

Департамент образования, науки и молодёжной политики Воронежской области
ГБПОУ ВО «Воронежский государственный профессионально-педагогический
колледж»

**Методические указания к практическим работам по дисциплине
«Основы алгоритмизации и программирования»**

Савченко Е.А., Польшников П.М., Желобкова А.А.

Учебно-методическое пособие

Рекомендовано

советом учебно-методического центра

в качестве учебно-методического пособия по дисциплине «Информационные
технологии» для студентов колледжа

специальности 44.02.01 Профессиональное обучение (по отраслям),

профиля подготовки «Компьютерные системы и комплексы»

Воронеж, 2017 г.

УДК 37.01

ББК 74.57

С 12

Рецензенты:

Дорохов С.В. руководитель структурного подразделения ИТ ГБПОУ ВО
«ВГППК»

Савченко Е.А., Польшников П.М., Желобкова А.А. Методические указания к практическим работам по дисциплине «Основы алгоритмизации и программирования»: учебно-методическое пособие для студентов СПО – Воронеж; ВГППК, 2017г.

Учебно-методическое пособие по дисциплине ОП.09 «Основы алгоритмизации и программирования» разработано на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования 051001 Профессиональное обучение (по отраслям) специальности профиля подготовки Компьютерные системы и комплексы.

Учебно-методическое пособие предназначено для студентов колледжа специальности 051001 Профессиональное обучение (по отраслям), профиля подготовки «Компьютерные системы и комплексы»

.Печатается по решению совета учебно-методического центра ГОБУ СПО ВО «Воронежский государственный профессионально-педагогический колледж»

© Е.А. Савченко, П.М. Польшников

ГОБУ СПО ВО «ВГППК», 2017

Оглавление

Введение.....	5
Практическая работа № 1	7
Практическая работа № 2, №3	21
Практическая работа № 4.....	33
Практическая работа № 5, №6	42
Практическая работа № 7	54
Практическая работа № 8, №9, № 10	62
Практическая работа №11	82
Практическая работа № 13	92
Практическая работа № 14.....	101
Практическая работа № 15, 16.....	120
Практическая работа №17	132
Практическая работа № 18, 19, 20.....	141

Введение

Внедрение новых информационных технологий во все сферы современной жизни привело к тому, что умение работать на компьютере является необходимым атрибутом профессиональной деятельности любого специалиста и во многом определяет уровень его востребованности в обществе.

К специалистам по компьютерным технологиям предъявляются особые требования: знание языков программирования и умение составлять компьютерные программы любой сложности.

Знакомство с языком программирования начинается с изучения алфавита и структуры языка программирования. Не зная теоретических основ нельзя научиться программировать.

При проведении работ студенты пользуются подробными инструкциями, в которых указаны: тема, цель работы, порядок выполнения работы.

Работа с данным учебно-методическим пособием способствует формированию у студентов общих и профессиональных компетенций определенных федеральным государственным образовательным стандартом для дисциплины «Основы алгоритмизации и программирования»:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, определять методы решения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Оценивать риски и принимать решения в нестандартных ситуациях.

ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности.

ОК 6. Работать в коллективе и команде, взаимодействовать с руководством, коллегами и социальными партнерами.

ОК 7. Ставить цели, мотивировать деятельность обучающихся, организовывать и контролировать их работу с принятием на себя ответственности за качество образовательного процесса.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Осуществлять профессиональную деятельность в условиях обновления ее целей, содержания, смены технологий.

ОК 10. Осуществлять профилактику травматизма, обеспечивать охрану жизни и здоровья обучающихся.

ОК 11. Строить профессиональную деятельность с соблюдением правовых норм ее регулирующих.

ОК 12. Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).

ПК 1.1. Определять цели и задачи, планировать занятия.

ПК 1.3. Проводить лабораторно-практические занятия в аудиториях, учебно-производственных мастерских и в организациях.

ПК 1.4. Организовывать все виды практики обучающихся в учебно-производственных мастерских и на производстве.

ПК 2.2. Определять цели и задачи, планировать деятельность по педагогическому сопровождению группы обучающихся.

ПК 2.3. Организовывать различные виды внеурочной деятельности и общения обучающихся.

ПК 2.4. Осуществлять педагогическую поддержку формирования и реализации обучающимися индивидуальных образовательных программ.

ПК 3.2. Систематизировать и оценивать педагогический опыт и образовательные технологии в области начального профессионального образования и профессиональной подготовки на основе изучения профессиональной литературы, самоанализа и анализа деятельности других педагогов.

Практическая работа № 1

Тема: Выражения в паскаль

Цель работы: Ознакомиться с системой программирования ABCPascal, получить основные навыки работы с ней, освоить приёмы создания, компиляции и исполнения программы, научиться создавать программы для вычисления выражений.

Ход работы:

Теоретическое обоснование:

Язык программирования Паскаль был разработан Норбертом Виртом в 1971 году. Швейцарский профессор Норберт Вирт создал язык Паскаль как учебный язык структурного программирования.

Наибольший успех в распространении этого языка обеспечили персональные компьютеры. Фирма BorlandInternational, Inc (США) разработала Систему программирования Турбо Паскаль для ПК. Турбо Паскаль – это не только язык и транслятор с него, но еще и операционная оболочка, обеспечивающая пользователю удобство работы. Турбо Паскаль вышел за рамки учебного предназначения и стал профессиональным языком.

По сути дела расхождения между алгоритмическим языком и языком Паскаль заключается в следующем: алгоритмический язык – русскоязычный, язык Паскаль – англоязычный. Синтаксис языка Паскаль определен строго и однозначно в отличие от сравнительно свободного синтаксиса алгоритмического языка.

Структура программы на языке Паскаль.

Программа состоит из заголовка программы, раздела описаний и раздела операторов. В конце программы ставится точка.

Program< имя программы>;		– заголовок программы
Label< раздел меток>;	}	– раздел описаний
Const< раздел констант>;		
Type< раздел типов >;		
Var< раздел переменных >;		
Procedure (Function) < раздел подпрограмм>;		
Begin	}	

действия;

– раздел операторов

End.

Заголовок программы начинается со слова **Program**, за которым следует произвольное имя, придуманное программистом.

Раздел описаний может содержать не все пункты.

Раздел операторов имеется в любой программе и является основным. Начало и конец раздела операторов отмечаются словами **Begin** (начало) и **End** (конец). Все команды в разделе операторов отделяются друг от друга точкой с запятой. В конце программы обязательно ставится точка.

Описание переменных.

Раздел описания переменных начинается со слова **Var**, за которым следует список переменных. Тип переменной указывается после двоеточия.

VAR имя переменной: тип;

В языке Паскаль используются следующие типы переменных:

INTEGER – целый (значениями являются целые числа)

REAL – вещественный (значениями являются целые и дробные числа)

CHAR – символьный (значениями являются символы, например ‘+’, ‘e’)

STRING – строковый (значениями являются строки символов, например ‘+*/+*+’, ‘мама’)

BOOLEAN – логический (принимает значения **TRUE** – истина и **FALSE** – ложь)

Пр. **Var** a:integer;

 b:real;

 c:string;

Описание констант.

CONST название константы = значение;

Пр. **Const**n=10;

M='+';

Константа – переменная, которая не меняет свое значение во время выполнения программы.

Команда ввода.

READ(названия переменных); – курсор после ввода остается в той же строке.или

READLN(названия переменных); – курсор после ввода переходит на следующую строку.

Пр. read(a,b);

readln(s,d);

Команда вывода.

WRITE(названия переменных или фраза); – курсор после вывода остается в той же строке.или

WRITELN(названия переменных или фраза); – курсор после вывода переходит на следующую строку.

Если переменная вещественного типа, то при выводе указывают формат вывода.

название переменной:0:количество цифр после запятой

Пр. write(a,d);

writeln('сумма=',s:0:2);

Арифметические операции в языке Паскаль.

Знак	Выражение	Типы операндов	Типы результатов	Операция
+	A+B	вещ., вещ. цел., цел. цел., вещ. вещ., цел.	вещественный целый вещественный вещественный	Сложение
-	A-B	вещ., вещ. цел., цел. цел., вещ. вещ., цел.	вещественный целый вещественный вещественный	Вычитание
*	A*B	вещ., вещ. цел., цел. цел., вещ. вещ., цел.	вещественный целый вещественный вещественный	Умножение
/	A/B	вещ., вещ. цел., цел. цел., вещ. вещ., цел.	вещественный вещественный вещественный вещественный	Вещественное деление
div	A div B	цел., цел.	целый	Целое деление
d	A mod B	цел., цел.	целый	Остаток от целого деления.

Стандартные функции языка Паскаль.

Обращение	Тип аргумента	Тип результата	Функция
Pi	----	вещественный	Число П. = 3,141592
Abs (x)	цел., вещ.	цел., вещ.	Модуль аргумента x
Sqr(x)	I,R	I, R	Квадрат x
Sqrt(x)	I,R	R	Корень квадрат. из x
Sin(x)	вещ.	вещественный	Синус x в радианах
Cos(x)	вещ.	вещественный	Косинус x (в радианах)
Exp(x)	цел., вещ.	вещественный	e ^x - экспонента
Ln (x)	цел., вещ.	вещественный	Натуральный логарифм x

В языке Паскаль нет стандартной операции возведение в степень, поэтому при возведении в вещественную степень пользуются формулой: $x^y = e^{y \ln x}$.

Пр. $x^{2.3} = \exp(2.3 * \ln(x))$

Если y – целое значение, то степень вычисляется через умножение, например: $x^3 = x * x * x$; большие степени следует вычислять умножением в цикле.

Работа в среде программирования PascalABC.

Загрузка ABC Паскаля

Загрузите ABC Паскаль с помощью ярлыка на рабочем столе или из главного меню ПУСК.

Краткое знакомство с интегрированной средой

После загрузки системы экран разделен на три части:

1. служебную область;
2. окно ввода алгоритма программы - рабочее окно;
3. окно ввода данных и вывода результатов программы.

