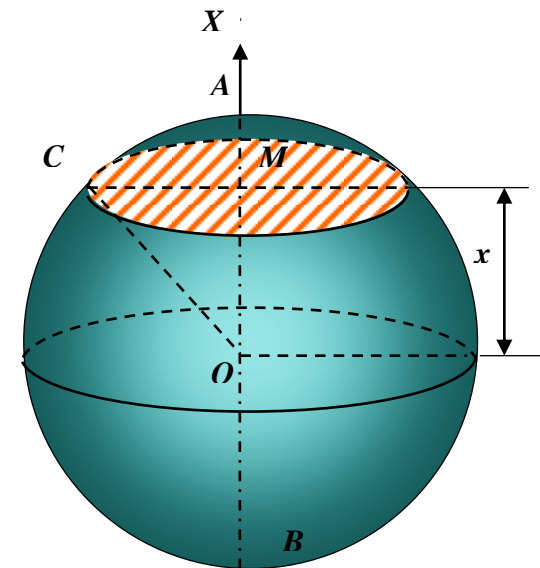


Рис. 1



## Практическое занятие № 11.

### Использование различных возможностей динамических (электронных) таблиц для выполнения учебных заданий из различных предметных областей.

**Цель:** научиться моделировать различные процессы в электронных таблицах.

#### ЗАДАЧА ОБОИ И КОМНАТА

В магазине продаются обои (см табл): Для удобства обслуживания надо составить таблицу, которая позволит определить необходимое количество рулонов для оклейки любой комнаты, имеющей размеры

Длина – *a*, ширина – *b*, высота потолка – *h*. Считается, что 15% площади стен занимают окна и двери. Как правило. 1 рулон приобретается под запас.

Для того, чтобы узнать количество обоев нужно:

1. Узнать площадь всех стен без дверей и окон:

$$S_{\text{стен}} = (a + b) \cdot h \cdot (100\% - 15\%)$$

Узнать площадь одного рулона

$$\text{обоев: } S_{\text{обоев}} = L \cdot d$$

2. Разделить площадь стен на площадь обоев, причем учесть, что количество не может быть дробным и 1 рулон про запас.

$$N = \left\lceil \frac{S_{\text{стен}}}{S_{\text{обоев}}} \right\rceil + 1$$

Наименование	Длина ( <i>L</i> )	Ширина ( <i>d</i> )
Erfurt01	10,05	0,53
SIRPI	10,05	0,7
Venecia	10,05	1,06
HandyArt	15	1,06
ERISMANN	25	1,06
OmexCo	6	0,7
Brasco Di Gala	6	0,9

#### Ход работы

Открыть электронную таблицу. Назвать лист Обои и комната

Заполнить по образцу расчетную таблицу. Ввести формулы и сравнить с результатом:

#### ФОРМУЛЫ

	A	B	C	D	E
1	Исходные данные				
2	<b>Комната</b>				
3	Высота h	2,6			
4	Длина a	5			
5	Ширина b	3			
6	Неклеиваемая поверхность	0,15			
7	Площадь стен S_стен	=(B4+B5)*B3*(100%-B6)			
8	<b>Обои</b>				
9	Наименование	Длина L	Ширина d	Площадь рулона S_обоев	Количество рулонов
10	Erfurt01	10,05	0,53	=B10*C10	=ЦЕЛОЕ(\$B\$7/D10)
11	SIRPI	10,05	0,7		
12	Venecia	10,05	1,06		
13	OmexCo	15	1,06		
14	ERISMANN	25	1,06		
15	HandyArt	6	0,7		
16	Brasco Di Gala	6	0,9		

#### РЕЗУЛЬТАТ

Исходные данные					
<b>Комната</b>					
3	Высота h	2,6			
4	Длина a	5			
5	Ширина b	3			
6	Неклеиваемая поверхность	15%			
7	Площадь стен S_стен	17,68			
<b>Обои</b>					
	Наименование	Длина L	Ширина d	Площадь рулона S_обоев	Количество рулонов N
10	Erfurt01	10,05	0,53	5,3265	4
11	SIRPI	10,05	0,7	7,035	3
12	Venecia	10,05	1,06	10,653	2
13	OmexCo	15	1,06	15,9	2
14	ERISMANN	25	1,06	26,5	1
15	HandyArt	6	0,7	4,2	5
16	Brasco Di Gala	6	0,9	5,4	4

Провести эксперимент, рассчитав количество рулонов для комнат Вашей квартиры (поочередно ввести размеры кухни, спальни, гостиной, и т. д.). Скопировать табличку расчетов (Выделить обе таблички, Копировать)

Открыть программу Microsoft Word

Написать: Кухня, вставить результаты (Вставить, Параметры вставки- Рисунок) и т. д.  
Сохранить документы в таблице и в текстовом редакторе.

### ЗАДАЧА О БИОРИТМАХ.

Существует гипотеза, что жизнь человека подчиняется трем циклическим процессам, называемым биоритмами. Биоритмы характеризуют три стороны самочувствия человека: физическую, эмоциональную и интеллектуальную. Изменяются биоритмы синусоидально. Считается, что дни, когда график биоритма возрастает – благоприятные и наоборот. Дни, в которых график проходит через ось абсцисс- неблагоприятные. За точку отсчета берется день рождения человека. Переменная  $t$  показывает сколько дней прожил человек

Физический биоритм- это самочувствие человека. Его период – 23 дня  $\Phi = \sin\left(\frac{2\pi t}{23}\right)$

Эмоциональный биоритм – эмоциональное восприятие мира Период – 28 дней

$$\Xi = \sin\left(\frac{2\pi t}{28}\right)$$

Интеллектуальный биоритм- мыслительные способности. Период – 33

$$\text{дня } I = \sin\left(\frac{2\pi t}{33}\right)$$

Построить график биоритмов конкретного человека на 20 дней вперед и проанализировать результат.

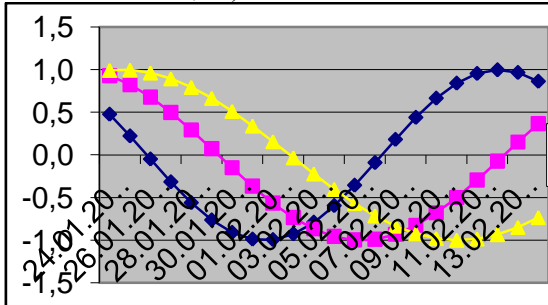
Назвать лист Биоритмы и заполнить по образцу расчетную таблицу.

Порядковый день	Ф	Э	И
=B3	=SIN(2*ПИ()* (A7-\$B\$2)/23)	=SIN(2*ПИ()* (A7-\$B\$2)/28)	=SIN(2*ПИ()* (A7-\$B\$2)/33)
=A7+1			

Порядковый день	Ф	Э	И
24.01.2011 16:09	0.5	0.9	1.0
25.01.2011 16:09	0.2	0.8	1.0
26.01.2011 16:09	0.0	0.7	1.0
27.01.2011 16:09	-0.3	0.5	0.9
28.01.2011 16:09	-0.6	0.3	0.8
29.01.2011 16:09	-0.8	0.1	0.7
30.01.2011 16:09	-0.9	-0.2	0.5
31.01.2011 16:09	-1.0	-0.4	0.3
01.02.2011 16:09	-1.0	-0.6	0.2
02.02.2011 16:09	-0.9	-0.7	0.0
03.02.2011 16:09	-0.8	-0.9	-0.2
04.02.2011 16:09	-0.6	-1.0	-0.4
05.02.2011 16:09	-0.4	-1.0	-0.6
06.02.2011 16:09	-0.1	-1.0	-0.7
07.02.2011 16:09	0.2	-0.9	-0.8
08.02.2011 16:09	0.4	-0.8	-0.9
09.02.2011 16:09	0.7	-0.7	-1.0
10.02.2011 16:09	0.8	-0.5	-1.0
11.02.2011 16:09	1.0	-0.3	-1.0
12.02.2011 16:09	1.0	-0.1	-0.9
13.02.2011 16:09	1.0	0.2	-0.8
14.02.2011 16:09	0.9	0.4	-0.7



Построить график для всех трех биоритмов (Выделить диапазон A7:D28, Лента Вставка, Группа Диаграмма, График, График с маркерами, ОК; В Контекстной ленте Конструктор Нажать на кнопку Выбрать данные, Справа: Подписи горизонтальной оси - Изменить, выделить мышью диапазон A7:A28, ОК; В том же окне слева: Элементы легенды выделить Ряд1, нажать Изменить, Ввести букву Ф, ОК, аналогично Э, И)



Провести эксперимент для реального человека Составить отчет в текстовом редакторе. В отчет должны войти:  
 — Фамилия, Имя, Отчество, график для трех биоритмов (выделить график в электронной таблице, перейти в Word, Вставить Параметр вставки-Рисунок) и таблица в виде:

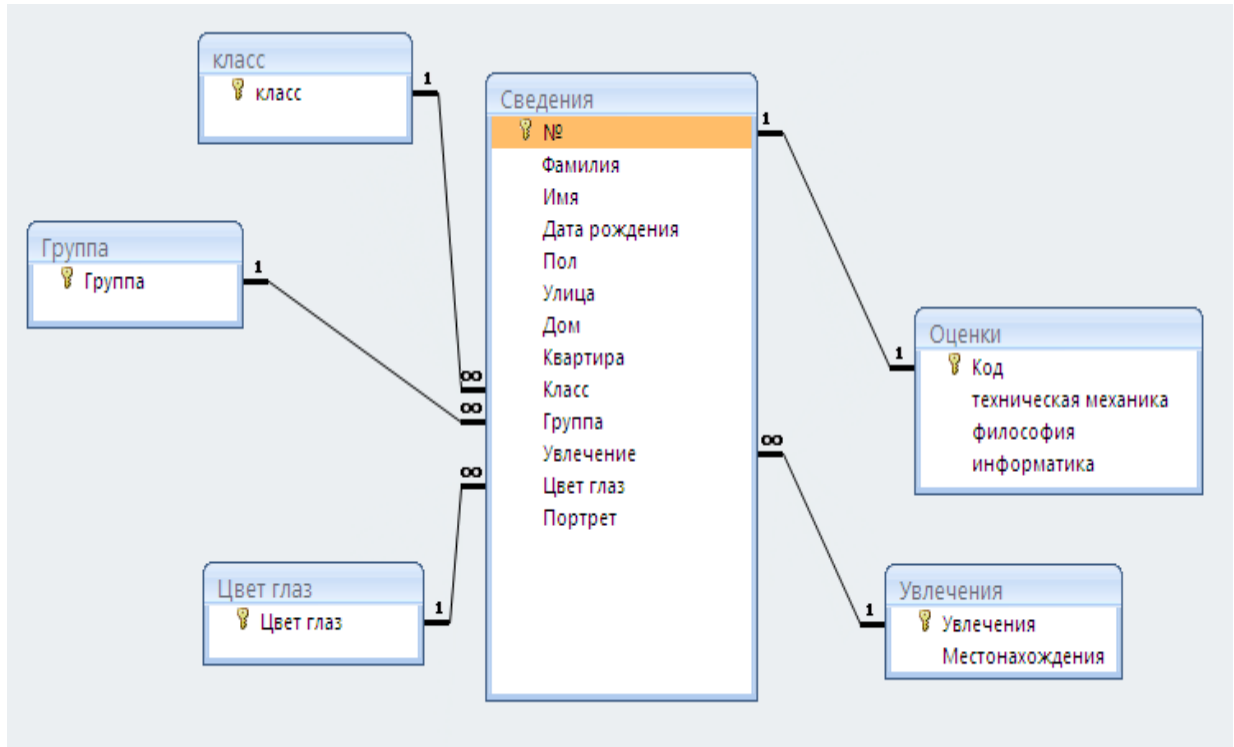
Дни, благоприятные для похода в театр	
Дни, благоприятные для туристического похода	
Дни, благоприятные для сдачи экзамена	

## Практическое занятие № 12.

### Организация работы с данными в БД. Формирование запросов.

Цель: научиться моделировать базы данных, формировать запросы.

#### Ход работы:



#### Ход работы:

1. Пуск – Программы– Microsoft Office– Microsoft Access
2. Файл – Создать – Новая база данных. Присвоить имя файлу- Сведения о школьниках, ОК. Появится окно базы данных Сведения о школьниках.  
Создание структуры базы данных.
3. Создание структуры таблицы Группа.  
Вызвать команду: Лента – Создание, Кнопка – Конструктор таблиц. Появится Конструктор новой таблицы. Ввести Имя поля- Группа, Тип данных – Текстовый. Сделать поле Группа ключевым (нажать на кнопку Ключевое поле); Щелкнуть по ярлычку таблицы, Сохранить Таблицу с именем Группа, Закрыть.
  - 3.1. Создание структуры таблицы Класс.  
Лента - Создание, Кнопка – Конструктор таблиц. Ввести Имя поля- Класс, Тип данных – Числовой (длинное целое). Сделать поле Класс ключевым; сохранить Таблицу с именем Класс, Закрыть.
  - 3.2. Создание структуры таблицы Цвет глаз.  
Лента - Создание, Кнопка – Конструктор таблиц. Ввести Имя поля- Цвет\_глаз, Тип данных – текстовый. Сделать поле Цвет\_глаз ключевым; сохранить Таблицу с именем Цвет глаз, Закрыть.
  - 3.3. Создание структуры таблицы Увлечения.  
Лента - Создание, Кнопка – Конструктор таблиц. Ввести Имя поля- Увлечение, Тип данных – текстовый. Следующее поле – Местонахождение. Тип данных – текстовый. Сделать поле Увлечение ключевым; сохранить Таблицу с именем Увлечения, Закрыть.

### 3.4. Создание структуры таблицы Оценки

Лента - Создание, Кнопка – Конструктор таблиц. Ввести Имя поля- Код, Тип данных – числовой. Следующее поле – Тех\_механика, Философия, Информатика Для всех тип данных -Числовой. Сделать поле Код ключевым; сохранить Таблицу с именем Оценки

### 3.5. Создание структуры таблицы Сведения

Лента - Создание, Кнопка – Конструктор таблиц. Ввести Имя поля- №, Тип данных – числовой. Следующие поля –Фамилия, Имя – текстовый, Дата рождения – Дата/время, Пол , Улица – текстовый, Дом, Квартира, Класс – числовой, {Группа, Увлечение, Цвет глаз} – текстовый, Портрет- Поле объекта OLE.

3.6. Выделить строку, содержащую поле Класс, и в свойствах поля сделать следующие установки: (см рис.2)

3.7. Аналогичные свойства поля (Подстановка) создать для полей Группа, Увлечение, Цвет глаз.

3.8. Сделать поле № ключевым; сохранить Таблицу с именем Сведения.