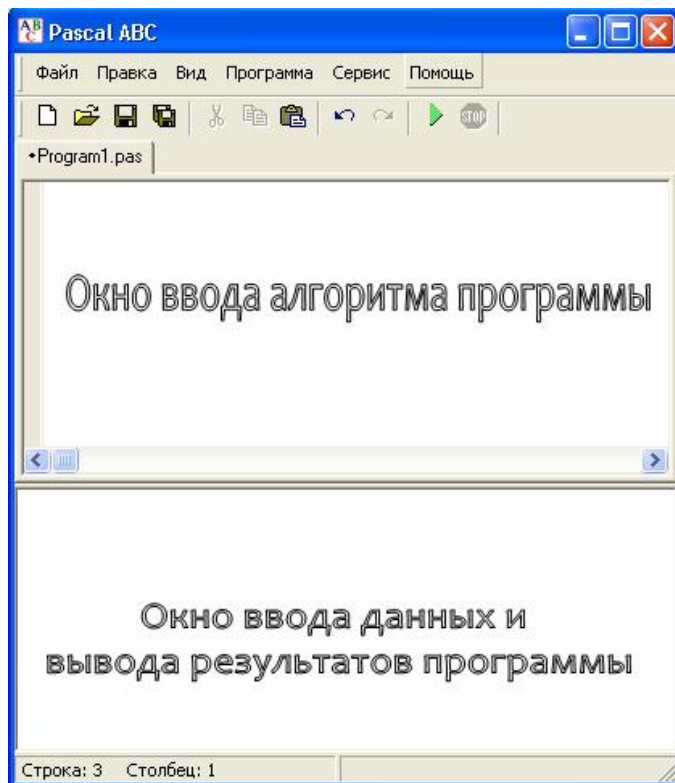


Рис. 1

Переход из основного окна в главное меню и обратно осуществляется посредством клавиши <F10>.

Первая программа

В рабочем окне редактора интегрированной среды наберем текст первой программы.

```
Program Myl_1 ;  
Var a, b, rez : Integer;  
Begin WriteLn ('Введите два числа через пробел');  
ReadLn (a, b);  
rez :=a*b;  
WriteLn ('Их произведение равно ', rez);  
WriteLn ('Нажмите <Enter>');  
ReadLn End.
```

Подсказка: можно выделить текст программы мышкой, нажать правую кнопку мыши и выбрать **Копировать**.

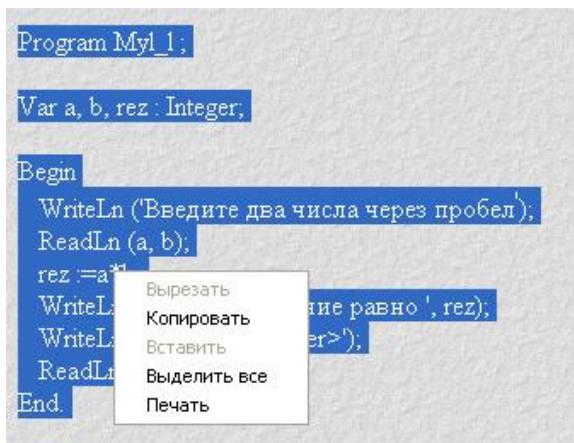


Рис. 2

Затем в системе **Pascal ABC** выбрать команды **Правка/Вставить**.

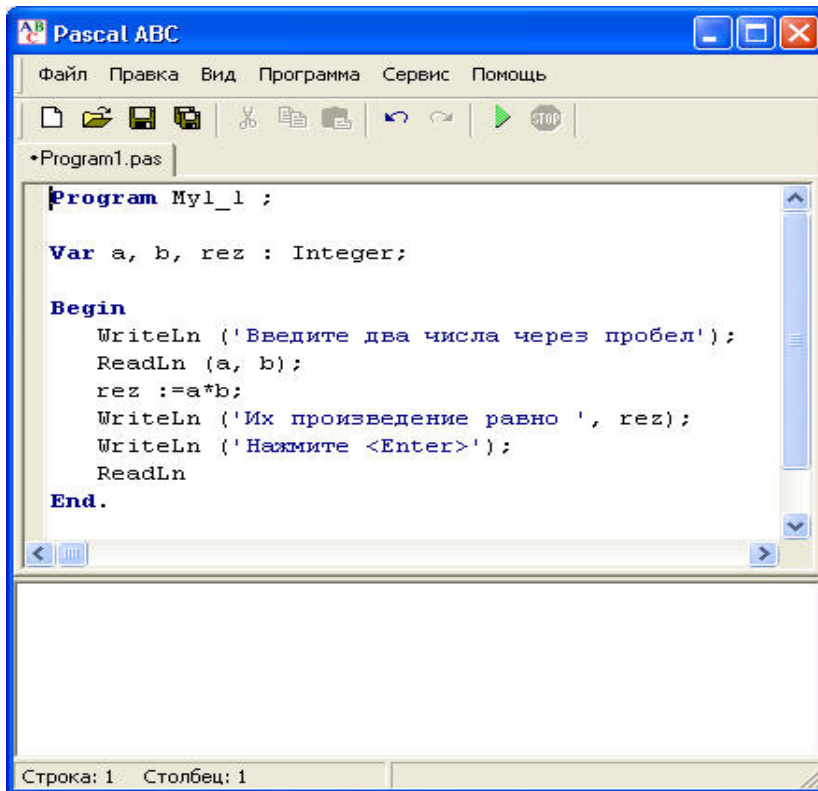


Рис. 3

В результате вы получите текст программы в рабочем окне.


Компиляция программы

Компилятор – специальная программа, с помощью которой исходный текст программы переводится в исполняемый двоичный файл программы.

Компиляция – процесс получения исполняемого двоичного файла из исходного текста программы, написанного на языке программирования.

Компиляция в Pascal-е происходит автоматически.

Запуск программы

Для того чтобы запустить программу, в главном меню выбираем пункт **Программа/Выполнить** или нажимаем значок . На экране в нижней части окна появляется сообщение:

Введите Два целых числа через пробел

Курсор находится в следующей строке. Вводим два целых числа через пробел и нажимаем <Enter>, после этого появляется сообщение:

Их произведение равно ...

нажмите <Enter>

Вместо точек будет выведено значение переменной rez, то есть число, равное произведению первого введенного числа на второе. Это сообщение останется на экране до тех пор, пока не будет нажата клавиша <Enter>.

Сохранение программы

Для того чтобы сохранить программу, необходимо:


- выйти в главное меню и выбрать режим **Файл**;
- в вертикальном меню выбрать пункт **Сохранить как...** ;
- в появившемся окне выбрать папку **Children**, затем выбрать свою папку (например, **Ivanov**), ввести имя файла (например, **prim1_1**) и нажать клавишу <Сохранить>.

Примечания.

1. В именах файлов нельзя употреблять следующие символы: *, =, +, [,], \, |, :, ., <, >, /, ?, символ пробела и буквы русского алфавита.
2. Для быстрого сохранения файла можно воспользоваться командами **Сохранить** или **Сохранить все** меню **Файл**.

Выход из системы программирования PascalABC

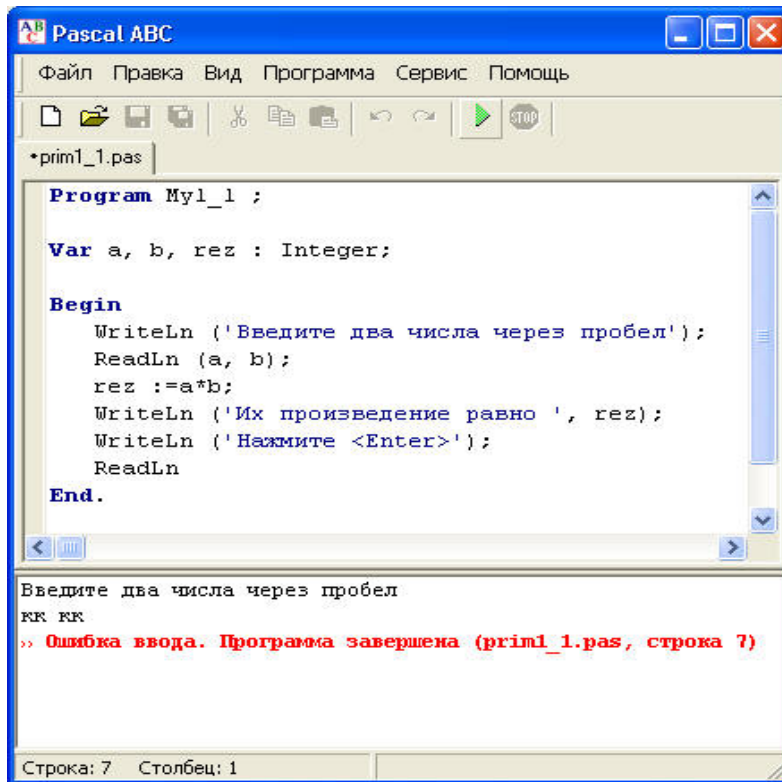
Для того чтобы закончить работу, необходимо:

- выйти в главное меню и выбрать пункт **Файл**;
- в вертикальном меню выбрать пункт **Выход**;
- или нажать на кнопку  в правом углу строки заголовка.

Эксперименты с программой

1. Введите в качестве исходных данных достаточно большие числа, например, 4567 и 789. Убедитесь, что у вас получается неправдоподобный результат — отрицательное число (—1117). Найдите экспериментальным путем тот интервал значений переменных a и b , когда результат умножения правильный.

2. Вместо числа введите какой-нибудь символ. Убедитесь, что компьютер выдает сообщение об ошибке.



```
Program My1_1 ;
Var a, b, rez : Integer;
Begin
  WriteLn ('Введите два числа через пробел');
  ReadLn (a, b);
  rez :=a*b;
  WriteLn ('Их произведение равно ', rez);
  WriteLn ('Нажмите <Enter>');
  ReadLn
End.
```

Введите два числа через пробел
kk kk
>> Ошибка ввода. Программа завершена (prim1_1.pas, строка 7)

Строка: 7 Столбец: 1

Рис. 4

3. Добавьте лишний знак апострофа в операторе WriteLn. Убедитесь, что программа не проходит компиляцию, а система сообщает об ошибке.

4. Измените оператор $rez :=a*b$ на $rez:=a*(a + b)$. Выполните алгоритм.



Примеры задач.

1. Даны стороны прямоугольника. Найти периметр и площадь.

```
Program prim;
```

```
Var a,b,p,s: real;
```

```
Begin
```

```
writeln('введите стороны прямоугольника');
```

```
readln (a,b);
```

```
p:=(a+b)*2;
```

```
s:=a*b;
writeln('периметр=',p:0:2);
writeln ('площадь=',s:0:2);
End.
```

2. Даны 2 числа. Найти их сумму и разность.

```
Program chisla;
Var a,b,p,s: real;
Begin
writeln('введите 2 числа');
readln (a,b);
p:=a-b;
s:=a+b;
writeln('разность=',p:0:2);
writeln ('сумма=',s:0:2);
End.
```

3. Даны 3 числа. Найти сумму их кубов.

```
Program chisla;
Var a,b,c,s: real;
Begin
writeln('введите 3 числа');
readln (a,b,c);
s:=a*sqr(a)+b*sqr(b)+c*sqr(c);
writeln('сумма кубов=',s:0:2);
End.
```

4. Дано целое число x . Вычислить $y = \sin(3x + 8)$, $z = \ln y$, $k = \frac{x^2 + 13}{e^{2x}}$,

$$m = \sqrt{x + y - z}, \quad n = x^3 - |k - 2|$$

```
Program chisla;
Var x:integer;
y,z,k,m,n: real;
Begin
writeln('введите целое число x');
```

```

readln (x);
y:=sin(3*x+8);  z:=ln(y);
k:=(sqr(x)+13)/exp(2*x);
m:=sqrt(x+y-z);
n:=x*sqr(x)-abs(k-2);
writeln('y=',y:0:2,' z=',z:0:2,' k=',k:0:2);
writeln ('m=',m:0:2,' n=',n:0:2);
End.

```

Задания для самостоятельного решения:

Вариант №1

1. Даны целые числа a, b, c. Вычислить их полусумму и произведение.
2. Даны стороны прямоугольника. Найти периметр и площадь прямоугольника.
3. Вычислить значение функции $y = \sqrt{x^2 + 8}$.
4. Вычислить значение функции $y = e^{x^2 - 8} + 3$.

Вариант №2

1. Даны действительные числа x, y. Вычислить их полуразность и частное.
2. Дана сторона квадрата. Найти периметр, площадь и диагональ квадрата.
3. Вычислить значение функции $y = x^2 - 7x + 10$.
4. Вычислить значение функции $y = \cos^2(x + 3)$.

Вариант №3

1. Даны действительные числа c, d. Вычислить их полусумму и произведение.
2. Дан радиус окружности. Найти диаметр и длину окружности, площадь круга.
3. Вычислить значение функции $y = x^2 + \sqrt{x - 1}$.
4. Вычислить значение функции $y = |x + 3|$.

Вариант №4

1. Даны целые числа a и b. Найти полусумму их квадратов.
2. Даны стороны треугольника. Найти периметр и площадь треугольника.

3. Вычислить значение функции $y = \sqrt{x} + 3x$.

4. Вычислить значение функции $y = \ln(x^2 + 4)$.

Вариант №5

1. Ширина стороны прямоугольника B см, а длина в 2 раза больше. Найти площадь

2. Скорость первого автомобиля v_1 км/ч, второго — v_2 км/ч, расстояние между ними s км. Какое расстояние будет между ними через t ч, если автомобили движутся в разные стороны?

3. Вычислить значение функции $c = x - \ln x + \frac{y}{\cos x - \frac{x}{3}}$

4. Вычислить значение функции $c = \sin \sqrt{x+1} - \sin \sqrt{x-1}$
Ошибка! Закладка не определена.

Вариант № 6

1. Напишите программу, которая вводит радиус круга и которая вычисляет площадь этого круга и длину окружности.

2. Длина прямоугольника A см, а ширина меньше длины на 2 см. Найдите периметр прямоугольника и площадь квадрата, сторона которого равна длине прямоугольника

3. Вычислить значение функции $c = \frac{\cos x}{\pi - 2x} + 16 \cdot \cos(xy) - 2$

4. Вычислить значение функции $c = -\frac{y^2 + 12xy - 3x^2}{18y - 1}$

Вариант №7

1. Найти высоту и площадь равнобедренной трапеции, если известно одно основание, а другое в 3 раза больше заданного и угол при основании 45 градусов.

2. Середины сторон квадрата, заданного стороной, последовательно соединены. Найти периметр полученной фигуры.

3. Вычислить значение функции $c = \frac{1 + \sin \sqrt{x+1}}{\cos(12y - 4)}$

4. Вычислить значение функции $c = \frac{x^2 - 7x + 10}{x^2 - 8x + 12}$

Вариант № 8

1. Вводятся стороны прямоугольника. Найти периметр фигуры, образованной последовательным соединением середин сторон прямоугольника.

2. Известны диагонали ромба. Найти его площадь.

3. Вычислить значение функции $c = \frac{x + y}{x + 1} - \frac{xy - 12}{34 + x}$

4. Вычислить значение функции $c = \frac{\ln|\cos x|}{\ln(1 + x^2)}$

Вариант № 9

1. Город А находится в Х милях от Лондона. Напишите программу, которая вычислит расстояние между этими городами в километрах (5 миль равны 8 километрам).

2. Определить стоимость набора конфет, в который входят:

"Красная шапочка" - 200 г. "Алые паруса" - 150 г.

"Чародейка" - 100 г. если известна стоимость этих конфет за 1 кг.

3. Вычислить значение функции $c = \frac{\sin x + \cos y}{\cos x - \sin y} * \operatorname{arctg} xy$

4. Вычислить значение функции $c = 3 - 4x + (y - \sqrt{|x|})$

Вариант № 10

1. Длина стороны треугольника равна А, периметр равен Р, длины двух других сторон равны между собой. Найти эти длины.

2. Сколько времени в минутах затратит школьник на дорогу из школы, если известна длина расстояния S в километрах и средняя скорость движения школьника V км/час?

3. Вычислить значение функции $c = \frac{b + \sqrt{b^2 + 4ac}}{2a} - a^3c + b - 2$

4. Вычислить значение функции $c = \frac{d}{c} * \frac{b}{d}$

$$- \frac{ab-c}{cd}$$

Вариант № 11

1. Определить площадь треугольника по длине сторон

2. Длина прямоугольника A см, а ширина больше длины на 3 см. Найдите периметр прямоугольника и площадь квадрата, сторона которого равна высоте прямоугольника

3. Вычислить значение функции $c = \frac{3 + \sin\sqrt{x-1}}{\cos(12y+4)}$

4. Вычислить значение функции $c = \sqrt[3]{x} + (y - 4x)$

Вариант № 12

1. Найти высоту и площадь равнобедренной трапеции, если известно одно основание, а другое в 2 раза меньше заданного и угол при основании 30 градусов.

2. Скорость первого автомобиля v_1 км/ч, второго — v_2 км/ч, расстояние между ними s км. Какое расстояние будет между ними через t ч, если автомобили движутся в одном направлении?

3. Вычислить значение функции $c = \frac{2 + \cos\sqrt{x+2}}{\sin(6y+2)}$

4. Вычислить значение функции $c = \sqrt[4]{x+20} + (y+6)$

Вариант 13

1. Вводятся стороны прямоугольника. Найти периметр фигуры, образованной последовательным соединением середин сторон прямоугольника.

2. Дан радиус окружности. Найти диаметр и длину окружности, площадь круга.

3. Вычислить значение функции $c = \frac{b + \sqrt{b^2 + 4ac}}{2a} - a^3c + b - 2$

4. Вычислить значение функции $y = e^{x^2 - 8} + 3$.

Оформить отчет по практической работе

Требование к отчету

Отчет о проделанной работе должен содержать:

- название и цель работы;
- номер варианта для выполнения задания и условие своего варианта;
- блок-схемы решения задач;
- тексты программ;
- полученные при расчетах численные результаты;
- письменные ответы на контрольные вопросы

Контрольные вопросы

1. Как в программе на языке Pascal описываются переменные?
2. Какие бывают типы переменных?
3. Какой вид имеет оператор присваивания?
4. Каким символом отделяются друг от друга операторы в программе?
5. В каких случаях после оператора не ставится точка с запятой?

Практическая работа № 2, №3

Тема: Составление программ линейной структуры в Pascal

Цель работы: выработать практические навыки работы с системой ABCPascal, научиться создавать, вводить в компьютер, выполнять и исправлять простейшие программы на языке Pascal в режиме диалога, познакомиться с диагностическими сообщениями компилятора об ошибках при выполнении программ, реализующих линейные алгоритмы.

Ход работы:

Теоретическое обоснование:

Понятие алгоритма в программировании является фундаментальным. Для алгоритма важен не только набор определенных действий, но и то, как они организованы, т.е. в каком порядке они выполняются. Это одно из общих свойств алгоритма.

Алгоритмом обычно называют точно определенное правило действий, для которого задано указание, как и в какой последовательности исполнитель алгоритма должен применять это правило к исходным данным задачи, чтобы получить ее решение. Здесь имеется в виду самый общий смысл понятия «исполнитель» — им может быть человек или какое-либо устройство, например компьютер.

Алгоритм - это предназначенное для конкретного исполнителя точное описание последовательности действий, направленных на решение поставленной задачи.

Понятие алгоритма — одно из основных в математике. Нахождение алгоритмов для различных классов задач есть одна из целей математики. С практической точки зрения особую ценность представляют алгоритмы, приводящие к решению задачи наиболее коротким путем.

Слово «алгоритм» происходит от латинского Alhorithmi — латинская форма написания имени математика IX в. аль-Хорезми, сформулировавшего правила выполнения четырех арифметических действий над многозначными числами. Сегодня мы бы сказали, что он определил правила выполнения арифметических действий в столбик.

Свойства алгоритма

Детерминированность (определенность) — однозначность результата процесса исполнения алгоритма при заданных исходных данных, Один и тот же алгоритм не может получать два разных результата при одних и тех же данных. Так, если на вход алгоритма сложения поступают два числа, то, сколько бы раз мы ни предъявляли эти числа, ответ должен быть одним. Разный ответ получается не потому, что в разные моменты времени сложение выполняется по-разному, а потому, что ученик плохо усвоил правило сложения. Дискретность определяемого алгоритмом процесса — расчлененность его на отдельные элементарные акты, возможность выполнения которых человеком или машиной не вызывает сомнения. Одно дело — сказать: «Найди наиболее общий делитель чисел 1 345 672 и 164 930», и уж совсем иное дело — подробно показать, как это делается.