Имя поля	Тип данных
Имя	Текстовый
Дата рождения	Текстовый
Пол	Текстовый
Улица	Текстовый
Дом	Числовой
Квартира	Числовой
Класс	Числовой
Группа	Текстовый
Увлечение	Текстовый
Цвет глаз	Текстовый
Портрет	Поле объекта С

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	класс
Присоединенный столбец	1
Число столбцов	1
Заглавия столбцов	Нет
Ширина столбцов	
Число строк списка	16
Ширина списка	Авто
Ограничиться списком	Нет

Рисунок 1

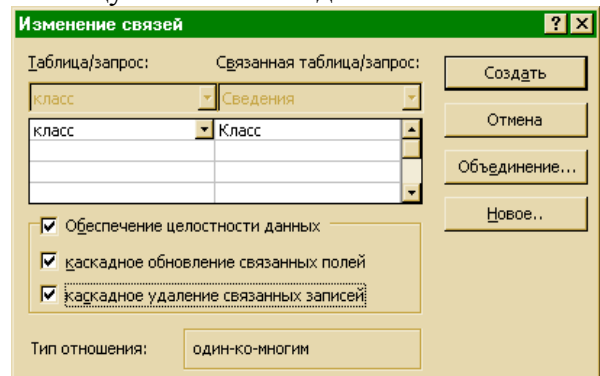


Рисунок 2

## 4. Создание связей

В Microsoft Access связи создаются в специальном конструкторе Схема данных. Вызвать команду Лента - Работа с базами данных, Группа - Отношения, Кнопка – Схема данных для вызова Появится окно Добавить таблицу. В появившемся списке выделить все таблицы, нажать последовательно Добавить, Закрыть. Раздвинуть рамки таблицы Сведения, чтобы выявить все поля.. Выделить поле Класс таблицы Класс, перетащить и наложить на поле Класс таблицы Сведения. В результате появится окно Изменение связей. Заполнить как показано на рис3, Создать. Появится связь 1:∞. Аналогично создать связь для полей Группа, Увлечения, Цвет глаз. Для полей Код (таблица оценки) и № (таблица Сведения) создать связь 1:1, при этом в окне Изменение связей установить флажок только для Обеспечения целостности данных. Сохранить. Закрыть Схему данных.

## Практическое занятие № 13.

### Среда программирования.

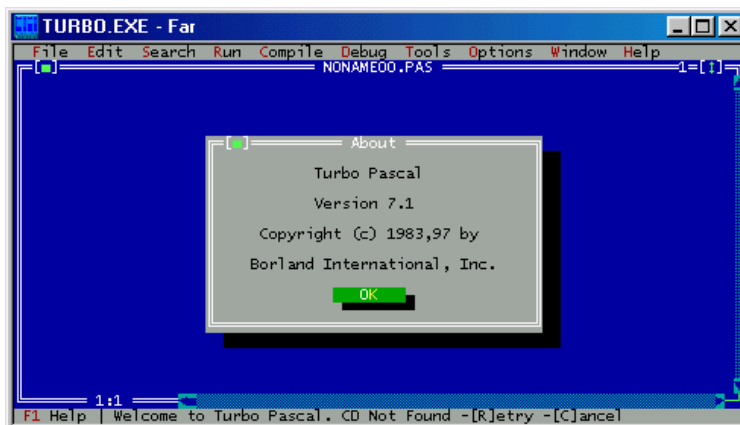
**Цель:** познакомиться со средой программирования Turbo Pascal 7.0, со структурой программы, написать первую программу на Turbo Pascal 7.0, сохранить её.

#### Элементы окна программы Turbo Pascal и их назначение.

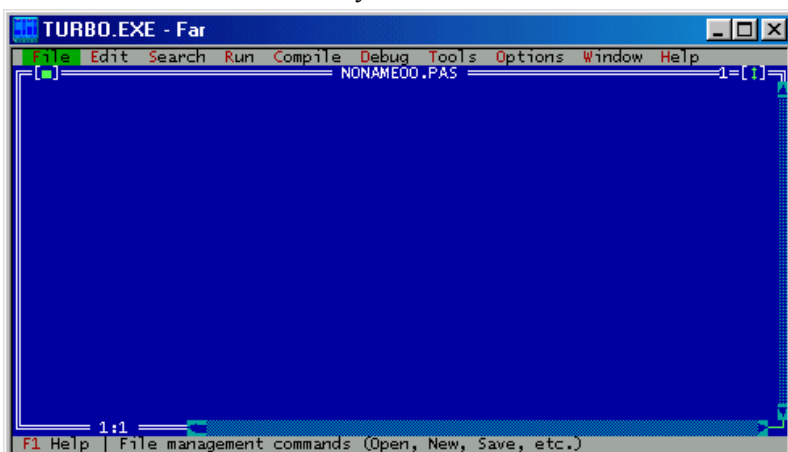
Рассмотрим окно программы Turbo Pascal. В верхней части синего окна программы Turbo Pascal находится её главное меню.

Оно содержит разделы:

- File - файл;
- Edit - редактирование текста;
- Search - поиск, исследование;
- Run - запуск на выполнение;
- Compile - компиляция;
- Debug - отладка;
- Tools - инструменты;
- Options - опции;
- Windows - окна;
- Help - помощь;

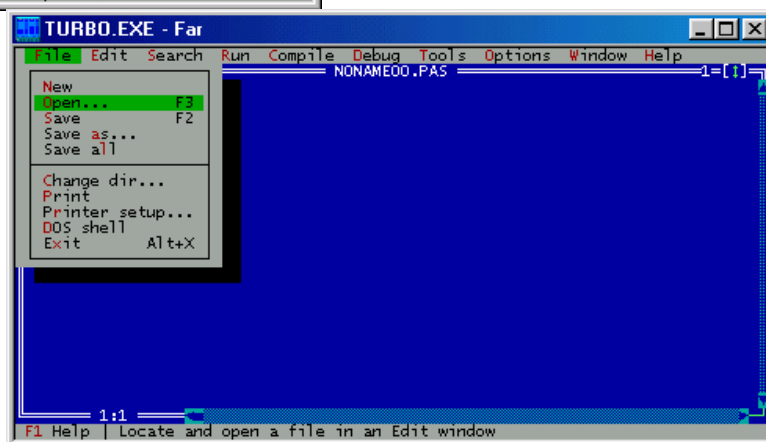


Будем пользоваться манипулятором мыши (или мышкой). А при ее отсутствии клавиатурой. Запустим программу Turbo.exe. Перед нами возникнет вышеуказанный рисунок. Удалим светлое окно "About" с основного синего окна программы Turbo Pascal, щелкнув по "Ок" на светлом окне левой кнопкой мыши или нажав на клавишу "ESC".



Чтобы активизировать окно (то есть, чтобы был доступ к его главному меню), нажмем на функциональную клавишу F10. При этом раздел "File" подсветится зеленым цветом. Теперь можно пользоваться стрелочками вправо или влево (> или <) и легко переходить от одного раздела главного меню к другому. Клавишей "Enter" можно раскрыть раздел и увидеть команды этого раздела в его

ниспадающем меню (окне). Существует еще другой способ, позволяющий активизировать главное меню и раскрыть вниз окно с командами выбранного Вами раздела. Для этого нужно нажать клавишу Alt и, не отпуская ее, нажать вместе еще клавишу заглавной буквы, выделенной в названии этого раздела красным цветом: F, E, S, R, C и так далее. При этом Вы сразу войдете в нужный Вам раздел. Вы увидите все команды этого раздела и сможете сразу же выбрать из них нужную Вам строку. Прделайте это. В нижеследующей таблице указаны входы в разделы главного меню с использованием клавиатуры:



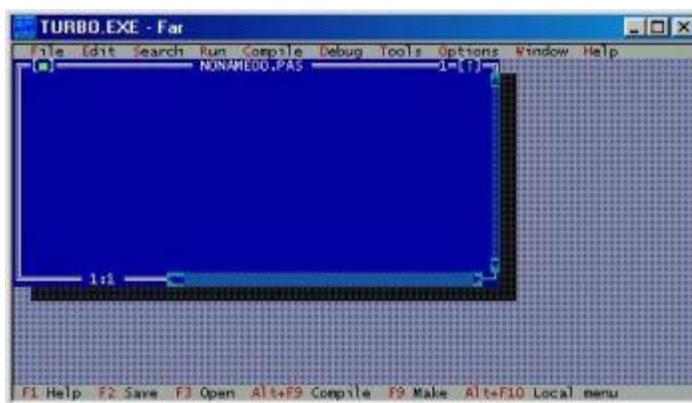
Кроме основного окна, рассмотренного выше, в Turbo Pascal имеется еще одно окно, называемое окном Output. Доступ к нему нужен, чтобы увидеть результат выполнения Вашей

программы, то есть другими словами просмотреть решение задачи. Для этого нажмите клавишу F5. Вы увидите черное окно

После вызова программы на экран редактора Turbo Pascal Вы можете совместить компиляцию исходного текста программы и запуск на выполнение скомпилированной программы. При обнаружении ошибки Вы, как обычно, должны будете тщательно проанализировать текст программы, найти причину ошибки и устранить ее. Для этого Вы должны нажать одновременно две клавиши: **Ctrl+F9**. Это все равно, что активизировать главное меню, войти в раздел "RUN" и выполнить команду "Run" этого раздела. При этом Turbo Pascal, как всегда, обратится прежде всего к проверке программы на наличие или отсутствие ошибок через этап компиляции. И только при хорошем исходе этого этапа начнет выполнять программу. Результатом этого способа компиляции будет компактный исполняемый код новой программы. Интегрированная среда программирования Turbo Pascal может работать с несколькими открытыми окнами. Главное здесь, чтобы существовала достаточная для этих целей внутренняя память компьютера.

Введем понятие активного окна. **Активным** считается окно, с которым Вы в данный момент работаете. Остальные окна считаются неактивными в данный момент. Каждое окно имеет:

- в верхней части:
  - главное меню команд;
  - закрывающую кнопку окна;
  - полосу заголовка и имя файла с исходным текстом программы;
  - номер окна;
- двойную рамку как признак активного окна;
- полосы прокрутки (скроллинга) - вертикальную и горизонтальную;
- в нижней части:
  - номер редактируемой строки программы, на которой находится курсор;
  - номер позиции в редактируемой строке;
  - строку статуса;
  - уголок для изменения размеров окна.



Уголок изменения размеров окна находится в нижнем правом углу окна и имеет одинарную рамку вместо двойной рамки.

### Общая структурная схема Pascal - программы.

В первых трансляторах языка PASCAL порядок следования разделов PASCAL - программы был строго фиксированным. Однако, в современных реализациях языка PASCAL порядок следования описаний LABEL, CONST, TYPE, VAR произвольный. Ниже в таблице представлены наименования разделов Pascal - программы и внешний вид каждого раздела в Pascal - программе.

Наименование раздела Pascal - программы	Внешний вид раздела в Pascal - программе
Заголовок программы	<b>Program</b> <имя_программы>;
Раздел описания меток	<b>Label</b> <метка>, ..., <метка>;

Раздел описания констант	<b>Const</b> <имя_константы>=<константа>; <имя_константы>=<константа>; ...
Раздел описания типов	<b>Type</b> <имя_типа>=<тип>; <имя_типа>=<тип>; ...
Раздел описания переменных	<b>Var</b> <имя_переменной>,...,<имя_переменной>:<тип>; <имя_переменной>,...,<имя_переменной>:<тип>; ...
Раздел процедур	<b>Procedure</b> <заголовок_процедуры>; <Блок>;
Раздел функций	<b>Function</b> <заголовок_функции>; <Блок>;
Раздел операторов	<b>Begin;</b> <оператор>; ...; <оператор>; <b>End.;</b>

В **ЗАГОЛОВКЕ ПРОГРАММЫ** после служебного слова PROGRAM обязательно указывается имя программы. Оно обозначает всю программу в целом. Внутри самой программы это имя не используется.

После заголовка программы следует **РАЗДЕЛ ОПИСАНИЯ МЕТОК ( LABEL )**. Любой оператор программы можно выделить (или пометить), поставив перед ним метку. Когда возникает необходимость перейти к какому-то оператору программы для продолжения алгоритма или вернуться назад и многократно повторить действия, пользуются метками. **МЕТКА** - это целое число без знака, обычно содержащее не более 4-х знаков. Метка отделяется от оператора двоеточием (:). Например:28:READ(X0, Y0, Z0); Метки в программе дают возможность ссылаться на нужные операторы в специальных операторах управления и изменять естественный ход выполнения программы. Все метки, используемые в программе, должны быть перечислены в специальном разделе меток (LABEL).Например:LABEL1, 17, 35, 128; Вообще же раздел меток может отсутствовать, если в программе нет меток.

За разделом меток следует **РАЗДЕЛ ОПИСАНИЯ КОНСТАНТ (CONST)**. Константами, как известно, называются величины, которые не меняются в процессе выполнения программы или алгоритма. Константы могут быть представлены в программе именем. Тогда в разделе описания констант этим именам должны соответствовать (должны быть присвоены) некоторые числовые или символьные значения. Символьные значения иногда называются строковыми или текстовыми значениями. Например:CONSTPI=3.1415826;HAND=10000;СТРОКА='РАСЧЕТ ЗАПЛАТЫ'; *Отметим*, что выделив группу констант в задаче, мы можем без особого труда при необходимости изменить их значения, не меняя самой программы. Еще отметим, что программы, имеющие выделенную группу констант, делаются более наглядными и удобочитаемыми. Это немаловажное обстоятельство, особенно при отладке или изменении программы. Тип константы определяет форма записи ее значения.

**РАЗДЕЛ ОПИСАНИЯ ТИПОВ (TYPE)** служит для определения новых типов данных, определяемых пользователем. Каждая величина, используемая в программе, занимает определенную память компьютера. Тип величины определяет, как расположена эта величина в памяти ЭВМ. С учетом этого ведется дальнейшая обработка данных, участвующих в расчетах. Каждая величина в программе сопоставляется с одним и только с одним типом данных.

Тип переменной величины обязательно задается в аналогичном **РАЗДЕЛЕ ОПИСАНИЯ ПЕРЕМЕННЫХ (VAR)**. На **СТРУКТУРНОЙ СХЕМЕ PASCAL - ПРОГРАММЫ** показан вид раздела переменных. Здесь показатель "ТИП" - это имя\_типа, например, стандартное или определенное в разделе типов TYPE.