Массовость, или повторяемость - означает, что исходные данные для алгоритма можно выбирать из некоторого множества данных (потенциально бесконечного), т. е. алгоритм должен обеспечивать решение любой задачи из класса однотипных задач. Так, если вы описываете алгоритм решения некоторого класса алгебраических уравнений, то должны описать, как найти корни для любого уравнения из этого класса. В противном случае это не алгоритм, а лишь ваша догадка, что вы знаете, как решать такие уравнения.

Понятность алгоритма - состоит в том, что он должен быть описан в виде последовательности команд, каждая из которых принадлежит системе команд, понятной для исполнителя этого алгоритма. Если исполнителем алгоритма является человек, то он не должен наделять команды, участвующие в записи алгоритма, никакими собственными интерпретациями, ему следует выполнять в точности те действия, которые там указаны. Если исполнителем алгоритма является компьютер, то понятность в данном случае состоит в том, что программа, предлагаемая компьютеру для вычисления и записанная на каком-либо языке программирования, должна транслироваться на язык машинных команд соответствующим транслятором. Если такой транслятор не входит в математическое обеспечение вашего компьютера, то алгоритм, сколь бы хорош он ни был, так и останется простым текстом и не заставит машину совершать никаких действий.

Конечность (результативность) алгоритма - означает возможность получения результата после выполнения конечного количества операций. Так как компьютер не может задавать наводящие вопросы, уточняя ту или иную команду алгоритма, то последний должен иметь форму, которая не имеет неоднозначности. Для этого алгоритм представляется в виде программы, представленной на том или ином формальном (или еще говорят — алгоритмическом) языке. Формальный характер языка позволяет совершенно однозначно трактовать каждую команду этой программы, по-другому говорят, что алгоритмический язык обладает точной семантикой, в отличие от естественного языка, который обладает неоднозначностью, т. е. некоторые его предложения могут пониматься по-разному.

В настоящий момент в мире имеется огромное количество алгоритмических языков. Наиболее распространенными сейчас являются системы программирования, основанные на языках Бейсик, Паскаль и Си. Это языки очень популярны и обладают неплохими системами отладки и редактирования, позволяющими создавать эффективные программы и быстро их отлаживать.

Способы представления алгоритмов:

- естественный язык для словесно-пошагового способа записи алгоритма;
- язык блок-схем как графический способ записи алгоритма;
- языки программирования.

Рассмотрим все три способа записи алгоритма на примере решения конкретной задачи.

Задача. Найти площадь прямоугольника по двум его сторонам.

Решение: обозначим стороны прямоугольника переменными a и b , а его площадь переменной S . Формула площади прямоугольника $S=a*b$.

Теперь составим алгоритм решения данной задачи для компьютера и запишем его тремя способами.

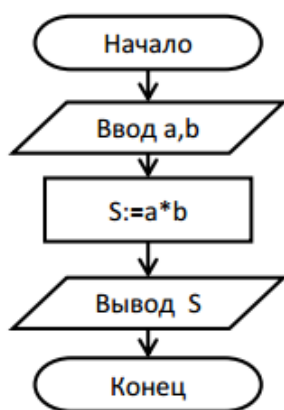
Из выше изложенного ясно, что для того чтобы вычислить площадь прямоугольника, компьютеру необходимо «знать» чему равны переменные a и b . Только тогда он сможет вычислить площадь по формуле. Чтобы алгоритм удовлетворял свойству массовости, необходимо, чтобы переменные a и b вводились с клавиатуры пользователем во время работы алгоритма. После того как компьютер

вычислит площадь по формуле, он должен вывести значение переменной S на экран, чтобы пользователь увидел результат работы алгоритма.

Словесная форма записи алгоритма:

- 1) Ввести в компьютер значение стороны a;
- 2) Ввести значение стороны b прямоугольника;
- 3) Рассчитать $S=a*b$;
- 4) Вывести на экран компьютера значение площади прямоугольника S.

Решение задачи в виде блок-схемы:



Запись алгоритма на языке Паскаль:

```
Program Task1;
Var a,b,s : integer;
Begin
    WriteLn('Введите стороны a и b:');
    ReadLn(a,b);
    S:=a*b;
    WriteLn('S=' , S , 'кв.см. ');
End.
```

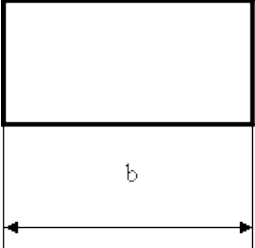
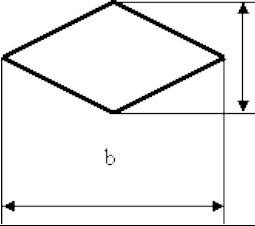
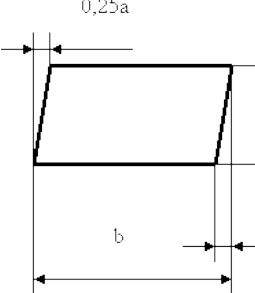
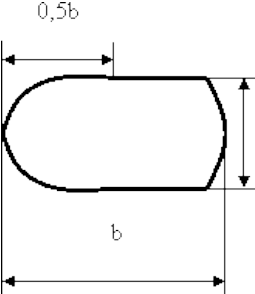
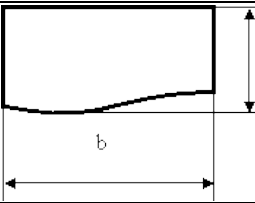
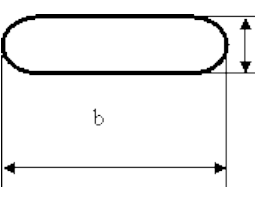

Блок-схема – это графическое представление алгоритма при помощи стандартных обозначений. Блок-схемы составляются в соответствии с ГОСТами. ГОСТы алгоритмов: ГОСТ 19.002-80, ГОСТ 19.003-80. На схемах алгоритмов выполняемые действия изображаются в виде отдельных блоков, которые соединяются между собой линиями связи в порядке выполнения действий. На линиях связи могут ставиться стрелки, причем, если направление связи слева направо или сверху вниз, то стрелки не ставятся. Существует популярный ГОСТ, который описывает требования и правила выполнения блок-схем: ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

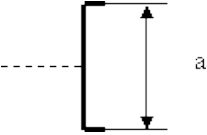
Правила оформления блок-схем алгоритмов

В таблице 1.2 показана форма и приведено содержание наиболее часто используемых блоков. Практически все блоки, приведенные в таблице, строятся на основе «базового» прямоугольника размерами «a x b». Первый приведенный в таблице блок под названием «процесс» по существу является «базовым» прямоугольником. Размер a = 10, 15, 20 ... мм, т.е. кратен пяти. Размер b = 1,5a,

допускается $b = 2a$. Размер «а» выбирается в зависимости от масштаба блок – схемы и должен быть одинаковым для всех ее блоков.

Таблица 1.2

Форма и содержание блоков блок – схем алгоритмов		
Наименование	Обозначение	Функции
1	2	3
Процесс		Выполнение операций присваивания, например, $A = 0$, сложения с присваиванием, например, $C = A+B$, вычитания, умножения и т.д.
Решение		Выбор направления выполнения алгоритма (программы) в зависимости от некоторых переменных условий
Ввод - вывод		Ввод – вывод информации без указания (независимо) от типа устройства ввода или вывода
Дисплей		Ввод информации с дисплея (с клавиатуры), вывод информации на дисплей
Документ		Вывод информации на бумагу (на принтер)
Пуск - останов		Начало – конец алгоритма (программы)
Соединитель		Переход на блок номер 5 (номер блока приведен для примера)

Комментарий		-
-------------	--	---

Каждая блок – схема начинается блоком «Пуск», внутри которого пишется слово «Начало», и заканчивается блоком «Останов», внутри которого пишется слово «Конец». Блоки соединяются линиями. Если линия, соединяющая блоки «приходит» в блок по направлению «сверху - вниз» или «слева – направо», то стрелка на ее конце не ставится (подразумевается). Если же по направлению «снизу – вверх» или «справа – налево», то стрелка на ее конце ставится обязательно.

Применение блока «Соединитель» позволяет значительно упростить блок – схему. Он применяется в том случае, когда линию, соединяющую блоку нужно вести на значительное расстояние и часто с пересечением других соединительных линий. Вместо этого достаточно после блока, из которого должна выходить соединительная линия, изобразить блок «Соединитель» и внутри его написать номер блока, в который эта линия должна прийти.

Блок «Комментарий» применяется в том случае, когда внутри какого – либо блока не удастся разместить (написать) всю необходимую информацию. В этом случае к линии, соединяющей блоки, перед блоком, для которого необходимо написать дополнительную информацию, посредством пунктирной линии присоединяется (справа или слева) блок «Комментарий». Информацию можно размещать по высоте - в пределах высоты данного блока, по ширине – до края страницы.

Пример. Определить расстояние на плоскости между двумя точками с заданными координатами $M_1(x_1, y_1)$ и $M_2(x_2, y_2)$.



Решение задачи. Расстояние на плоскости между двумя точками $M_1(x_1, y_1)$ и $M_2(x_2, y_2)$ высчитывается по формуле:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Составим блок-схему алгоритма, а затем уточним содержимое блоков "Вычисление расстояния" и "Вывод расстояния" (см. рис.1):

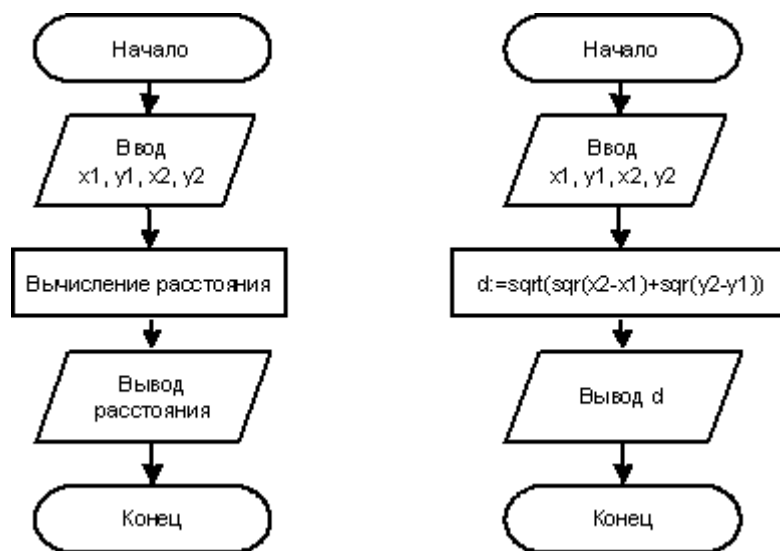


Рис. 1. Блок-схема алгоритма

Переводим блок-схему на язык Паскаль, доработав программу, чтобы улучшить ее интерфейс:

```

program example1;
var x1, x2, y1, y2: Integer;
d:Real;
begin
  
```

Writeln('Эта программа вычисляет расстояние между двумя точками на плоскости');

Writeln('Введите координаты двух точек:');

Write('x1= '); Readln(x1);

Write('y1= '); Readln(y1);

Write('x2= '); Readln(x2);

Write('y2 ='); Readln(y2);

d:=sqrt(sqr(x2-x1)+sqr(y2-y1));

Writeln('d= ',d:6:2);

Writeln('нажмите Enter для завершения работы программы');

Readln;

end.