**РАЗДЕЛ ОПИСАНИЯ ПРОЦЕДУР и РАЗДЕЛ ОПИСАНИЯ ФУНКЦИЙ (PROCEDURE, FUNCTION)** может отсутствовать в программе. Он присутствует в программе, если помимо стандартных функций и процедур, программист определяет свои процедуры или функции, к которым осуществляется

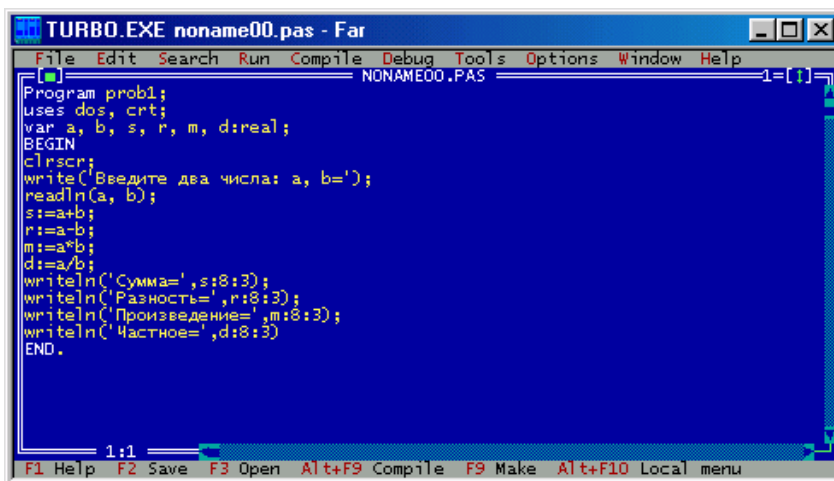
обращение из основной программы. Каждая процедура или функция может иметь описания меток, констант, типов, переменных, процедур или функций.

Далее следует **РАЗДЕЛ ОПЕРАТОРОВ**. Это составной оператор, **BEGIN...END**. Содержит последовательность действий, ведущую к решению задачи, реализует алгоритм решения задачи, ведущий к достижению результата.

Некоторые из вышеназванных разделов описаний могут отсутствовать в разделе описаний PASCAL-программы.

А теперь рассмотрим приведенную ниже PASCAL-программу.

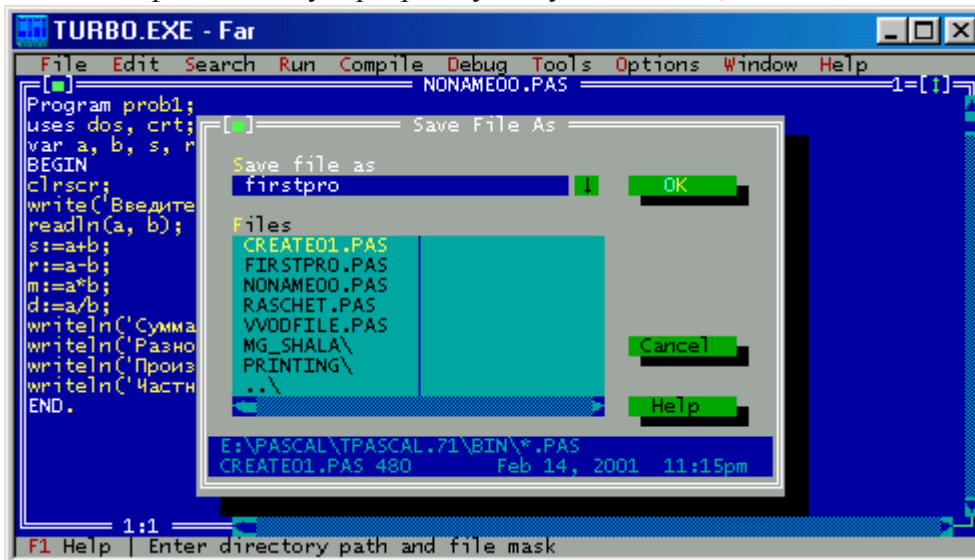
В этой программе даны два числа: *a* и *b*. Необходимо рассчитать сумму, разность, произведение и частное от деления этих чисел (при условии, что число *b* отлично от нуля).



Наберите текст программы:

```
Program prob1;
  uses dos, crt;
  var a, b, s, r, m, d:real;
  BEGIN
    clrscr;
    write('Введите два числа a, b=');
    readln( a, b);
    s:=a+b;
    r:=a-b;
    m:=a*b;
    d:=a/b;
    writeln('Сумма =',s:8:3);
    writeln('Разность=',r:8:3);
    writeln('Произведение=',m:8:3);
    writeln('Частное=',d:8:3)
  END.
```

Сохраним нашу программу. С учетом того, что Вы еще не дали наименование



файлу с исходным текстом программы на Паскале, здесь пока что записано **noname00.pas**, то есть программа без имени. Ведь название дается программе при записи ее исходного текста



на диск. Займемся этим сейчас.

Вторая строка окна программы Turbo Pascal это строка меню команд. Найдите в ней раздел "File" с выделенной красным цветом заглавной буквой "F". Откройте этот раздел, щелкнув, как принято, левой кнопкой "мыши" по имени раздела или нажав вместе две клавиши: "Alt+F". Перед Вами раскроется вниз удлинненное окно с командами. Это окно называют ниспадающим меню файловых операций.

Выберите здесь команду "Save as...". Щелкните по ней левой кнопкой мыши или нажмите клавишу "Enter". Как известно, троеточие в конце команды означает, что для ее выполнения необходима дополнительная информация. Запишите текст программы на жесткий, гибкий или CD RW диск. В дополнительное светлое окно "Save file as" занесите "имя программы". Дайте своей первой программе имя *firstpro.pas*. Нажмите кнопку "Ok". Программа запишется на внешний носитель.

#### Построчные пояснения к каждой строке программы

**Строка 1: Program prob1;** Вводится для идентификации программы, то есть для того, чтобы дать ей имя или наименование. "Program" это в общем совершенно любое имя, которое вздумается назвать составителю программы. Оно никак транслятором не контролируется. Главное одно: чтобы оно было. Если же вздумаете вовсе опустить эту строку, то ничего страшного не произойдет: просто система не выдаст сообщение об ошибке. Стало быть эта строка вовсе необязательна. Но, однако, если уж она написана, то ключевое слово "Program" нужно писать точно, не допуская в нем ошибок. Это транслятор улавливает. Уместно здесь же пояснить, что нет никакой существенной разницы, какими буквами будет написано это слово: "program" или "Program" или "PROGRAMM". Оказывается транслятор не различает в этом случае строчные и прописные буквы.

**Строка 2: uses dos, crt;**

Здесь перечисляются названия пользовательских библиотек Паскаля, из которых берутся подпрограммы, библиотечные функции при компиляции программы.

**Строка 3: var a, b, s, r, m, d:real;**

Это раздел переменных. Он обозначается в программе служебным словом "var". После этого ключевого слова обязательно перечисляются используемые в задаче переменные с указанием их типов.

Очевидно, что это:

исходные числа: a и b;

s - это сумма исходных чисел a и b;

r - разность этих чисел;

m - это их произведение;

d - результат выполнения операции деления.

**Строка 4: BEGIN**

Это обозначение начала программы. В программе каждому "begin" должно соответствовать одно "end". Входит в составной оператор begin...end или в операторные скобки.

**Строка 5: clrscr;**

Функция "clrscr;" имеет назначение: очистить экран.

**Строка 6: write ('Введите два числа a, b=');**

Оператор write предназначается для вывода на экран подсказки пользователю. Подсказка указывается в кавычках на русском языке. Этот текст будет выведен на экран. Он показывает или напоминает, что нужно сделать в данный момент.

**Строка 7: readln(a, b);**

Оператор read используется для чтения информации в память компьютера. В переменные a и b заносится исходная числовая информация при вводе ее с клавиатуры.

**Строка 8: s:=a+b;**

Вычисляется "s" как сумма чисел a и b.

**Строка 9: r:=a-b;**

Вычисляется "r" как разность чисел a и b.

**Строка 10: m:=a\*b;**

Вычисляется "m" как произведение чисел a и b.

**Строка 11: d:=a/b;**

Вычисляется "d" как частное от деления чисел a и b.

**Строка 12: writeln('Сумма =',s:8:3);**

Выдается на экран пояснение 'Сумма =' и результат: число, под которое отведено 8 знаков, в том числе три дробных знака, после запятой.

**Строка 13: writeln('Разность =',r:8:3);**

Выдается на экран пояснение 'Разность =' и результат: число, под которое отведено 8 знаков, в том числе три дробных знака, после запятой.



Строка 14: `writeln('Произведение=',m:8:3);`

Выдается на экран пояснение 'Произведение = ' и результат: число, под которое отведено 8 знаков, в том числе три дробных знака, после запятой.

Строка 15: `writeln('Частное=',d:8:3)`

Выдается на экран пояснение 'Частное = ' и результат: число, под которое отведено 8 знаков, в том числе три дробных знака, после запятой.

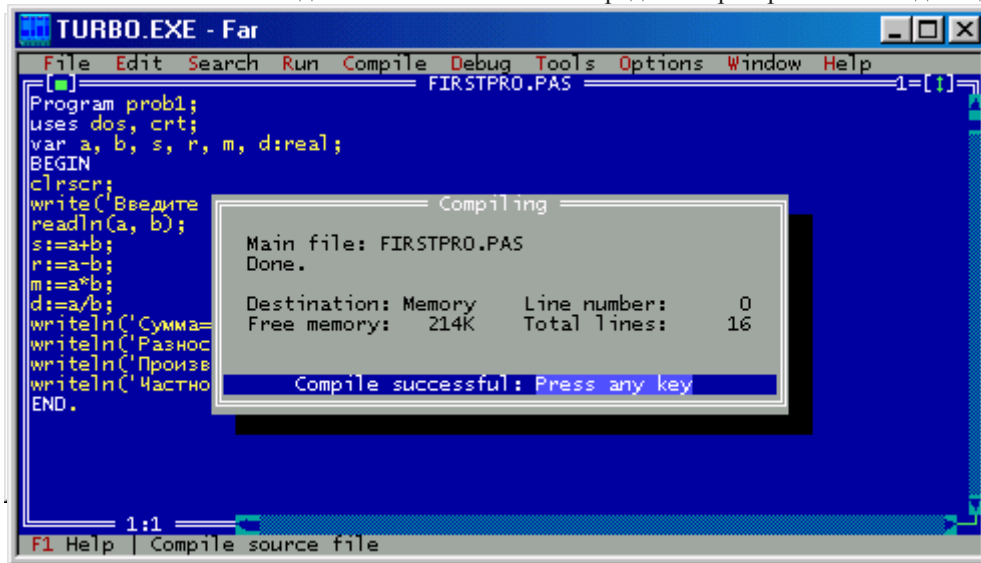
Строка 16: `END.`

Конец программы. Закрывает операторные скобки `begin...end.`

**Теперь необходимо выполнить этап трансляции программы и завершить все решением задачи после успешной трансляции.**

**Этап трансляции программы. Исправление обнаруженных ошибок в тексте.**

Во второй строке окна Turbo Pascal найдите раздел "Compile" - компиляция. Раскройте его щелчком мышки или нажав вместе две клавиши: "Alt+F9". Перед Вами раскроется выпадающее меню.



Выберите в нем команду: "Compile" и выполните ее, нажав клавишу "Enter" или щелкнув мышкой по тексту команды.

Компиляция программы выполнена без ошибок. Вы получили сообщение:

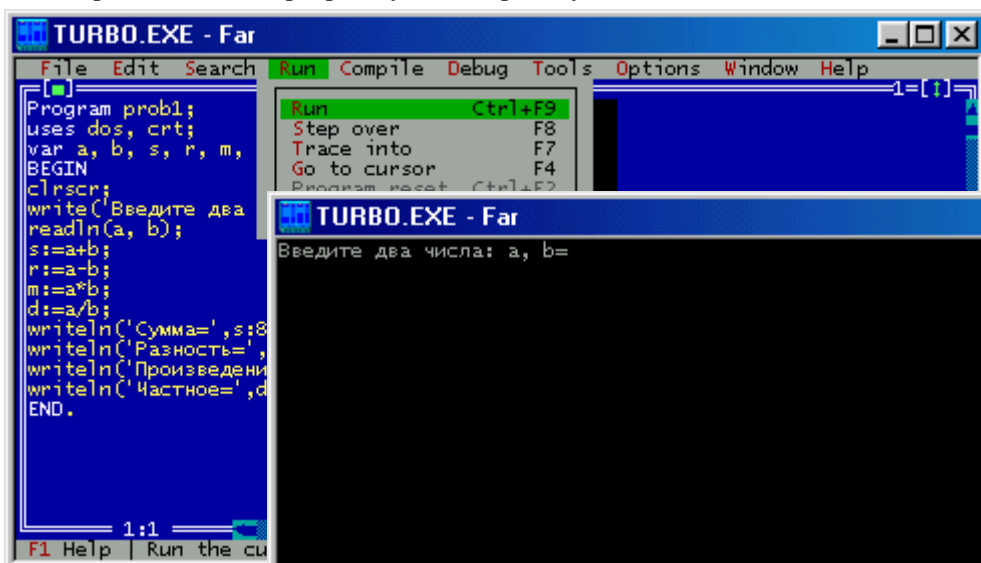
"Compile successful".

Компиляция завершена успешно.

Вполне понятно, что такая

маленькая программка, как начальная программа, может не содержать ошибок. Тогда мой совет: поэкспериментируйте с текстом программы. Внесите в него искусственно ошибки и пристально наблюдайте, как Turbo Pascal сообщает Вам о них. Изучайте сообщения транслятора об ошибках и учитесь понимать их на лету.

**Теперь выполните программу, не содержащую ошибок компиляции.**



Для этого войдите в раздел "Run", выберите в нем команду "run" и щелкните мышкой по этой команде или

нажмите клавишу "Enter".

Вы могли бы сразу нажать две клавиши вместе:

"Ctrl+F9".

В результате будет снова автоматически выполнена компиляция и, если отсутствуют ошибки компиляции, то за ней автоматически последует обращение к готовой программе и решение задачи.

**Программа запросила любые два числа.** В числах взамен обыкновенной запятой Вы должны нажимать точку. После первого числа Вы нажимаете пробел и вводите второе число. После ввода второго числа нажимаете "Enter". Перед Вами появляется ответ с пояснениями.

Вы можете после ввода первого числа нажать клавишу "Enter", но тогда будете набирать второе число на следующей строке. Но это никак не повлияет на правильность результата.

## Практическое занятие № 14.

### Программирование линейного алгоритма.

**Цель:** научиться создавать, вводить в компьютер, выполнять и исправлять простейшие программы на языке Паскаль в режиме диалога, познакомиться с диагностическими сообщениями компилятора об ошибках при выполнении программ, реализующих линейные алгоритмы.

**Линейным** называется алгоритм, в котором результат получается путем однократного выполнения заданной последовательности действий при любых значениях исходных данных. Операторы программы выполняются последовательно, один за другим, в соответствии с их расположением в программе.