Задачи для самостоятельного решения:

Вариант № 1

1. Вычислите высоты треугольника по трем заданным сторонам a, b, c
 2. Вычислить значение функции трех переменных при заданных значениях параметров: $x=2y+3 \operatorname{sh} t$ - при $y=2; t=5 / (1+y^2); z=4$
 3. Обменять значения двух переменных, используя третью (буферную) переменную
 4. Найти сумму и произведение 3 введенных с клавиатуры целых чисел
 5. Даны x, y, z . Вычислить a, b , если: $a = \frac{\sqrt{|x-1|-\sqrt{y}}}{1+\frac{x^2}{2}+\frac{y^2}{4}}, b = x(\operatorname{arctg}(z) + e)$
-

Вариант № 2

1. По координатам двух точек, которые вводит пользователь, определить уравнение прямой, проходящей через эти точки
 2. Вычислить значение функции трех переменных при заданных значениях параметров: $x=3 y^2 / (4 \operatorname{tg} z - 2t^2)$ при $t=0.5; z=6; y=t+2 \operatorname{ctg} z$.
 3. По координатам двух точек, которые вводит пользователь, определить уравнение прямой, проходящей через эти точки
 4. Найти стоимость заданного количества ткани. Цена и количество вводятся с клавиатуры
 5. Даны x, y, z . Вычислить a, b , если: $a = \frac{3+e^2}{1+x^2|y-\operatorname{tg}(z)|}, b = 1 + |y - z| + \frac{(y-x)^2}{2} + \frac{(x-y)^2}{3}$
-

Вариант № 3

1. Обменять значения двух переменных, используя третью (буферную) переменную.
 2. Вычислить значение функции трех переменных при заданных значениях параметров: $x=4y^2 / (4y e^z - 2t^3)$ при $t=1; z=3; y=\operatorname{sin} t$.
 3. Ввести с клавиатуры два целых числа, найти результат деления первого числа на второе и вывести числа и полученный результат на экран
 4. Вычислить длину окружности радиуса x .
-

5. Даны x, y, z . Вычислить a, b , если: $a = (1 + y) \frac{x + \frac{y}{(x^2+4)}}{e^2 + \frac{1}{(x^2+4)}}$, $b = \frac{1 + \cos^3(y-x)}{\frac{x^2}{2} + \sin^2(z)}$

Вариант № 4

1. Вычислите высоты треугольника по трем заданным сторонам a, b, c

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x = 4 \ln y^3 - z / \operatorname{tg} \operatorname{pr} = 2$; $z = 3$; $y = \cos(t+z)$.

3. Вывести на экран действительное число в различных форматах представления.

4. Обменять значения двух переменных, используя третью (буферную) переменную

5. Даны x, y, z . Вычислить a, b , если: $a = y + \frac{x}{y^2 + \left| \frac{x^2}{y+x^2} \right|}$, $b = (1 + \operatorname{tg}^2 \frac{z}{2})^2$

Вариант № 5

1. По координатам двух точек, которые вводит пользователь, определить уравнение прямой, проходящей через эти точки

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x = 6 t^2 - (\operatorname{ctg} z + 1) / y^2$ при $u = 2$; $z = 4$; $t = \sin(2+z)$.

3. Вычислите площадь поверхности цилиндра если известен диаметр и высота

4. Найти сумму и произведение 3 введенных с клавиатуры целых чисел

5. Даны x, y, z . Вычислить a, b , если: $a = \frac{2 \cos^4(x - \frac{\pi}{6})}{\frac{1}{2} + \sin^2 y}$, $b = 1 + \frac{z^2}{3 + \frac{z^2}{5}}$

Вариант № 6

1. Вычислите высоты треугольника по трем заданным сторонам a, b, c

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x = (8z^2 + 1) / (y e^t + t^2)$ при $z = 1$; $t = 2$; $y = \operatorname{tg} t + z$.

3. Вычислить длину окружности радиуса x

4. Найти стоимость заданного количества ткани. Цена и количество вводятся с клавиатуры

5. Даны x, y, z . Вычислить a, b , если: $a = \frac{\sqrt{|x-1| - \sqrt{y}}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$, $b = x^3 (\operatorname{arctg}^3 z + e)$

Вариант № 7

1. По координатам двух точек, которые вводит пользователь, определить уравнение прямой, проходящей через эти точки

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x=6 \operatorname{ch} t-3 z^2/(y+1)$ при $t=2$; $z=t+1$; $y=3$.

3. Обменять значения двух переменных, используя третью (буферную) переменную

4. Ввести с клавиатуры два целых числа, найти результат деления первого числа на второе и вывести числа и полученный результат на экран

5. Даны x, y, z . Вычислить a, b , если: $a = \frac{1+\sin^2(x+y)}{2+\left|x-\frac{2x}{(1+x^2y^2)}\right|} + x$, $b = \cos^2 \operatorname{arctg} \frac{1}{2}$

Вариант № 8

1. Вычислите высоты треугольника по трем заданным сторонам a, b, c

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x=8z / (e^t+2)-y^2$ при $t=1$; $z=\operatorname{ctg} t + 2$; $y=4$.

3. Ввести с клавиатуры три целых числа, найти результат деления суммы первого и второго числа на третье и вывести числа и полученный результат на экран

4. Вывести на экран действительное число в различных форматах представления

5. Даны x, y, z . Вычислить a, b , если: $a = \ln \left| \left(y - \sqrt{|x|} \right) \left(x - \frac{y}{z + \frac{x^2}{4}} \right) \right|$, $b = x -$

$$\frac{x^2}{3} + \frac{x+y}{x}$$

Вариант № 9

1. Вывести на экран действительное число в различных форматах представления.

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x=\cos(2a^3 + t^5)/z$ при $a=1$; $t=2$; $z=3t^2$

3. Вычислите площадь поверхности цилиндра радиуса r и высоты h

4. Найти сумму и произведение 3 введенных с клавиатуры целых чисел

5. Даны x, y, z . Вычислить a, b , если: $a = \frac{x^2}{8+\frac{x^2}{3}+\frac{y^2}{6}}$, $b = x(\cos^3(x+z) + 1)$

Вариант № 10

1. По координатам двух точек, которые вводит пользователь, определить уравнение прямой, проходящей через эти точки

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x = \cos(a^2) + t - a/\sin(z)$ при $z = \pi/2$; $a = 2$; $t = \exp(z)$

3. Вывести на экран действительное число в различных форматах представления

4. Найти стоимость заданного количества ткани. Цена и количество вводятся с клавиатуры

5. Даны x, y, z . Вычислить a, b , если: $a = \frac{|5-2e|}{1+x^2(y-tg(z))}$, $b = |y - 4| + \frac{(y-x)^2}{6} + \frac{(x-y)^2}{7}$

Вариант № 11

1. Вычислите высоты треугольника по трем заданным сторонам a, b, c

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x = 4 \ln y^3 - z / t$ при $t = 2$; $z = 3$; $y = \cos(t+z)$.

3. Вычислите площадь поверхности цилиндра диаметра d и высоты h

4. Вычислить длину окружности радиуса x

5. Даны x, y, z . Вычислить a, b , если: $a = (2 + x) \frac{1 + \frac{y}{(x^2+3)}}{y^2 + \frac{1}{(z+4)}}$,

$$b = (1 + tg^2 \frac{x}{2})^2 + |y - 4| + \frac{(y - x)^2}{6} + \frac{(x - y)^2}{7}$$

Вариант №12

1. Обменять значения двух переменных, используя третью (буферную) переменную.

2. Вычислить значение функции трех переменных при заданных значениях параметров: $x = (8z^2 + 1)/(y e^t + t^2)$ при $z = 1$; $t = 2$; $y = t g t + z$.

3. Вывести на экран действительное число в различных форматах представления

4. Ввести с клавиатуры два целых числа, найти результат деления первого числа на второе и вывести числа и полученный результат на экран

5. Даны x, y, z . Вычислить a, b , если: $a = \frac{z+10}{x^2 + \left| \frac{y^2}{y+x^2} \right|} + y, b = (1 + \cos^2 \frac{z}{2})^2$

Контрольные вопросы:

1. Какая процедура служит для вывода информации на печать?
2. Какая процедура служит для ввода значений с клавиатуры?
3. Какие функции служат для вычисления квадрата, квадратного корня, модуля, экспоненты числа или числового выражения?
4. Какие стандартные тригонометрические функции существуют в языке Pascal?
5. Как в среде Pascal запустить программу на выполнение?

Практическая работа № 4

Тема: Составление программ с использованием операций `div` и `mod`.

Цель работы: ознакомиться с операциями `div` и `mod` Научиться составлять программы с использованием данных операций

Ход работы:

Операции `div` и `mod`.

Целый тип данных

Переменные целого типа описываются посредством идентификатора `Integer`. Они могут принимать значения в диапазоне от -32768 до 32767. К данным целого типа можно применить операции "+"-сложение, "-"-вычитание, "*" -умножение и некоторые другие.

Так как в результате деления одного целого числа на другое не всегда получается целое число, то имеются операции:

div - целая часть от деления;

mod - остаток от деления.

Иногда нам требуется найти частное либо же остаток от деления. В такие моменты на помощь нам приходят такие операции, как `div` и `mod`. Заметим, что эти операции выполняются только над целыми числами.

Div Для того, чтобы найти частное от деления, мы используем операцию `div`.

Примеры:

- $25 \text{ div } 20 = 1;$
- $20 \text{ div } 25 = 0;$
- $39 \text{ div } 5 = 7;$
- $158 \text{ div } 3 = 52.$

Mod Для того, чтобы найти остаток от деления, мы используем операцию `mod`.

Примеры:

- $25 \text{ mod } 20 = 5;$
- $20 \text{ mod } 25 = 0;$
- $39 \text{ mod } 5 = 4;$
- $158 \text{ mod } 3 = 2.$

Round(12.75) = 13 Округление чисел с плавающей запятой до целого числа

Trunc(12.75) = 12 Возвращает целочисленную часть числа с плавающей запятой

Int(12.75) = 12 то же самое что и Trunc, но возвращает целое число в вещественном значении

Frac(12.75) = 0.75 Возвращает дробную часть числа с плавающей запятой

Примечание. Переменной целого типа присваивать значение, получаемое в результате выполнения обычной операции деления "/", **нельзя**, так как при делении одного числа на другое целое число результат не всегда является целым числом. При использовании операторов **div** и **mod** переменные описываются как переменные целого типа, т.е. ,например, integer или longint.

Чтобы окончательно понять, с чем мы имеем дело, решим следующую задачу:



Пример 1. Заданы два целых числа k и d . Используя только арифметические операции, найдите целую и дробную части от деления k на d

Этапы выполнения задания.

I. Определение исходных данных: переменные k, d .

II. Определение результатов: переменные $cel, drobn$.

III. Алгоритм решения задачи.

1. Ввод исходных данных

2. Вычисление значений $cel=k \text{ div } d$, $a=k/d$, $drobn=a-cel$.

3. Вывод результата.

IV. Описание переменных:

Переменные $a, drobn$ имеют тип *real*, а переменные k, d, cel имеют тип *integer*..

V. Программа:

```
var a,drobn:real;
```

```
k,d,cel:integer;
```

```
Begin
```

```
writeln('введите два числа'); readln(k,d);
```

```
cel:=k div d;           {целочисленное деление}
```

```
a:=k/d;                {обычное деление}
```

```
drobn:=a-cel;
```

writeln('целая часть - ',cel);

writeln('дробная часть - ',drobn);

End.

VI. Тестирование

1. Запустите программу и введите значения $k=2$, $d=3$

Проверьте, результат должен быть следующим:

cel= 1, drobn= 0.666666666667

Проверить правильность вычислений можно на калькуляторе.



Пример 2. Дано трехзначное число. Чему равны его цифры?

Этапы выполнения задания.

I. Определение исходных данных: переменная a (трехзначное число).

II. Определение результатов: переменные c1,c2,c3 (цифры числа).

III. Алгоритм решения задачи.

1. Ввод исходных данных

2. Выделение цифр числа:

$c1:=a \text{ div } 100;$

$c2:=(a \text{ mod } 100) \text{ div } 10;$ {или $c2:=(a \text{ div } 10) \text{ mod } 10;$ }

$c3:=a \text{ mod } 10;$

3. Вывод результата.

IV. Описание переменных:

Все переменные, определенные для решения задачи, имеют тип *integer*.

V. Программа:

var a,c1,c2,c3:**integer**;

Begin

writeln('введите трёхзначное число '); **readln**(a);

$c1:=a \text{ div } 100;$

$c2:=(a \text{ mod } 100) \text{ div } 10;$ {или $c2:=(a \text{ div } 10) \text{ mod } 10;$ }

$c3:=a \text{ mod } 10;$

writeln('первая цифра -',c1);

writeln('вторая цифра -',c2);

writeln('третья цифра -',c3);

End.

VI. Тестирование

1. Запустите программу и введите значения

a=234

Проверьте, результат должен быть следующим:

первая цифра - 2

вторая цифра - 3

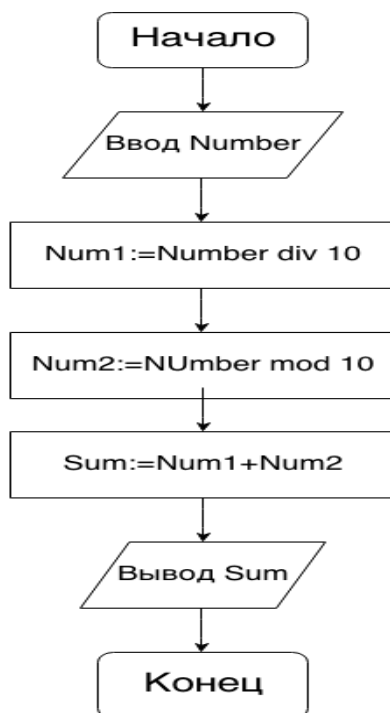
третья цифра - 4

2. Введите другие числа. Убедитесь в правильности работы программы.



Пример 3. Найти сумму цифр двухзначного числа.

Так как эта задача очень простая, составим блок-схему и программу.



Блок-схема

```
programSumoftwo;  
var Number,Num1,Num2,Sum: integer;  
begin  
write('Введите двухзначное число: ');  
read(Number); { Возьмемчисло 25 }  
Num1 := Number div 10; { 25 div 10 = 2 }  
Num2 := Number mod 10; { 25 mod 10 = 5 }  
Sum := Num1 + Num2; { 2 + 5 = 7 }
```

```
write('Сумма двух чисел -- ',Sum);  
end.
```

Задачи для самостоятельного решения

Вариант № 1

1. Найти разность между трехзначным числом и числом, составленным из тех же цифр, но взятых в обратном порядке
 2. Дано четырехзначное число. Определить: является ли сумма его цифр двузначным числом
 3. Найти остаток от деления целой части значения функции $y = \ln(x^2+ab)$ на 7 и, в зависимости от его величины, напечатать сообщение об одном из дней недели, пронумеровав их от 0 до 6. Значения переменных, a , b , x получить случайным образом на произвольном интервале (границы интервала ввести с клавиатуры).
 4. Найти сумму цифр трехзначного числа.
-

Вариант № 2

1. Составить программу перестановки первой и последней цифр введенного натурального числа
 2. Дано четырехзначное число. Определить: является ли произведение его цифр трехзначным числом
 3. Заданы два целых числа k и d . Используя только арифметические операции, найдите целую и дробную части от деления k на d
 4. Найти сумму цифр четырехзначного числа.
-

Вариант № 3

1. Составьте программу, которая подсчитывает сумму и произведение цифр введенного четырехзначного числа

2. Дано четырехзначное число. Определить: больше ли числа a произведение его цифр

3. Дано шестизначное число a . Найти сумму цифр числа

4. Найти сумму цифр пятизначного числа

Вариант № 4

1. Написать программу переработки данного целого числа $n > 9$ в целое число, записанное теми же цифрами, но в обратном порядке

2. Дано четырехзначное число. Определить: кратна ли пяти сумма его цифр

3. Дан номер дня в году. Сколько полных недель прошло с начала года?

4. Найти сумму цифр шестизначного числа

Вариант № 5

1. Дано пятизначное число. Нужно вычислить квадрат суммы всех цифр

2. Дано четырехзначное число. Определить: кратна ли сумма его цифр числу a

3. Найти остаток от деления целой части значения функции $y = \ln(x^2 + ab)$ на 7 и, в зависимости от его величины, напечатать сообщение об одном из дней недели, пронумеровав их от 0 до 6. Значения переменных, a , b , x получить случайным образом на произвольном интервале (границы интервала ввести с клавиатуры).

4. Дано натуральное четырехзначное число N . Найти разницу между суммой цифр и произведением цифр данного числа

Вариант № 6

1. Дано трехзначное число. Нужно вычислить квадрат суммы всех цифр

2. Дано четырехзначное число. Определить: равна ли сумма двух первых его цифр сумме двух его последних цифр

3. Заданы два целых числа k и d . Используя только арифметические операции, найдите целую и дробную части от деления k на d

4. Дано расстояние L в метрах (Целое число). Написать программу, которая переводит в км. и м. Для $L=7255$ результат вывести в виде: $7255 \text{ м} = 7 \text{ км. } 255 \text{ м}$

Вариант № 7

1. Дано шестизначное число a . Найти сумму цифр числа
 2. Дано четырехзначное число. Определить: кратна ли трем сумма его цифр
 3. Дано количество секунд, прошедших с начала суток. Сколько по отдельности часов, минут и секунд прошло с полуночи?
 4. Дано натуральное двухзначное число N . Написать программу, которая находит сумму цифр данного числа и частное от деления $T.e.$ Для $N=23$ результат вывести в виде: $N=23$ Сумма=5 Деление= 0.66
-

Вариант № 8

1. Дано трехзначное число. Определить: кратно ли произведение его цифр числу a . Число, a вводится с клавиатуры
 2. Дано четырехзначное число. Определить: кратно ли четырем произведение его цифр
 3. Сколько квадратов со стороной, a поместится в прямоугольнике со сторонами x и y ? Вычислите незаполненную площадь.
 4. Дано целое число N . Найти сумму, произведение цифр данного числа.
-

Вариант № 9

1. Найти остаток от деления целой части значения функции $y = \ln(x^2+ab)$ на 7 и, в зависимости от его величины, напечатать сообщение об одном из дней недели, пронумеровав их от 0 до 6. Значения переменных a , b , x получить случайным образом на произвольном интервале (границы интервала ввести с клавиатуры).
 2. Дано четырехзначное число. Определить: кратно ли произведение его цифр числу a
 3. Дано количество секунд, прошедших с начала суток. Сколько по отдельности часов, минут и секунд прошло с полуночи?
 4. Дано натуральное двухзначное число N . Найти разницу между первой и второй цифрой.
-

Вариант № 10

1. Заданы два целых числа k и d . Используя только арифметические операции, найдите целую и дробную части от деления k на d
 2. Дано четырехзначное число. Определить: является ли произведение его цифр двухзначным числом
 3. Сколько квадратов со стороной a поместится в прямоугольнике со сторонами x и y ? Вычислите незаполненную площадь.
 4. Дано натуральное пятизначное число N . Найти разницу между сумой цифр и произведением цифр данного числа
-

Вариант № 11

1. Дан номер дня в году. Сколько полных недель прошло с начала года?
 2. Дано четырехзначное число. Определить: равна ли сумма двух центральных его цифр сумме двух его крайних цифр
 3. Дано пятизначное число. Нужно вычислить квадрат суммы всех цифр
 4. Дано натуральное трёхзначное число N . Найти частное от деления произведения цифр на сумму цифр данного числа
-

Вариант № 12

1. Дано количество секунд, прошедших с начала суток. Сколько по отдельности часов, минут и секунд прошло с полуночи?
 2. Дано четырехзначное число. Определить: кратна ли пяти произведение его цифр
 3. Дано шестизначное число a . Найти сумму цифр числа
 4. Дано натуральное четырёхзначное число N . Найти частное от деления произведения цифр на сумму цифр данного числа
-

Контрольные вопросы

1. Что такое «тип данных»? Для чего нужен тип данных? Какие типы данных есть в Паскале?
2. Как записываются арифметические операции в языке Паскаль?
3. Для чего нужен формат вывода данных? Как он задается?
4. Для чего предназначены операторы `div` и `mod`?
5. Запишите встроенные математические функции языка Паскаль.