**Пример.** *Определить объем и площадь боковой поверхности цилиндра с заданными радиусом основания  $R$  и высотой  $H$ .*

#### Тест

Данные		Результат	
$R = 1$	$H = 1$	$V = 3.14$	$S = 6.28$

```
Program Cylinder;
Uses Crt; {Подключение библиотеки Crt}
Var
  R, {радиус основания цилиндра}
  H, {высота цилиндра }
  V, {объем цилиндра }
  S: Real; {площадь боковой поверхности цилиндра}
BEGIN
  ClrScr; {Вызов из библиотеки Crt процедуры очистки экрана}
  Write('Введите высоту цилиндра : '); ReadLn(H);
  Write('Введите радиус основания : '); ReadLn(R);
  V := Pi * R * R * H;
  S := 2 * Pi * R * H; WriteLn;
  WriteLn('Объем цилиндра = ', V : 5 : 2); {Здесь 5 - общее
    количество позиций, занимаемых переменной V при выводе,
    а 2 - количество позиций в дробной части значения V}
  WriteLn('Площадь боковой поверхности = ', S : 5 : 2);
  ReadLn
END.
```

**Решите задачи на программирование алгоритмов линейной структуры (по вариантам):**

1. Даны два числа  $a$  и  $b$ . Получить их сумму, разность и произведение.
2. Даны действительные числа  $x$  и  $y$ . Получить  $(|x| - |y|) / (1 + |x \cdot y|)$ .
3. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
4. Даны два действительных положительных числа. Найти среднее арифметическое и среднее геометрическое этих чисел.
5. Даны катеты прямоугольного треугольника. Найти его гипотенузу и площадь.
6. Определить периметр правильного  $n$ -угольника, описанного около окружности радиуса  $r$ .
7. Дана сторона равностороннего треугольника. Найти площадь этого треугольника.
8. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
9. Найти площадь кольца, внутренний радиус которого равен  $20$ , а внешний – заданному числу  $r$  ( $r > 20$ ).
10. Найти площадь равнобокой трапеции с основаниями  $a$  и  $b$  и углом  $\alpha$  при большем основании  $a$ .
11. Найти площадь сектора, радиус которого равен  $3,7$ , а дуга содержит заданное число радиан  $\varphi$ .
12. Три сопротивления  $R_1, R_2, R_3$  соединены параллельно. Найти сопротивление соединения.
13. Определить время падения камня на поверхность земли с высоты  $h$ .
14. Вычислить высоту треугольника, опущенную на сторону  $a$ , по известным значениям длин его сторон  $a, b, c$ .
15. Вычислить объем цилиндра с радиусом основания  $r$  и высотой  $h$ .
16. Определить расстояние, пройденное физическим телом за время  $t$ , если тело движется с постоянным ускорением  $a$  и имеет в начальный момент времени скорость  $V_0$ .
17. Вычислить площадь треугольника по формуле Герона, если заданы его стороны.
18. Определить координаты вершины параболы  $y = ax^2 + bx + c$  ( $a \neq 0$ ). Коэффициенты  $a, b, c$  заданы.
19. По данным сторонам прямоугольника вычислить его периметр, площадь и длину диагонали.

### **Практическое занятие № 15.**

#### **Графический режим. Моделирование геометрических и физических задач.**

**Цель:** усвоить параметры графического экрана, расширить представление студентов о возможностях машинной графики на различных примерах.

В языке TURBO Pascal имеется значительное количество графических процедур и функций. Нам понадобятся лишь некоторые из них. Для того, чтобы компилятор "узнавал" их названия, мы должны после заголовка программы разместить строчку следующего вида:

**uses Graph;** (что в переводе на русский означает "используется графика").

До сих пор во время нашей работы за компьютером экран всегда находился в текстовом режиме.

Для рисования прямых, окружностей и пр. необходимо перевести экран в графический режим. Для включения графического режима используется процедура **InitGraph**.

Простейшая программа может иметь вид:

**Пример 1.**

```
PROGRAM Primer_1;
  uses graph,crt;
  var driver, mode: Integer;
BEGIN
  driver:=detect;
  Initgraph (driver, mode, ' '); { Включить графический режим }
  Line (0,0,639,479); { Нарисовать отрезок прямой }
  ReadLn
END.
```

Мы видим, что у процедуры InitGraph три параметра. В качестве первых двух фактических параметров должны стоять имена целых (integer) переменных.

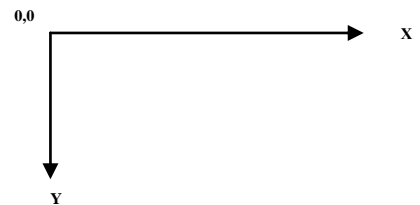
Первый параметр является кодом графического адаптера (т.е. электронной схемы, управляющей выводом информации на экран). (Дело в том, что на IBM-совместимых компьютерах применяется ряд стандартных графических адаптеров, носящих названия CGA, EGA, VGA.).

По нашей программе можно догадаться, что в используемых нами компьютерах используется адаптер VGA, и компилятор сам "узнает" слово VGA и заменит его на нужное целое число (на самом деле равное 9, но мы этого можем и не запоминать).

Каждый графический адаптер позволяет использовать несколько графических режимов, отличающихся количеством цветов и разрешающей способностью. И второй из параметров как раз предназначен для того, чтобы указать, какой из графических режимов следует включить. Пока что мы ограничимся лишь одним графическим режимом VGAHi.

Третий параметр является строкой, содержащей путь к файлу, который называется EGAVGA.BGI. В этом файле содержится драйвер (такая специальная программа), необходимый для работы с адаптерами EGA и VGA. И, как легко увидеть из нашего примера, файл этот находится в подкаталоге TP/BGI.

Для того, чтобы мы могли что-либо нарисовать на экране, нам нужно уметь задавать положение на экране того, что мы рисуем. Для этого с экраном связывается система координат следующего вида:



Каждая точка на экране на самом деле представляет собой очень маленький прямоугольник (и поскольку это не совсем точка, то иногда используют специальный термин - "пиксел"). Количество точек (пикселов), умещающихся на экране по вертикали и горизонтали, называют разрешающей способностью. Разрешающая способность экрана в режиме VGAHi - 640x480. Это означает, что по горизонтали на экране умещается 640 точек, а по вертикали - 480.

Точка в левом верхнем углу экрана имеет координаты (0,0). Координата X любой точки экрана лежит в пределах от 0 до 639, а координата Y - в пределах от 0 до 479.

Как Вы уже догадались, процедура Line (x1,y1,x2,y2) рисует на экране прямую, соединяющую точки (x1,y1) и (x2,y2).

Довольно обидно работать на цветном мониторе только с черно-белыми изображениями. Для задания цвета рисования прямых, окружностей, точек и пр. используется процедура SetColor. В качестве единственного ее параметра нужно указать

целое число, являющееся кодом цвета. Цвета кодируются в соответствии со следующей кодовой таблицей :

Black	0	черный	DarkGray	8	темно-серый
Blue	1	синий	LightBlue	9	голубой
Green	2	зеленый	LightGreen	10	светло-зеленый
Сyan	3	цвет морской волны	LightCyan	11	светло-циановый
Red	4	красный	LightRed	12	розовый
Magenta	5	сиреневый	LightMagenta	13	светло-сиреневый
Brown	6	коричневый	Yellow	14	желтый
LightGray	7	светло-серый	White	15	белый

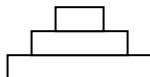
- CLS – очищает экран перед выводом программы (обязательно!)
- LINE (x1,y1)-(x2,y2),c – рисует отрезок с началом в т.(x1,y1) и концом в т. (x2,y2), цветом с (по умолчанию рисует белым цветом)

*например:* line(100,100)-(200,200)

- LINE (x1,y1)-(x2,y2),c,B(BF) –рисует незакрашенный (закрашенный) прямоугольник, где (x1,y1) – координаты верхнего левого угла, (x2,y2) – координаты правого нижнего угла.
- PSET(X,Y),c – рисует точку с заданными координатами x,y
- COLOR c1,c2 – задает цвет изображения c1, цвет фона c2

*например:* COLOR 4,14

**Задание:** Составить программы, рисующие следующие фигуры.



*Для справки:*

CIRCLE(x,y),r,c – рисует окружность с центром в точке (x,y), радиусом r, цветом c

*Пример: нарисовать в центре экрана окружность радиусом 50*

*CIRCLE (320,240), 50, 5*

***Набрать программы и запустить на выполнение:***

### **Рисуем Российский флаг.**

```
uses graph,crt;
```

```
var
```

```
  x,y:integer; {координаты левого верхнего угла флага}
```

```
  l,h:integer; {длина и высота флага}
```

```
  w:integer; {ширина полосы флага}
```

```
  driver:integer;
```

```
  mode:integer;
```

```
begin
```

```
  driver:=detect;
```

```
  initgraph(driver,mode,"");
```

```
  x:=100;
```

```
  y:=100;
```

```
  l:=50;
```

```
  h:=25;
```

```

w:=round(h/3);
{ рисуем флаг }
setfillstyle(solidfill,white);
bar(x,y,x+l,y+w);
setfillstyle(solidfill,blue);
bar(x,y+w,x+l,y+2*w);
setfillstyle(solidfill,red);
bar(x,y+2*w,x+l,y+3*w);
outtextXY(x,y+h+5,'Rossia');
readln;
closegraph;
end.

```

### Рисуем Олимпийский флаг.

```

uses graph,crt;
var
  driver:integer; { драйвер }
  mode:integer; { графический режим }
begin
  driver:=detect;
  initgraph(driver,mode,"");
  { рисуем флаг }
  setfillstyle(solidfill,lightgray); { сплошная заливка серым цветом }
  bar(80,80,200,135);
  { кольца }
  setcolor(green);
  circle(100,100,15);
  setcolor(black);
  circle(140,100,15);
  setcolor(red);
  circle(180,100,15);
  setcolor(yellow);
  circle(120,115,15);
  setcolor(blue);
  circle(160,115,15);
readln;
closegraph;
end.

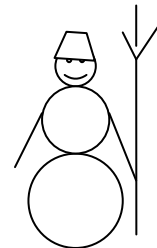
```

**Задание:** Нарисовать снеговика (внести недостающие данные в программу)

```

Circle (320,370), 80, 3
Circle (320,230), 60, 3
Circle (320,130), 40, 3
Line (290,110)-(300, 50), 1
Line (340, 50), 1
Line (350,110), 1
Line (290,110), 1
Paint (320, 80), 8, 1
Pset (300,120), 0
Pset (340,120), 0
Circle (320,140), 10, 13, 0.5

```



## Практическое занятие № 16.

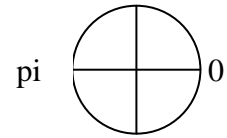
### Вывод на монитор фигуры, состоящей из дуг и отрезков.

**Цель:** выработать у студентов навыки составления программ, содержащих операторы графики.

*Для справки:*

`CIRCLE(x,y),r,c,f1,f2` – рисует дуги, где `f1` – начало дуги, `f2` – конец дуги (в радианах)

*Пример:* `CIRCLE (320,240), 50, 5, 0, pi`



`CIRCLE(x,y),r,c,f1,f2,k`- полный формат оператора, где `k`-коэффициент сжатия, т.е. отношение вертикальной оси к горизонтальной.

*Пример: нарисовать эллипс* `CIRCLE(320,240),50,5,,,0.5`

`CIRCLE (320,240), 50, 5, 4`

Процедура `Circle` вычерчивает на экране окружность. В качестве параметров этой процедуры используются координаты центра окружности и длина радиуса. Процедура `Arc` вычерчивает на экране дугу, у которой указываются координаты центра, начальный угол, конечный угол и длину радиуса.

### Пример. Рисование окружности и дуги с помощью процедур `Circle` и `Arc`.

```
1) Program Graphika12;
2) Uses crt, Graph;
3) var y:integer;
4) Procedure Graphinterface;
5) Var gd, gm, error:Integer;
6) s:String;
7) Begin
8) gd:=detect;
9) s:=' ';
10) Initgraph(gd,gm,s);
11) error:=GraphResult;
12) if error<>GrOk then
13) begin
14) writeln(GraphErrorMsg(Error));
15) Halt(error)
16) end
17) end;
18) begin
19) Graphinterface;
20) setcolor(2);
21) Circle(300,200,100);
22) y:=0;
23) while y<=360 do
24) begin
25) arc( 600,200,0,y,100);
26) y:=y+10;
27) delay(1000);
28) end;
29) readln;
30) end.
```

---

Строка №3. Указываем переменную для хранения значения конечного угла при построении дуги.

В строке №20 записываем процедуру `SetColor`. Она применима к процедурам `Circle` и `Arc` и будет определять цвет окружности и дуги, которые мы зададим в строках №21 и №25.

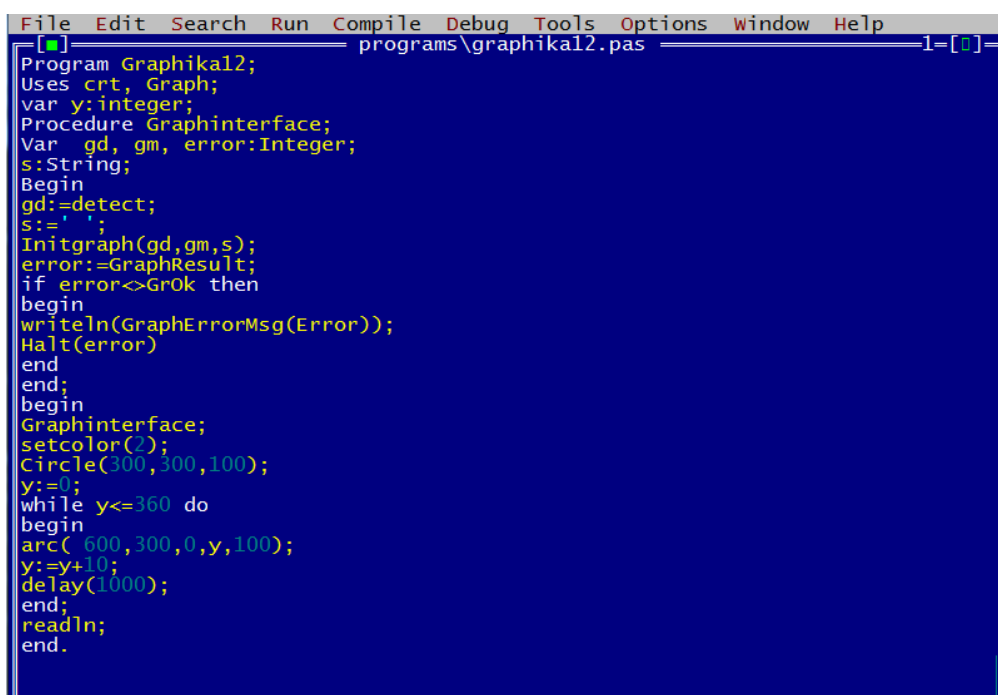
Строка №21. Процедура Circle рисует окружность на экране. После записи процедуры в круглых скобках указываются параметры процедуры. Первые 2 параметра определяют координаты X и Y для центра окружности (в нашем примере координата X имеет значение 300 пикселей, а координата Y – 200 пикселей). Третий параметр процедуры Circle определяет радиус окружности (в нашем примере радиус имеет значение 100 пикселей). Таким образом, на экран выведется окружность зеленого цвета с центром в точке (300,200) и радиусом 100 пикселей.

Строка №22. Присваиваем переменной «у» начальное значение равное 0, т.е. конечный угол дуги равен 0.

Строка №23. Записываем цикл. Пока «у», т.е. конечный угол меньше либо равен 360 градусов, должно выполняться условие, которое следует за словом До. За словом До следуют 3 операции.