Практическая работа № 5, №6

Тема: Составление программ ветвящейся структуры в Pascal.

Цель работы: научиться составлять программы для решения задач содержащих условия, продолжить освоение работы в системе программирования ABCPascal.

Ход работы:

Теоретическое обоснование

Конструкция ветвления- это часть алгоритма, в которой в зависимости от выполнения или невыполнения некоторого условия выполняется либо одна, либо другая последовательность действий.

Алгоритм, в котором используется конструкция ветвления, называется алгоритмом с ветвлением.

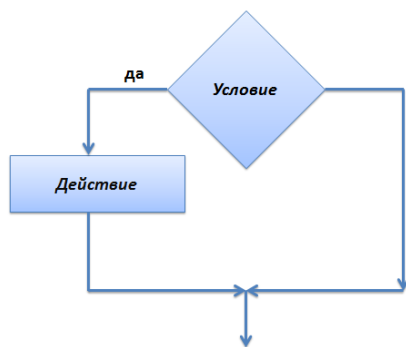
Условие бывает:

Неполное

Полное

Неполное условие

Блок-схема неполного условия выглядит следующим образом:



Неполное условие

**ЕСЛИ УСЛОВИЕ ИСТИННО, ТО ВЫПОЛНЯЕТСЯ
ДЕЙСТВИЕ, ИНАЧЕ НИЧЕГО НЕ ПРОИСХОДИТ**

На языке Паскаль данная алгоритмическая конструкция выглядит следующим образом:

IF условие THEN действие;

Если условие, стоящее после IF истинно, то выполняется действие, стоящее после слова THEN.

Что такое условие?

Условие — это выражение, которое может быть либо истинным, либо ложным. Условие обязательно содержит логические операторы $<$, $>$, $=$, $<=$, $>=$, $<>$ (не равно).



Например:

```
var a:integer;
begin
  readln(a);
  if a=7 then writeln('Привет!');
end.
```

Если введенное значение переменной a равно 7, то на экране мы увидим слово Привет! Если не равно — то на экран ничего выводится не будет.



Рассмотрим другой пример:

```
var a:integer;
begin
  readln(a);
  if a=7 then writeln('Привет!'); writeln('Довстречи');
end.
```

Что мы увидим на экране, введя число 7? Увидим:

Привет!

До встречи

Что мы увидим на экране, введя число 10?

До встречи

Почему так? Почему До встречи выводится на экран в любом случае?

Команда

```
writeln('До встречи');
не относится к конструкции If-Then
var a:integer;
begin
  readln(a);
```

```
if a=7 thenwriteln('Привет!'); writeln('До встречи'); // условие подсвечено голубым цветом
end.
```

Как сделать так, чтобы оба действия относились к конструкции If-Then?

Необходимо заключить эти действия в так называемые операторные скобки: begin... end;

Получим:

```
var a:integer;
begin
writeln(a);
if a=7 then begin writeln('Привет!');
writeln('До встречи');
end;
end.
```

Теперь, если мы введем число 10, то на экране ничего не увидим.

Составное (сложное) условие

Иногда приходится использовать сложное условие. Для его составления используются логические союзы: and или or.

Если мы используем and, то составное условие будет истинно, когда все простые условия истинны.

Если мы используем or, то составное условие будет истинно, когда хотя бы одно простое условие будет истинно.

Например:

$a > 7$ и $a < 15$ на языке паскаль будет записываться

$(a > 7) \text{ and } (a < 15)$

т.е. оба условия должны выполняться одновременно, чтобы составное условие было истинно

Или:

$(a = 7) \text{ or } (a > 15)$

Составное условие будет истинно, если: либо $a = 7$, либо $a > 15$.

Задача:

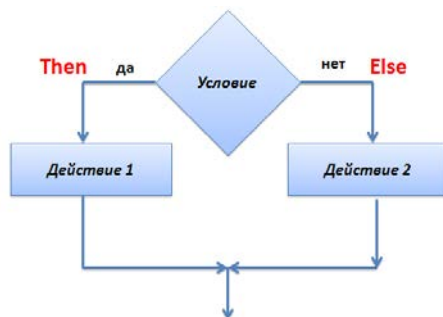
Используя конструкцию If-Then, найти максимальное среди трех введенных чисел.

Решение

```
var a, b, c:integer;  
begin  
  readln(a);  
  readln(b);  
  readln(c);  
  if (a>b) and (a>c) then writeln('число ', a, ' максимальное');  
  if (b>a) and (b>c) then writeln('число ', b, ' максимальное');  
  if (c>a) and (c>b) then writeln('число ', c, ' максимальное');  
end.
```

Полное условие

Блок-схема полного условия выглядит так:



Полное условие. Блок-схема

```
IF условие THEN действие_1 ELSE действие_2;
```

Если условие истинно, то выполняется действие, стоящее после слова Then.

Если условие ложно, то выполняется действие, стоящее после слова Else.

Если действий, которые выполняются, если условие истинно или ложно несколько — используются операторные скобки.

```
if условие then
```

```
begin
```

```
  действие;
```

```
  действие;
```

```
.....
```

```
end
```

```
else
begin
    действие;
    действие;
    .....
end;
(перед else точка с запятой не ставится)
```



Пример. Проверить является ли введенное число трехзначным, и вывести четные цифры числа.

Этапы выполнения задания.

I. Определение исходных данных: переменная a (трехзначное число).

II. Определение результатов: переменные a_1 , a_2 , a_3 , в том случае если они четные или сообщение, что четных цифр нет.

III. Алгоритм решения задачи.

1. Ввод исходных данных

2. Проверка является ли число трехзначным. Трехзначное число больше 99 и меньше 1000.

3 Если число трехзначное, то выделяем цифры числа и проверяем каждую из них на четность.

а) Для выделения первой цифры (переменная a_1) трехзначного числа необходимо найти целую часть от деления числа на 100.

б) Для выделения второй цифры (переменная a_2) трехзначного числа необходимо найти остаток от деления числа на 100 и от него найти целую часть при делении на 10.

в) Для определения последней цифры (переменная a_3) трехзначного числа необходимо найти остаток от деления числа на 10.

г) Для проверки цифры на четность нужно проверить, равен ли нулю остаток от деления цифры на 2.

4. Вывод результата.

IV. Описание переменных:

Все переменные, определенные для решения задачи, имеют тип integer.

V. Программа:

```
Var a,a1,a2,a3: integer;
```

```
Begin
```

```
  Write('введите значение a');
```

```
  Readln(a);
```

```
  If (a>99) and (a<1000) then
```

```
    Begin
```

```
      a1:=a div 100;
```

```
      a2:=a mod 100 div 10;
```

```
      a3:=a mod 10;
```

```
      if a1 mod 2=0 then Writeln('цифра ',a1,' четная');
```

```
      if a2 mod 2=0 then Writeln('цифра ',a2,' четная');
```

```
      if a3 mod 2=0 then Writeln('цифра ',a3,' четная');
```

```
      If (a1 mod 2=1) and (a2 mod 2=1) and (a3 mod 2=1)then
```

```
        Writeln('в числе нет четных цифр');
```

```
    End
```

```
    Else Writeln('число не является трехзначным');
```

```
End.
```



Пример. Написать программу для решения геометрической задачи.

Определить, является ли треугольник со сторонами a , b , c равнобедренным. Если "да", то вычислить его периметр.

Этапы выполнения задания.

I. Определение исходных данных: переменные a , b , c .

II. Определение результатов: переменная p , в том случае если треугольник равнобедренный или сообщение, что треугольник не равнобедренный.

III. Алгоритм решения задачи.

1. Ввод исходных данных

2. Проверка является ли треугольник равнобедренным.

3 Если треугольник равнобедренный, то вычислим его периметр $p:=a+b+c$, иначе выведем сообщение "треугольник не равнобедренный" .

4. Вывод результата.

IV. Описание переменных:

Все переменные, определенные для решения задачи, имеют тип integer.

V. Программа:

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

```
Begin
```

```
  Write('введите значения сторон треугольника a,b,c:');
```

```
  Readln(a,b,c);
```

```
  If (a=b) or (b=c) then
```

```
    Begin p:=a+b+c;  Writeln('p= ',p); End
```

```
  Else Writeln('треугольник не равнобедренный');
```

```
End.
```

Задачи для самостоятельного решения:

Вариант № 1

1. Дано целое число. Если оно является положительным то прибавить к нему 1, в противном случае вычесть из него два. Вывести полученное число.

2. Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, т. е. имеют ли они равные площади.

3. Ввести 2 числа. Если их произведение отрицательно, умножить его на -2 и вывести на экран, в противном случае увеличить его в $1,5$ раза и вывести на экран. (Написать программу, начертить блок-схему).

4. Для данного x вычислить значение функций: $f(x) =$

$$\begin{cases} x^2 - 3x + 9, & \text{если } x \leq 3; \\ \frac{1}{x^3+6}, & \text{если } x > 3. \end{cases}$$

Вариант № 2

1. Ввести число. Если оно четное, разделить его на 4, если нечетное - умножить на 5.

2. Ввести рост человека. Вывести на экран "ВЫСОКИЙ", если его рост превышает 180 см, и "НЕ ОЧЕНЬ ВЫСОКИЙ" в противном случае.