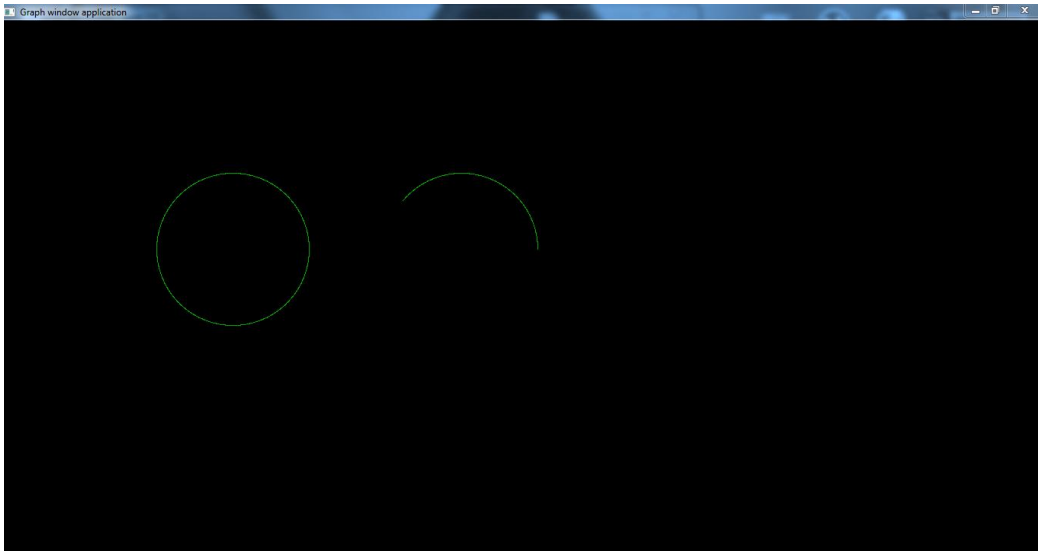
Строка №25. Первая операция – это процедура Arc, которая вычерчивает на экране дугу. Данная процедура имеет 5 параметров. Первые 2 параметра (в нашем примере 600,200) определяют координаты центра дуги. Третий параметр определяет начальный угол дуги в градусах (в нашем примере он равен 0). Четвертый параметр определяет конечный угол дуги в градусах (в нашем примере он равен переменной «у»). Пятый параметр определяет радиус дуги (в нашем примере 100 пикселей).

Строка №26. Увеличиваем переменную «у» на 10 градусов. Благодаря тому, что строка №25 и №26 находятся в цикле, значение конечного угла будет увеличиваться от 0 до 360 с шагом в 10 градусов, причем это увеличение будет идти со скоростью 10 градусов в секунду благодаря процедуре Delay в строке №27. Таким образом, на экран выведется дуга с центром в точке (600,200) и радиусом 100 пикселей. Начальный угол дуги будет равен 0 градусов. Конечный угол дуги также равен 0 градусов и растет на 10 градусов в секунду.



```
File Edit Search Run Compile Debug Tools Options Window Help
programs\graphika12.pas 1-[0]
Program Graphika12;
Uses crt, Graph;
var y:integer;
Procedure Graphinterface;
Var gd, gm, error:Integer;
s:String;
Begin
gd:=detect;
s:= ' ';
Initgraph(gd, gm, s);
error:=GraphResult;
if error<>GrOk then
begin
writeln(GraphErrorMsg(Error));
Halt(error)
end;
end;
begin
Graphinterface;
setcolor(2);
Circle(300,300,100);
y:=0;
while y<=360 do
begin
arc( 600, 300, 0, y, 100);
y:=y+10;
delay(1000);
end;
readln;
end.
```





**Задание:**

- Разработать свой рисунок, содержащий не менее 10 фигур.
- Нарисовать радугу.
- Нарисовать волну с использованием процедуры рисования эллипса
- Подсказки. Процедура рисования эллипса позволяет рисовать различные дуги. Можно нарисовать дугу от 0 до 180 градусов, и можно нарисовать дугу от 180 до 360 градусов. Если эти две дуги сместить относительно друг друга (по горизонтали) на величину диаметра то и получится нечто похожее на волну. Останется только повторить эту процедуру многократно.

**Практическое занятие № 17.**

**Создание структуры Условного перехода в конструкторе блок-схем MS Visio. Условный оператор.**

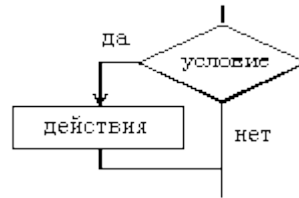
**Цель:** знакомство с базовой структурой Ветвления, умение составлять блок-схемы, реализующие алгоритмы ветвления.

**Базовая структура "ветвление".** Обеспечивает в зависимости от результата проверки условия (**да** или **нет**) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Язык блок-схем

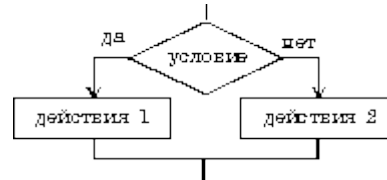
1. **если—то**

**если условие  
то действия**



## 2. если—то—иначе

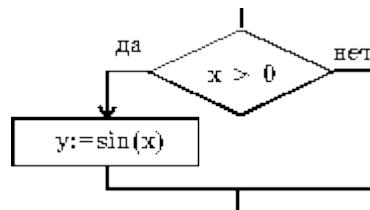
**если условие  
то действия 1  
иначе действия 2**



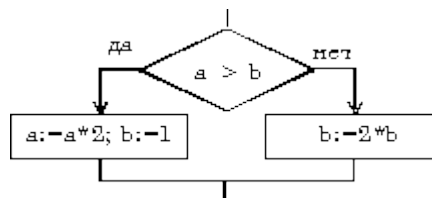
## Примеры структуры ветвление

### Язык блок-схем

**если  $x > 0$   
то  $y := \sin(x)$**



**если  $a > b$   
то  $a := 2*a; b := 1$   
иначе  $b := 2*b$**

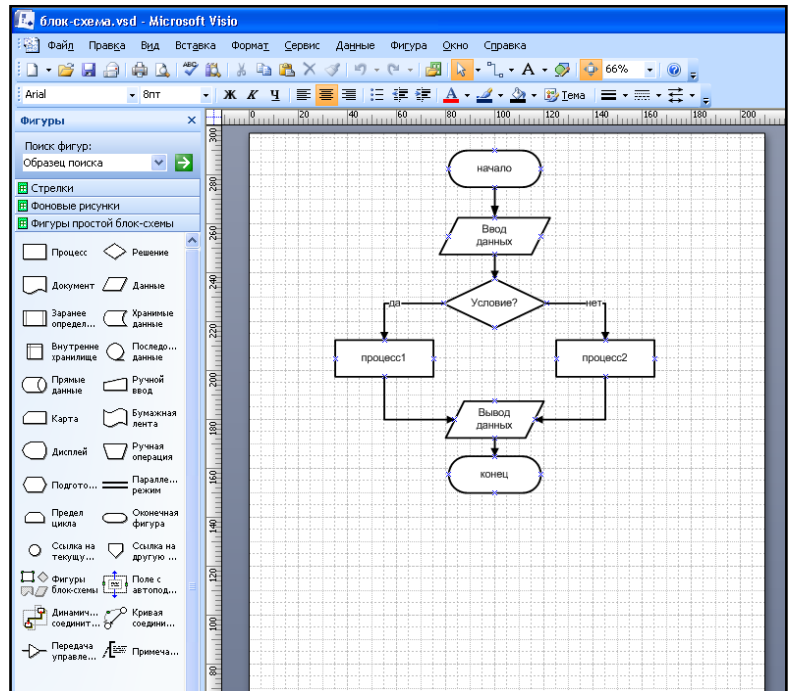


**Создать блок–схему решения задачи (в программе MS Visio):**

### Задания:

1. Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае не изменять его. Вывести полученное число.
2. Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае вычесть из него 2. Вывести полученное число.
3. Дано целое число. Если оно является положительным, то прибавить к нему 1; если отрицательным, то вычесть из него 2; если нулевым, то заменить его на 10. Вывести полученное число.
4. Даны два числа. Вывести большее из них.
5. Даны два числа. Вывести вначале большее, а затем меньшее из них.
6. Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных А и В.

7. Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной большее из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных А и В.
8. Даны три числа. Найти наименьшее из них.
9. Даны три числа. Найти среднее из них (то есть число, расположенное между наименьшим и наибольшим).



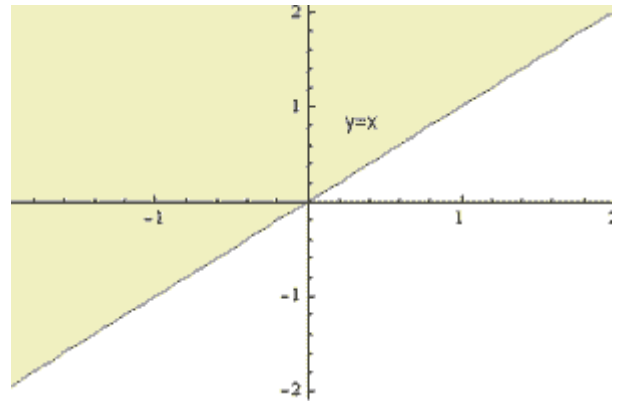
## Практическое занятие № 18.

### Восстановление области по заданному логическому выражению.

**Цель:** уметь определять принадлежность точки заштрихованной области.

**Поставим задачу иначе:** 1. Пусть необходимо определить попадает ли заданная точка с координатами  $(x, y)$  в заданную область:

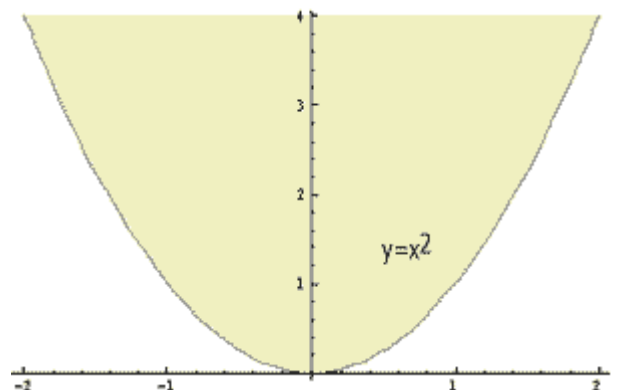
В данном случае, очевидно, что точка  $(x, y)$  попадает на линию графика, если  $y=x$ . Точка попадает в закрашенную область, лежит выше линии графика, если  $y>x$ . Наконец, точка  $(x, y)$  лежит ниже линии графика, если  $y<x$ . Таким образом, условие попадания точки в закрашенную область:



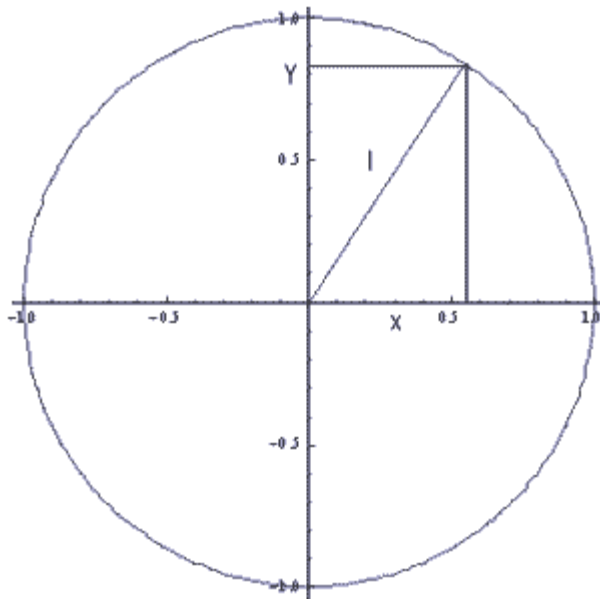
`if  $y>x$  then writeln('Да, попадает') else writeln('Нет');`

2. Аналогично, точка  $(x, y)$  попадает в закрашенную область, если  $y>x^2$ .

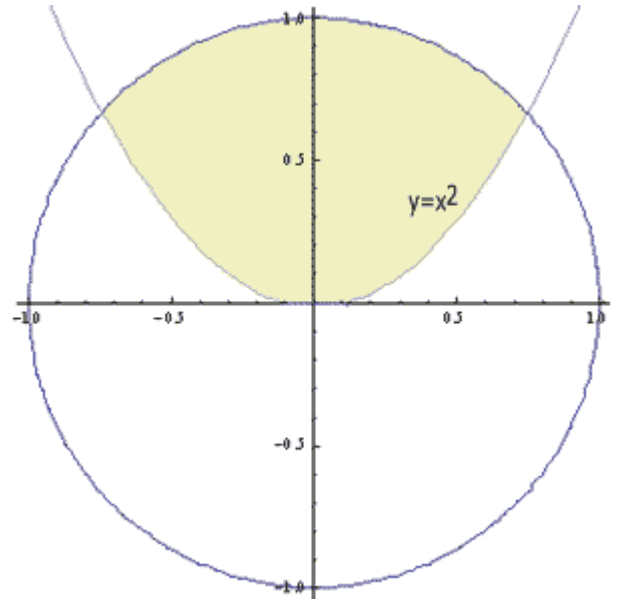
`if  $y>x*x$  then writeln('Да, попадает')`  
`else writeln('Нет');`



3. Рассмотрим попадание точки в круг. Уравнение окружности:  $x^2+y^2=r^2$ , в представленном на рисунке случае  $r=1$ .



Точка  $(x, y)$  лежит на линии окружности, если расстояние до нее  $l = \sqrt{x^2 + y^2}$ , если  $l=r$ .

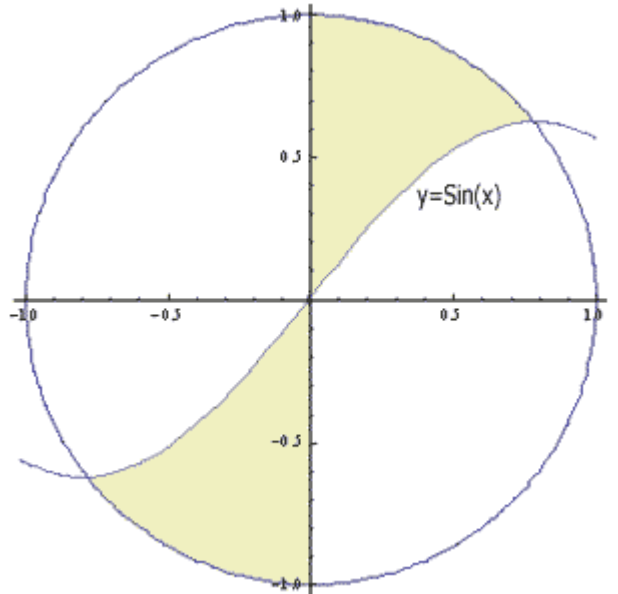


Если  $l>r$ , то точка лежит вне окружности.  
Если  $l<r$ , то точка — внутри круга.

```
readln(x,y);
l:=sqrt(x*x+y*y);
if l<1 then writeln('Да, попадает')
  else writeln('Нет');
```

4. Для закрашенной области программный код имеет вид:

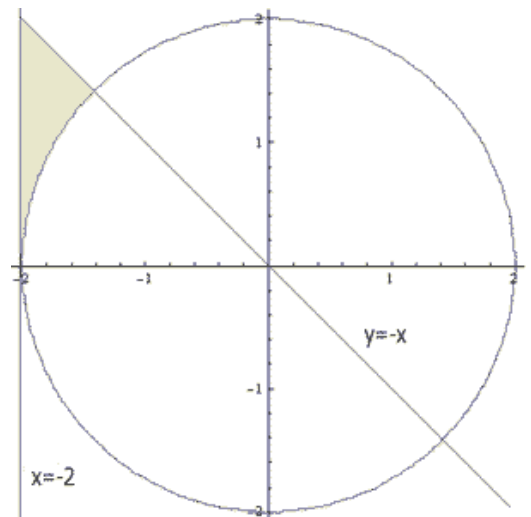
```
readln(x,y);
l:=sqrt(x*x+y*y);
if (l<1) and (y>x*x)
  then writeln('Да, попадает')
  else writeln('Нет');
```



5. Использование связок `and`, `or` весьма эффективно:

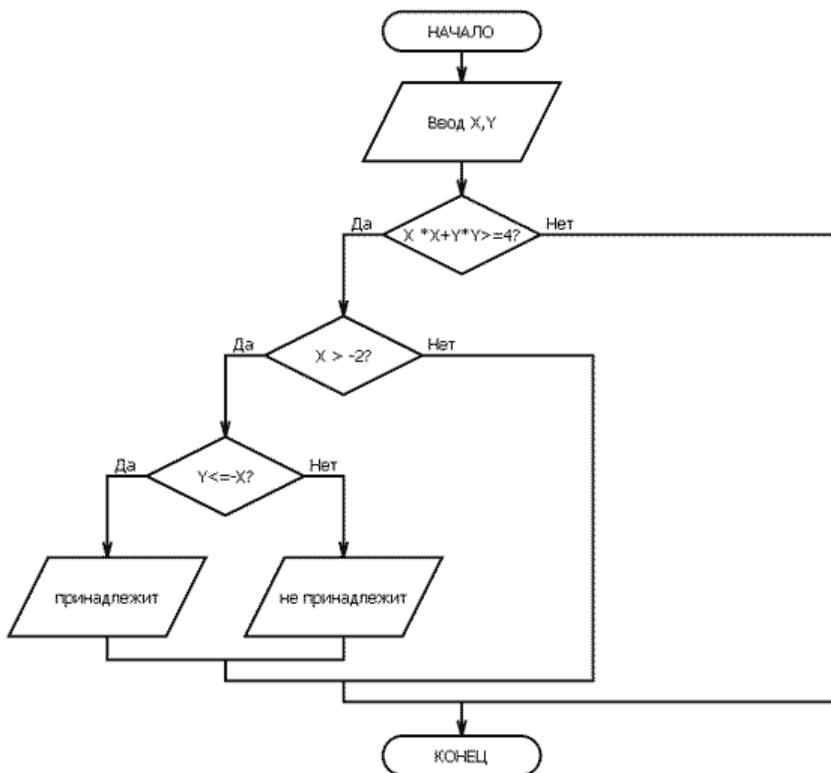
```
readln(x,y);
l:=sqrt(x*x+y*y);
if (l<1) and ((x>0) and (y>Sin(x)) or
  (x<0) and (y<Sin(x)))
  then writeln('Да, попадает')
  else writeln('Нет');
```

**Теперь перейдем к рассмотрению собственно графического варианта задачи.**



6. Требуется написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости ( $x, y$  — действительные числа) и определяется принадлежность этой точки заданной заштрихованной области (включая границы).

**Блок-схема.**



```

var x,y:real;
begin

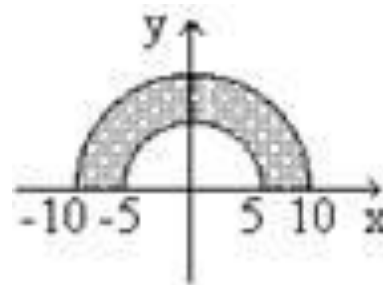
```

```

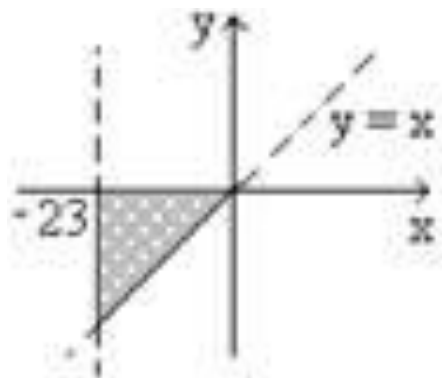
  readln(x,y);
  if (x*x+y*y>=4) and(x>=-2)
    and (y<=-x) and (y>=0)
  then write('принадлежит')
  else write('не принадлежит')
  end.

```

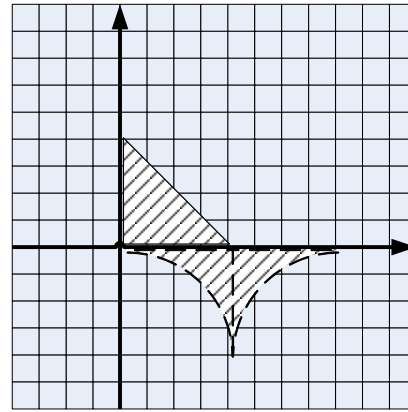
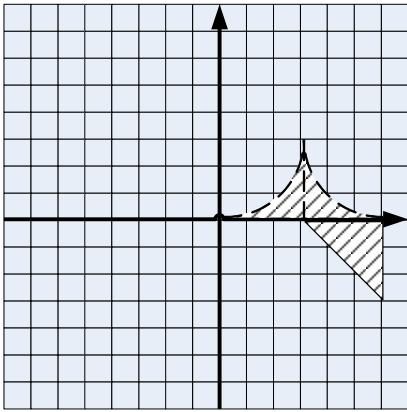
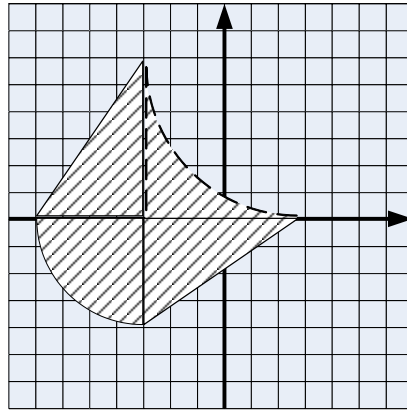
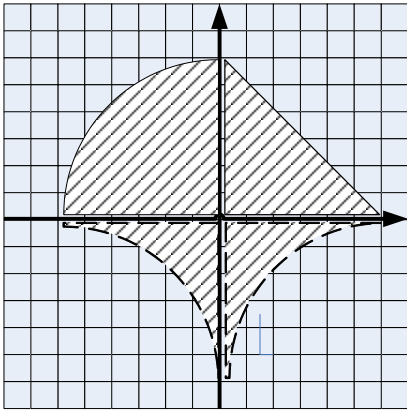
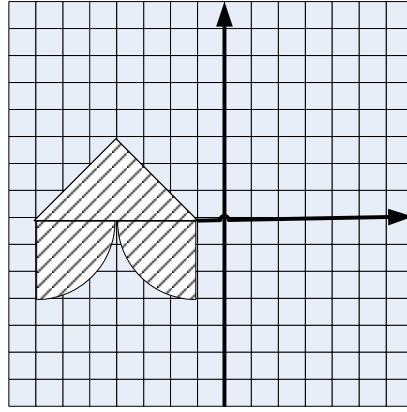
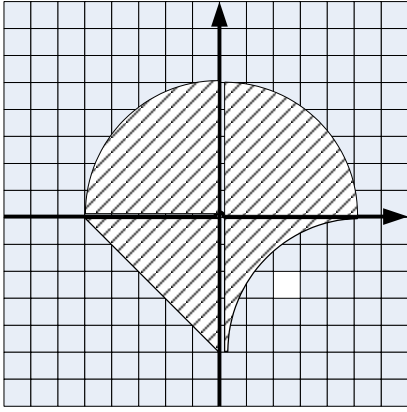
7. Напишите программу: если  $(5 < (x^2 + y^2)^{1/2} < 10$  и  $y > 0$ ), значит точка лежит внутри заштрихованной области? Проверьте

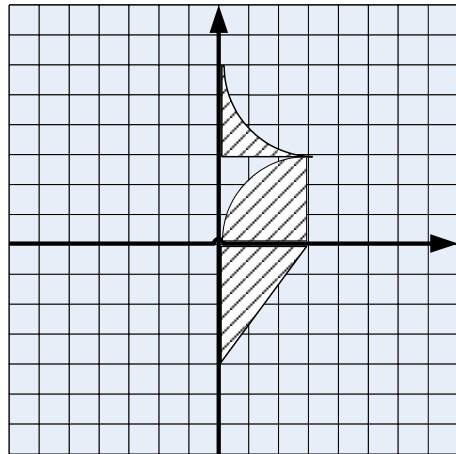
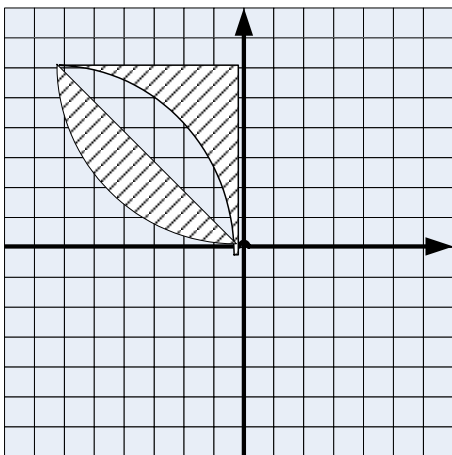
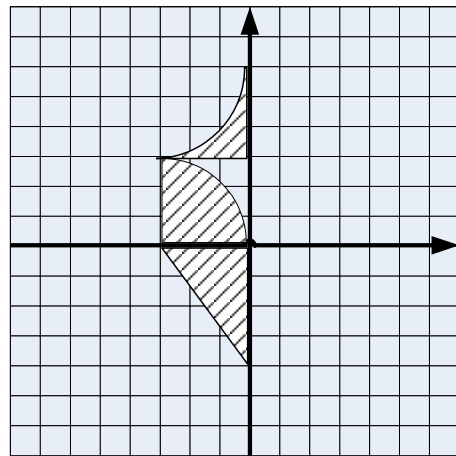
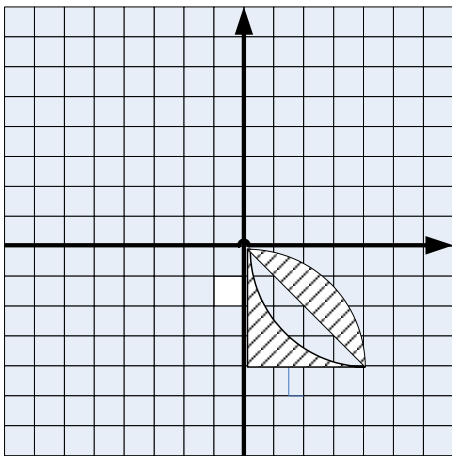
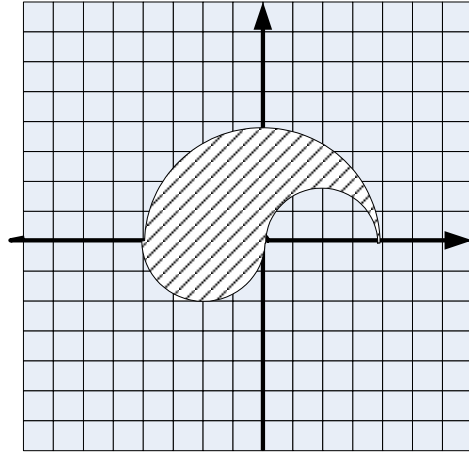
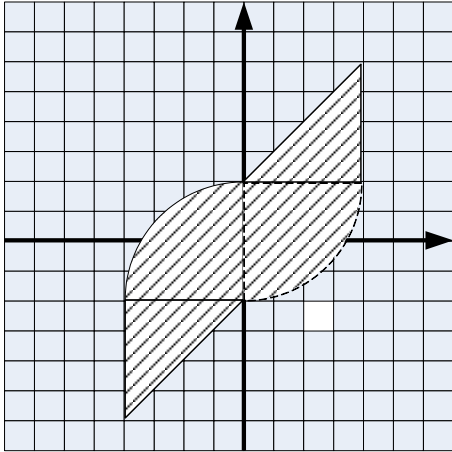


8. Дана точка на плоскости с координатами  $(x, y)$ . Составить алгоритм, который выдает сообщение "Да" если точка лежит внутри заштрихованной области, "Нет" – если точка лежит вне заштрихованной области, "На границе" – если точка лежит на границе заштрихованной области.



Если  $(x < 0$  и  $y < 0$  и  $23 - (23 - |x|) > |y|$ ), то лежит внутри заштрихованной области? Проверьте.





## Практическое занятие № 19.

### Вычисление сложной функции с помощью Оператора выбора.

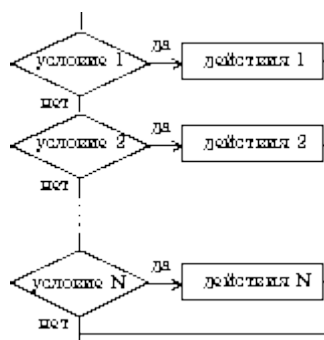
**Цель:** работа с базовой структурой Ветвления, уметь создавать, вводить в компьютер, выполнять и исправлять программы, реализующие Оператор выбора.

Структура ветвление существует в четырех основных вариантах, два из которых мы рассмотрели ранее. Теперь познакомимся с Оператором выбора. Если необходимо проверять достаточно много условий и в зависимости от них выполнять те или иные действия, используют оператор выбора Case .

выбор

выбор

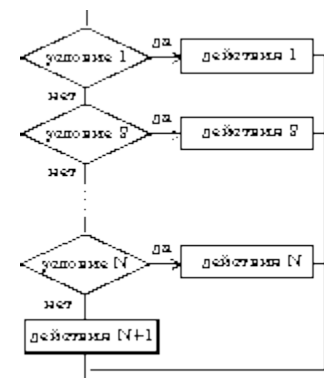
при условии 1: действия 1  
при условии 2: действия 2  
.....  
при условии N: действия N



выбор—иначе

выбор

при условии 1: действия 1  
при условии 2: действия 2  
.....  
при условии N: действия N  
иначе действия N+1



#### Формат Оператора выбора Case:

```
Case K of  
A1 : <оператор 1>;  
A2 : <оператор 2>;  
...  
AN : <оператор N>  
else <оператор N+1>  
end;
```

```
Case K of  
A1 : <оператор 1>;  
A2 : <оператор 2>;  
...  
AN : <оператор N>  
end;
```

где:1) Case ("случай") - служебное слово;

2) K - переменная, называемая селектором или переключателем;



3) A1, ... , AN - метки ветвей, которые отличаются по смыслу от меток, описываемых в разделе Label;

Отметим, что переключатель и метки должны быть одного и того же простого порядкового типа (целый, символьный, логический). Оператор Case передает управление тому оператору, с одной из меток которого совпало значение селектора K. Если значение селектора K не совпало ни с одной из меток, то исполняется оператор, следующий за словом else.

**Задание:** Используя оператор Case, составьте программу вычисления значения функции y:

$$y = \begin{cases} a+bx+cx^2, & \text{если } 1 \leq x < 2 \\ \sin(xb), & \text{если } 2 \leq x < 3 \\ a+bx^3+c, & \text{если } 3 \leq x < 4 \\ \ln|b+c/2x|, & \text{если } 4 \leq x < 5 \\ e^{\text{asin}x} + c, & \text{если } 5 \leq x < 6 \end{cases}$$

Оператор выбора работает следующим образом. Сначала вычисляется значение выражения, стоящее после зарезервированного слова case, а затем выполняется оператор (или составной оператор), соответствующий результату вычисления выражения.

Может случиться, что в списке выбора не окажется константы равной вычисленному значению ключа. В этом случае управление передается оператору, стоящему за словом ELSE.

Например,

```
case NUMBER mod 2 of
  0 : writeln (NUMBER, '- число четное');
else : writeln (NUMBER, '- число нечетное');
end;
```

Если один оператор выполняется при нескольких значениях, то их можно перечислить через запятую.

```
case MONTH of
  1, 2, 3 : writeln ('Первый квартал');
  4, 5, 6 : writeln ('Второй квартал');
  7, 8, 9 : writeln ('Третий квартал');
  10, 11, 12 : writeln ('Четвёртый квартал');
end;
```

Оператором может являться не только простой оператор, но также составной и пустой операторы.

```
case CODE of
  1 : for i := 1 to 5 do
      writeln ('*****');
  2 : begin {составной оператор}
      x:=sqr(y-1);
      writeln (x);
      end;
  3 : {пустой оператор}
```

end;

Любому заданному значению селектора соответствует лишь один вход в списке операторов. Константы должны принадлежать тому же типу, что и селектор. Если селектор принимает значение, которому не соответствует ни один вход, то будет выполняться оператор, следующий за словом else. Если же этого оператора нет, то никакие альтернативы не будут выполняться.

Если оператор должен выполняться при нескольких значениях селектора следующих друг за другом, образуя некоторый промежуток, то это можно записать в более сжатой форме.

Например,

```
case Chislo of
  0..9 : write ('Это число является цифрой');
```

Посмотрите, в каких вариантах еще можно использовать оператор выбора при решении задачи.

**Пример1:** Напишите программу, переводящую школьные отметки в оценки.

```
var BALL : Integer;
```

Begin

```
Write ('Введите величину отметки: ');
```

```
Read (BALL);
```

```
WriteLn;
```

```
Case BALL of { Перевод отметки в оценку }
```

```
1: WriteLn ('Очень плохо...');
```

```
2: WriteLn ('Плохо...');
```

```
3: WriteLn ('Удовлетворительно...');
```

```
4: WriteLn ('Хорошо!');
```

```
5: WriteLn ('Отлично!')
```

End.

**Пример2:** Напишите программу, которая для каждой введенной цифры (0-9) выводит соответствующее ей название.

```
Var
```

```
a, b, c : integer;
```

Begin

```
writeln('Введите цифру ');
```

```
readln(a);
```

```
case a of
```






```

0 : writeln ('ноль');
1 : writeln ('один');
2 : writeln ('два');
3 : writeln ('три');
4 : writeln ('четыре');
5 : writeln ('пять');
6 : writeln ('шесть');
7 : writeln ('семь');
8 : writeln ('восемь');
9 : writeln ('девять')
else writeln ('Это число не является цифрой');
end;
readln;
End.

```

**Пример3:** Напишите программу, рассчитывающую стоимость билета на пригородном поезде зависит от дальности поездки и рассчитывается по правилу:

Стоимость = Тариф\*Расстояние

Зоны	Расстояние	Тариф (рублей)
1		0.5
2		1,45
3		3,01
4	 	5.12
	50      80      110      140      ∞	

По введенному расстоянию появляется номер зоны и стоимость проезда

**Задания:**

1. Написать программу, позволяющую по последней цифре числа определить последнюю цифру его квадрата.
2. Пусть элементами круга являются радиус (первый элемент), диаметр (второй элемент) и длина окружности (третий элемент). Составить программу, которая по номеру элемента запрашивала бы его соответствующее значение и вычисляла бы площадь круга.
3. Решить квадратное уравнение с помощью оператора выбора

## Практическое занятие № 20.

### Табулирование функций с помощью цикла с предусловием.

**Цель:** уметь составлять таблицы значений функций.

**Формат цикла с предусловием:**

**While** <условие> **do** <тело цикла>;

Цикл выполняется пока условие истинно (верно), если условие сразу не верно – цикл не выполняется ни разу.

**Пример:** Необходимо, используя цикл с предусловием составить таблицу значений функций  $y=f(x)$  на отрезке от  $a$  до  $b$ , с шагом  $h$ .

$$f(x) = 3x-2 * \lg x$$

$$a=0,5; b=5; h=0,25.$$

Вспомним, что  $\lg(a)=\ln(a)/\ln(10)$

```
var a,b,h,x:real;
begin
  readln(a,b,h);
  x:=a;
  while x<=b do
  begin
    writeln(x:6:2,(3*x-2+ln(x)/ln(10)):6:2);
    x:=x+h;
  end;
end.
```

**Пример:** протабулировать функцию  $y = ax^2 - b$  для  $x$  изменяющегося от  $-7$  до  $+12$  с шагом  $+1$ .

```
var a, b, x, y: integer;
begin
  write ('a= '); readln (a);
  write ('b= '); readln (b);
  x:= -7;
  while x<= 12 do
  begin
    y:= a*x*x - b;
    writeln (x:3, y:10);
    x:= x+1;
  end;
end.
```

**Задание:**

1. Вычислить 9 значений функции  $y = 8x^3 - 7x + 8$  для  $x$  задаваемого с клавиатуры.
2. Вычислить  $y = 4x^2 + 3x$ ;  $x$  меняется от  $-3$  до  $6$  с шагом  $0.5$ .
3. Протабулировать функцию  $y = 6x^2 - 3\sin x$  на интервале от  $x_1$  до  $x_2$  с шагом  $s$ .

4. Вычислить  $\begin{cases} 5 \cos x, & \text{если } x > 0 \\ 7x, & \text{если } x \leq 0 \end{cases}$ ;  $x$  меняется от  $-5$  до  $5$  с шагом  $\frac{1}{2}$ .

### Практическое занятие № 21.

#### Аналитический расчет результатов выполнения циклических программ.

**Цель:** научиться рассчитывать результаты выполнения циклических программ.

1. Что выведется на экран в результате выполнения программы:

```
var i, a, c: integer;      a) 12, 3
begin c:= 3; a:= 0;      б) 1, 12
for i:= 2 to 4 do        в) 3, 6
c:= c+ i;                г) 1, 3
a:= a+ 1;                д) 1, 7
writeln (a, c);          е) другое
end.
```

2. Чему будут равны значения переменных a, b после выполнения программы:

```
var a, b, c: integer;      a) 1, -1
begin                      б) 5, -1
for c:= 1 do 5 do begin    в) 0, 0
a:= a+ 1;                  г) 5, -5
b:= b- 1;                  д) 5, 0
end;                        е) другому
end.
```

3. Что вычисляет данная программа:

```
var x, k: integer;          а) сумму
begin k:= 0;                б) количество
for x:= 5 to 10 do k:= k+ x; в) произведения
writeln (k);                г) максимальное значение
end.                          д) минимальное значение
                               е) другое
```

4. Что выведется на экран в результате выполнения программы:

```
var i, a, c: integer;      a) 12, 3
begin c:= 3; a:= 0;      б) 1, 12
for i:= 6 to 4 do        в) 3, 6
c:= c+ i;                г) 0, 3
a:= a+ 1;                д) 1, 7
writeln (a, c);          е) другое
end.
```

5. Что находит данная программа:

var x, k, a: integer;	
begin readln (a); k:= a;	а) сумму
for x:= 2 to 10 do begin	б) количество
readln (a);	в) произведения
if a> k then k:= a; end;	г) максимальное значение
writeln (k);	д) минимальное значение
end.	е) другое

6. Чему будут равны значения переменных a, b после выполнения программы:

var a, b, c: integer;	а) 1, -1
begin	б) 5, -1
for c:= 1 do 5 do	в) 0, 0
a:= a+ 1;	г) 5, -5
b:= b- 1;	д) 5, 0
end.	е) другому

7. Что находит данная программа:

var i, a, s: integer;	
begin s:= 0;	а) сумму
for i:= 1 to 10 do begin	б) количество
readln (a);	в) произведения
if a< 5 then s:= s+ 1;	г) максимальное значение
end;	д) минимальное значение
writeln (s);	е) другое
end.	

8. Сколько раз будет выполняться цикл:

	а) 1
	б) 0
for f:=3 to 7 do a:= a+ 2;	в) 4
for u:=3 to 2 do r:= r+ u;	г) 5
	д) 2

9. Чему будет равно значение переменной y после выполнения программы:

var x, y: integer;	а) 4
begin y:= 0;	б) 14
for x:= 2 to 5 do y:= y* x;	в) 120
writeln (y);	г) 15
end.	д) 0
	е) другому

11. Определить значение переменной s после выполнения следующих операторов:

```
s:= 1; n:= 3;
for i:= 2 to n do s:= s+ i* 2;
```

- a) 3; б) 5; в) 4; г) 6; д) 11.
12. Чему будет равно значение переменной S после выполнения фрагмента программы:  
 y:= 1; s:= 0;  
 while y<= 2 do begin s:= s+ y; y:=y+ 1 end;
- a) 1; б) 2; в) 3; г) 4.
13. Выберите фрагмент программы, который вычисляет значение функции  $y = \frac{1}{x+1}$ , если аргумент x изменяется от 0 до 6 с шагом 1,5:  
 а) x:= 0;  
 while x<= 6 do begin y:= 1/ (1+ x); x:= x+ 1.5 end;  
 б) x:= 0;  
 if x<= 6 then begin y:= 1/ (1+ x); x:= x+ 1.5 end;  
 в) while x<= 6 do begin x:= 0; y:= 1/ (1+ x); x:= x+ 1.5 end;  
 г) x:= 0;  
 while x<= 6 do y:= 1/ (1+ x); x:= x+ 1.5;
14. Какие из приведенных ниже программ содержат ошибку, приводящую к “зацикливанию”:  
 а) i:= 0; for m:= 10 to 0 do i:= i+ 1; writeln (i);  
 б) i:= 0; repeat i:= i+ 1 until i= 0; writeln (i);  
 в) i:= 0; while (i= 0) or (i< 10) do i:= i+ 1; writeln (i);  
 г) i:= 0; repeat i:= i+ 1 until i< 0; writeln (i);  
 д) i:= 0; while (i= 0) or (i> 0) do i:= i+ 1; writeln (i);
15. Сколько раз будут выполнены операторы внутри цикла в следующем фрагменте:  
 x:= 2; y:=10;  
 while y- x> 1 do begin  
 x:= x+ 1; y:= y- 1; end;  
 sum:= x+ y;
- a) 2; б) 3; в) 4; г) 5 д) 6.
16. Что будет выведено на экран после выполнения следующих операторов:  
 p:= 1; I:= 1;  
 repeat p:= p\* 2;  
 I:= I- 1  
 until I<= 1; writeln (p);
- a) 2; б) 4; в) 1; г) 8.

## Практическое занятие № 22. Построение графика функции.

**Цель:** уметь строить графики функций в среде Turbo Pascal.

Пример программы построения графика функции

$$f(x)=\sin x+\sin 2x+\sin 3x-1-\cos x-\cos 2x$$

```

program graphic;
uses graph;
var driver, mode, errorcode : integer; xm,ym,i,j : integer;
pi,pi300,x1,y1,x2,y2, sc : real;
st1,st2,st3 : string;

```

```

function f(x:real) : real;
begin
f:=sin(x)+sin(2*x)+sin(3*x)-1-cos(x)-cos(2*x);
end; {графика}

begin
st1:='x';st2:='y';
st3:=' Press ENTER';
sc:=50;
driver:=9; {egavga}
mode:=2; {640x480 пикселей}
initgraph(driver,mode,'d:\bp\bgi'); {инициализация графического режима }
errorcode:=graphresult;
if errorcode<>grok then {ошибка }
begin
writeln('Error init Graph');
closegraph;
halt;
end;

xm:=getmaxx div 2;
ym:=getmaxy div 2;
{ xm=320;ym=240;центр экрана }
line(xm,20,xm,460);{ось y}
line(20,ym,620,ym);{ось x}
outtextxy(630,ym,st1); {маркировка оси x}
outtextxy(xm,10,st2); {маркировка оси y}
pi:=3.1415926; pi300:=pi/300;
x1:=-pi;
for i:=0 to 24 do {разметка оси x вертикальными черточками}
begin
line(xm+round(80*x1),230,xm+round(80*x1),250);
x1:=x1+pi300*25;
end;
x1:=-pi; {собственно построение графика отрезками прямых}
while x1<pi do
begin

```



```

y1:=f(x1);x2:=x1+pi300;
y2:=f(x2);
line(xm+round(80*x1), ym-round(sc*y1),
xm+round(80*x2), ym-round(sc*y2));
x1:=x2;
end;
outtextxy(270,470,st3);
readln;
closegraph;
end.

```

Программа компилируется и выдает график функции.

### Практическое занятие № 23.

**Результат исполнения циклической программы в графическом режиме.**

**Цель:** применение циклов в графике.

**Задача:** Получить изображение, состоящее из одной повторяющейся геометрической фигуры.

*Алгоритм решения задачи:*

Повторяющиеся элементы изображения в сумме представляют собой орнамент. Используя правила переноса, можно создавать как простые, так и достаточно сложные орнаменты.

Количество повторений базовой фигуры определяется переменной цикла, для которой задаются начальное и конечное значения, а также шаг (значение сдвига по оси  $Ox$  и/или по оси  $Oy$ ).

Если перенос базовой фигуры осуществляется в одном направлении (например, по оси  $Ox$ ), то достаточно использовать одну циклическую конструкцию. Переменной цикла может служить координата  $x$  или  $y$ , которая изменяется от начального значения до конечного с заданным шагом (обычно равным ширине или высоте базовой фигуры).

Если перенос базовой фигуры производится по обеим осям (по горизонтали и вертикали), то используется два цикла, один из которых вложен в другой. Например, внутренний цикл прорисовывает каждую строку (изменяется только координата  $x$ ), а внешний осуществляет переход на новую строку (изменяется координата  $y$ ).

*Программа на языке Паскаль:*

```

uses graph,crt;
var
  driver,mode:integer;
  x,y:integer;
begin

```

```

driver:= detect;
initgraph(driver,mode,' ');
randomize;
y := 0;
repeat
  x := 0;
  repeat
    setcolor(10);
    moveto(x,y+50);lineto(x+30,y);lineto(x+30,y+50);
    setcolor(random(10)+6);
    circle(x+30,y+50,20);
    x := x + 60;
  until x > 800;
  y := y + 100;
until y > 600;
readln
end.

```

**Задача:** Изменить размер геометрической фигуры.

*Описание переменных:*

$x_1, y_1$  – координаты верхнего левого угла прямоугольника;

$x_2, y_2$  – координаты нижнего правого угла прямоугольника;

$a, b$  – коэффициенты масштабирования.

*Алгоритм решения задачи:*

Геометрический объект можно масштабировать, т.е. увеличивать или уменьшать по осям координат. Масштабирование может производиться относительно разных точек пространства. Например, относительно начала координат или относительно центра объекта.

При изменении размера фигуры используются коэффициенты масштабирования. Если коэффициент больше единицы, то объект увеличивается. Если коэффициент принадлежит диапазону от 0 до 1, то объект будет уменьшен.

Новые координаты  $X$  и  $Y$  определяются соотношением

$$X = k_x * x;$$

$$Y = k_y * y,$$

где  $k_x$  и  $k_y$  – коэффициенты.

*Программа на языке Паскаль:*

```

uses graph,crt;
var
  driver,mode,i:integer;
  x1,y1,x2,y2:integer;
  a,b:real;
begin
  driver:= detect;
  initgraph(driver,mode,"");

```

```

setcolor(10);
x1:= 100;
y1:= 100;
x2:= 200;
y2:= 150;
rectangle(x1,y1,x2,y2);
a:= 1.1;
b:= 0.9;
for i:= 1 to 5 do begin
    delay(1000);
    setcolor(0);
    rectangle(x1,y1,x2,y2);
    x2:= round(x2*a);
    y2:= round(y2*a);
    setcolor(10);
    rectangle(x1,y1,x2,y2);
end;
for i:= 1 to 5 do begin
    delay(1000);
    setcolor(0);
    rectangle(x1,y1,x2,y2);
    x2:= round(x2*b);
    y2:= round(y2*b);
    setcolor(10);
    rectangle(x1,y1,x2,y2);
end;
readln
end.

```

*Примечания:*

Т.к. координаты верхнего левого угла не меняются, то прямоугольник увеличивается относительно именно этой точки.

## **Практическое занятие № 24.**

### **Ввод и вывод элементов массива. Формирование элементов массива.**

**Цель:** научиться программировать линейные массивы.

Предположим, что программа работает с большим количеством однотипных данных. Скажем около ста разных целых чисел нужно обработать, выполнив над ними те или иные вычисления. Как вы себе представляете 100 переменных в программе? И для каждой переменной нужно написать одно и тоже выражение вычисления значения? Это очень неэффективно.

Есть более простое решение. Это использование такого типа данных как **массив**. Массив представляет собой последовательность ячеек памяти, в которых хранятся однотипные данные. При этом существует всего одно имя переменной связанной с массивом, а обращение к конкретной ячейке происходит по ее индексу (номеру) в массиве.

Нужно четко понимать, что индекс ячейки массива не является ее содержимым. Содержимым являются хранимые в ячейках данные, а индексы только указывают на них. Действия в программе над массивом осуществляются путем использования имени переменной, связанной с областью данных, отведенной под массив.

Итак, *массив* – это именованная группа однотипных данных, хранящихся в последовательных ячейках памяти. Каждая ячейка содержит элемент массива. Элементы нумеруются по порядку, но необязательно начиная с единицы (*хотя в языке программирования Pascal чаще всего именно с нее*). Порядковый номер элемента массива называется индексом этого элемента.

Помним, **все элементы определенного массива имеют один и тот же тип**. У разных массивов типы данных могут различаться. Например, один массив может состоять из чисел типа **integer**, а другой – из чисел типа **real**.

**Описание массива** определяет имя, размер массива и базовый тип.

Формат описания в разделе переменных:

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

Индексы элементов массива обычно целые числа, однако могут быть и символами, а также описываться другими порядковыми типами.

Массив можно создать несколькими способами.

Обращение к определенному элементу массива осуществляется путем указания имени переменной массива и в квадратных скобках индекса элемента.

**Линейный (одномерный) массив** – массив, у которого элементы – простые переменные. *Например:*

**Var** B: **array** [0..5] **of** real;

**Var** M: **array** [1..25] **of** char;

**Var** C: **array** ['A'..'Z'] **of** integer;

**Ввод и вывод массива** производится поэлементно. Обычно для этого используется цикл с параметром, где в качестве параметра применяется индексная переменная.

*Например:* В программе вводится десять значений целочисленного массива A и выводятся значения вещественного массива B, содержащего 50 элементов:

**Var** A: **array** [1..10] **of** integer; B: **array** [1..50] **of** real;

i : integer;

Begin for i:= 1 to 10 do

begin write ('A[' ,i,'] ='); readln(A[i]) end;

```
.....  
for i:= 1 to 50 do  
begin writeln (' B[' ,i,'] = ' , B[i]) end;
```

end.

**Пример:** Требуется заполнить массив числами, которые вводит пользователь, и вычислить их сумму. Если пользователь вводит ноль или превышен размер массива, то запросы на ввод должны прекратиться.

Для решения подобного рода задач иногда бывает уместно использовать цикл с постусловием (repeat).

```
const n = 10;  
var  
  arr: array[1..n] of integer;  
  sum: integer;  
  i: byte;  
begin  
  i := 1;  
  sum := 0;  
  repeat  
    write ('Число: ');  
    readln(arr[i]);  
    sum := sum + arr[i];  
    i := i + 1  
  until (arr[i-1] = 0) or (i > n);  
  for i := 1 to n do  
    write (arr[i], ' ');  
  writeln;  
  writeln ('sum = ', sum);  
readln  
end.
```

## Практическое занятие № 25.

### Формирование массива с помощью генератора случайных чисел и поиск наибольшего или наименьшего элемента.

**Цель:** научиться формировать массив с помощью функции random.

В задачах по программированию очень часто встречается необходимость заполнить массив данными и вывести их потом на экран. Методов этого не так уж много и особым разнообразием они не отличаются. Поэтому хорошо бы их оформить в виде процедур и использовать по мере необходимости.

Создадим две процедуры заполнения одномерного массива целыми числами: одну с использованием функции random и одну, где пользователь самостоятельно задает значения. Также создадим процедуру вывода массива на экран. Затем вызовем в основной ветке программы эти процедуры, передав им в качестве аргументов переменные, связанные с массивами.

```

const
  n = 10;
type
  miniarr = array[1..n] of integer;
var
  a, b: miniarr;
procedure arr_rand (k:integer; var arr: miniarr);
  var i: byte;
  begin
    writeln ('Заполнение массива случайными числами. ');
    randomize;
    for i := 1 to k do
      arr[i] := random (100);
    end;
procedure arr_user (k:integer; var arr: miniarr);
  var i: byte;
  begin
    write ('Введите ', k, ' чисел через пробел: ');
    for i := 1 to k do
      read (arr[i]);
    end;
procedure arr_out (k:integer; var arr: miniarr);
  var i: byte;
  begin
    write ('Вывод массива: ');
    for i := 1 to k do
      write (arr[i]:4);
    writeln;
  end;
begin
  arr_rand (n, a);
  arr_out (n, a);
  writeln;
  arr_user (n, b);
  arr_out (n, b);
readln
end.

```

**Задание:** Заполнить случайными числами из диапазона [0,1] вещественный линейный массив из N чисел. Найти максимальное значение и его индекс (первый, если таких значений несколько).

Простой массив является **одномерным**. Он представляет собой линейную структуру.

```

var ch: array [1..11] of char;
  h: char;
  i: integer;

begin
  for i := 1 to 11 do read (ch[i]);

  for i := 1 to 11 do write (ch[i]:3);

```

```
readln  
end.
```

В примере выделяется область памяти под массив из 11 символов. Их индексы от 1 до 11. В процессе выполнения программы пользователь вводит 11 любых символов (например, 'q', 'w', 'e', '2', 't', '9', 'u', 'I', 'I', 'o', 'p'), которые записываются в ячейки массива. Текущее значение переменной *i* в цикле **for** используется в качестве индекса массива. Второй цикл **for** отвечает за вывод элементов массива на экран.

**Пример:** Программа нахождения индекса минимального элемента в массиве.

```
Var  
  k, m, day: integer;  
  Dat: array[1..7] of integer;  
Begin  
  Dat[1]:=7; Dat[2]:=9;  
  Dat[3]:=10; Dat[4]:=8;  
  Dat[5]:=6; Dat[6]:=7;  
  Dat[7]:=6;  
  day:= 1;  
  m:=Dat[1];  
  for k:=2 to 7 do begin  
    if Dat[k] < m then begin  
      m:=Dat[k];  
      day:=k  
    end  
  end;  
  write(day);  
  readln;  
End.
```

Сперва массив *Dat* заполняется целыми числами  $Dat = [7, 9, 10, 8, 6, 7, 6]$ . Затем предполагают, что минимальным элементом является первый элемент массива и сохраняют его значение в переменной *m* ( $m=7$ ), а индекс этого элемента записывают в переменную *day* ( $day=1$ ).

После этого циклом просматривают оставшиеся элементы массива и сравнивают их с текущим минимальным элементом. Если элемент массива больше или равен минимальному, то никакие операции не выполняются, лишь переходят к следующему элементу. Если же рассматриваемый элемент массива меньше значения переменной *m* (т.е. минимума), то в *m* записывают новый минимальный элемент, а в *day* заносят индекс (позицию) этого элемента в массиве *Dat*.

**Пример.** *Определить, имеется ли в заданном целочисленном массиве  $A(N)$  хотя бы одна пара совпадающих по значению чисел.*

Номер теста	Проверяемый случай	Система тестов		
		N	Данные Массив A	Результат Otvet
1	Имеется	4	(1,3,2,3)	"Есть совпадающие числа"
2	Не имеется	3	(1,2,3)	"Нет совпадающих чисел"

```

Uses Crt;
Type Mas = Array [1..20] of Integer;
Var A : Mas;
    i, j, N : Integer;
    Flag : Boolean;

```

```

Procedure InputOutput; { Описание
процедуры ввода-вывода данных }
Begin ClrScr;
    Write('N = '); ReadLn(N);
    For i := 1 to N do
        begin Write('A[', i, '] = ');
ReadLn(A[i]) end;
    WriteLn; WriteLn('Массив A');
    For i := 1 to N do Write(A[i] : 4);
    WriteLn; WriteLn
End;

```

```

Procedure Search(Var A:Mas; Var
Flag:Boolean); { Описание
процедуры поиска решения }
Begin
    i:=1; Flag:= FALSE;
    While (i<=N-1) and not Flag do
{цикл по первому числу из пары}
        begin
            j:=i+1;
            While (j<=N) and not Flag do {цикл
по второму числу из пары}
                If A[i]=A[j] then Flag:=TRUE else
j:=j+1;
                i:=i+1
            end;
        End;
    BEGIN

```

```

    InputOutput; {вызов процедуры
ввода-вывода данных }
    Search(A, Flag); {вызов процедуры поиска решения задачи}
    WriteLn('О т в е т : ');
    If Flag then WriteLn('Есть совпадающие числа.')
        else WriteLn('Нет совпадающих чисел. ');
    ReadLn
END.

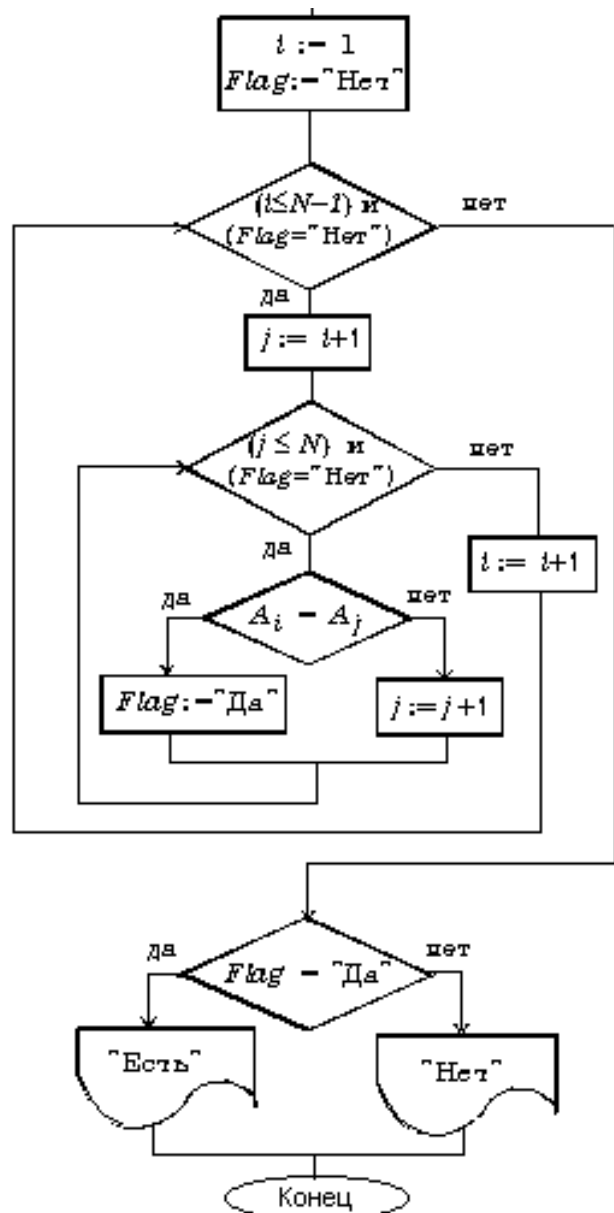
```

```

End; { of OutResult }
BEGIN
    InputOutput(A); {вызов процедуры ввода-вывода данных }
    Solution(A); {вызов процедуры поиска решения задачи}
    OutResult {вызов процедуры вывода результата }
END.

```

**Задание:** Дан целочисленный линейный массив. Отсортировать его элементы в порядке возрастания значений.





**Перечень рекомендуемой учебной литературы, информационных ресурсов сети Интернет соответствует пункту 3.2. рабочей программы учебной дисциплины «Информатика» специальностей**

**23.02.06           Техническая эксплуатация подвижного состава железных дорог**

**23.02.01           Организация перевозок и управление на транспорте (по видам)**

**13.02.07           Электроснабжение (по отраслям)**

**08.02.10           Строительство железных дорог, путь и путевое хозяйство**