3. Составить программу, которая спрашивает возраст человека и, если ему 18 лет и больше, сообщает “Замечательно. Вы уже можете водить автомобиль”, а в противном случае – “К сожалению, водить автомобиль Вам рановато”.

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} -x^2 + 3x + 9, & \text{если } x \geq 3; \\ \frac{1}{x^3 - 6}, & \text{если } x < 3. \end{cases}$

Вариант № 3

1. Вводятся три разных числа. Найти, какое из них является средним (больше одного, но меньше другого).

2. Вовочка, любитель стрелять из рогатки, 7 раз попадал в милицию. Ввести с клавиатуры целое положительное число – № попадания. Определить результат: 4,6,7 – милиционеры вставляли новое стекло, 2,5 – новое стекло вставлял папа Вовочки, 1, 3 – стекло не разбилось.

3. Составить программу, которая запрашивает ввод трех значений температуры и проверяет, есть ли среди них температура таяния льда?

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} 9, & \text{если } x \leq -3; \\ \frac{1}{x^2 + 1}, & \text{если } x > -3. \end{cases}$

Вариант № 4

1. Вводятся три значения ускорения свободного падения. Программа должна проверить, есть ли среди них ускорение свободного падения.

2. Составить программу, которая запрашивает ввод формул трех кислот и проверяет, есть ли среди них формула серной кислоты?

3. Составить программу, которая запрашивает ввод температуры тела человека и определяет, здоров он или болен (здоров при $36 < t < 37$)?

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} 0, & \text{если } x \leq 1; \\ \frac{1}{x+6}, & \text{если } x > 1. \end{cases}$

Вариант № 5

1. Вводятся три разных числа. Найти, какое из них является средним (больше одного, но меньше другого).

2. Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, т. е. имеют ли они равные площади.

3. Составить программу, которая спрашивает возраст человека и, если ему 18 лет и больше, сообщает “Замечательно. Вы уже можете водить автомобиль”, а в противном случае – “К сожалению, водить автомобиль Вам рановато”.

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} -3x + 9, & \text{если } x \leq 7; \\ \frac{1}{x-7}, & \text{если } x > 7. \end{cases}$

Вариант № 6

1. Дано целое число. Если оно является положительным то прибавить к нему 8, в противном случае вычесть из него два. Вывести полученное число.

2. Ввести рост человека. Вывести на экран “ВЫСОКИЙ”, если его рост превышает 180 см, и “НЕ ОЧЕНЬ ВЫСОКИЙ” в противном случае.

3. Составить программу, которая запрашивает ввод трех значений температуры и проверяет, есть ли среди них температура таяния льда?

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} 3x - 9, & \text{если } x \leq 7; \\ \frac{1}{x^2-4}, & \text{если } x > 7. \end{cases}$

Вариант № 7

1. Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, т. е. имеют ли они равные площади.

2. Составить программу, которая спрашивает возраст человека и, если ему 18 лет и больше, сообщает “Замечательно. Вы уже можете водить автомобиль”, а в противном случае – “К сожалению, водить автомобиль Вам рановато”.

3. Вовочка, любитель стрелять из рогатки, 7 раз попадал в милицию. Ввести с клавиатуры целое положительное число – № попадания. Определить результат: 4,6,7

– милиционеры вставляли новое стекло, 2,5 – новое стекло вставлял папа Вовочки, 1, 3 – стекло не разбилось.

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} x^2, & \text{если } 0 \leq x \leq 3; \\ 4, & \text{если } x > 3 \text{ или } x < 0. \end{cases}$

Вариант № 8

1. Ввести 2 числа. Если их произведение отрицательно, умножить его на -2 и вывести на экран, в противном случае увеличить его в $1,5$ раза и вывести на экран. (Написать программу, начертить блок-схему).

2. Вводятся три значения ускорения свободного падения. Программа должна проверить, есть ли среди них ускорение свободного падения.

3. Составить программу, которая запрашивает ввод температуры тела человека и определяет, здоров он или болен (здоров при $36 < t < 37$)?

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} x^2 + 4x + 5, & \text{если } x \leq 2; \\ \frac{1}{x^2 + 4x} 5, & \text{если } x > 2. \end{cases}$

Вариант № 9

1. Ввести число. Если оно четное, разделить его на 4, если нечетное - умножить на 5.

2. Ввести рост человека. Вывести на экран “ВЫСОКИЙ”, если его рост превышает 180 см, и “НЕ ОЧЕНЬ ВЫСОКИЙ” в противном случае.

3. Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, т. е. имеют ли они равные площади.

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} x^2 - x, & \text{если } 0 \leq x \leq 1; \\ x^2 - \sin \pi x^2, & \text{если } x > 1 \text{ или } x < 0. \end{cases}$

Вариант № 10

1. Составить программу, которая спрашивает возраст человека и, если ему 18 лет и больше, сообщает “Замечательно. Вы уже можете водить автомобиль”, а в противном случае – “К сожалению, водить автомобиль Вам рановато”.

2. Дано целое число. Если оно является положительным то прибавить к нему 1, в противном случае вычесть из него три. Вывести полученное число.

3. Составить программу, которая запрашивает ввод трех значений температуры и проверяет, есть ли среди них температура таяния льда?

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} \sin x, & \text{если } x \leq 0; \\ \cos x, & \text{если } x > 0. \end{cases}$

Вариант № 11

1. Составить программу, которая запрашивает ввод температуры тела человека и определяет, здоров он или болен (здоров при $36 < t < 37$)?

2. Ввести 2 числа. Если их произведение отрицательно, умножить его на -2 и вывести на экран, в противном случае увеличить его в 1,5 раза и вывести на экран. (Написать программу, начертить блок-схему).

3. Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, т. е. имеют ли они равные площади.

4. Для данного x вычислить значение функций: $f(x) = \begin{cases} x^2 + 4x + 5, & \text{если } x \leq 5; \\ \frac{1}{x^2 + 4x} 5, & \text{если } x > 5. \end{cases}$

Вариант № 12

1. Ввести число. Если оно четное, разделить его на 4, если нечетное - умножить на 5.

2. Вводятся три значения ускорения свободного падения. Программа должна проверить, есть ли среди них ускорение свободного падения.

3. Составить программу вычисления значений функции для любого x по желанию пользователя.

4. Для данного x вычислить значение функций: $f(x) =$

$$\begin{cases} x^2, \text{ если } 0 \leq x \leq 3; \\ 4, \text{ если } x > 3 \text{ или } x < 0. \end{cases}$$

Контрольные вопросы

- 1 Что понимают под алгоритмом ветвления?
- 2 Привести примеры случаев ветвления.
- 3 Как обозначается ветвление в блок-схемах?
- 4 Какие операторы ветвления существуют в языке ABC Pascal?
- 5 Какой формат имеет оператор IF?
- 6 Какие различия между полной и сокращенной формой оператора IF?

Практическая работа № 7

Тема: Составление программ для решения задач, содержащих оператор выбора CASE

Цель: научиться составлять программы для решения задач, содержащих оператор выбора CASE, продолжить освоение работы в системе программирования ABC Pascal.

Ход работы:

Теоретическое обоснование

Оператор case

Оператор выбора позволяет программировать ветвления по многим направлениям. Этот оператор организует переход на одну из нескольких ветвей в зависимости от значения заданного выражения (селектора выбора).

Формат оператора выбора:

Case K of

A1: <оператор 1>;

A2: <оператор 2>;

...

AN: <оператор N>

Else <оператор N+1>

End;

Здесь K – выражение – селектор, которое может иметь только простой порядковый тип (целый, символьный, логический). A1,...,AN – константы того же типа, что и селектор, выполняющие роль меток ветвей. Исполнение оператора начинается с вычисления выражения K, полученное значение сравнивается с константами (метками) и выполняется соответствующий оператор. Если ни одна из меток не совпала со значением K, то выполняется оператор после слова else.

Возможно использование неполного оператора выбора без ветви else. Метки ветвей также могут быть заданы списком или интервалом.



Пример 1. Вводится число от 1 до 4, определяющее пору года. Дать название этой пору года (1 - зима, 2 - весна, 3 - лето, 4 - осень).

Этапы выполнения задания.

- I. Определение исходных данных: переменная n.
- II. Определение результатов: вывод названия поры года.
- III. Алгоритм решения задачи.
 1. Ввод исходных данных
 2. Проверяем чему равно n и соответственно выводим название поры года.
- IV. Описание переменных:

Переменная n типа byte.

V. Программа:

```
var
n:byte;
Begin
writeln('Введите номер поры года');
readln(n);
case n of
1: writeln('зима');
2: writeln('весна');
3: writeln('лето');
4: writeln('осень');
else
writeln('неправильно ввели номер поры года');
end;
End.
```



Пример 1

VI. Тестирование программы:

1. Проверьте работу программы для следующих значений

Ввод n	Вывод
5	неправильно ввели номер поры года
2	весна
1	зима

2. Поэкспериментируйте с программой вводя различные значения.



Пример 2. Составим программу "КАЛЬКУЛЯТОР", которая после ввода двух чисел и одного из знаков +, -, *, / произведёт вычисления, а результат выдаст на экран.

Этапы выполнения задания.

I. Определение исходных данных: переменные a,b,sim.

II. Определение результатов: переменная s.

III. Алгоритм решения задачи.

1. Ввод исходных данных

2. Проверяем чему равно sim и соответственно выполняем нужное действие.

IV. Описание переменных:

Переменные a,b,s типа real, sim типа char .

V. Программа:

```
var
a,b,s:real;
sim:char;
Begin
writeln('Введите два числа');
readln(a,b);
writeln('Введите знак операции');
readln(sim);
case sim of
'+': s:=a+b;
'-': s:=a-b;
'*': s:=a*b;
'/': s:=a/b;
end;
writeln ('результат ',a,sim,b,' = ',s);
```

End.



Пример 2

VI. Тестирование программы:

1. Проверьте работу программы для следующих значений

Ввод a,b,sim	Вывод
Введите два числа 3 4 Введите знак операции *	результат $3*4 = 12$
Введите два числа 3 4 Введите знак операции +	результат $3+4 = 7$
Введите два числа 12 6 Введите знак операции -	результат $12-6 = 6$

2. Поэкспериментируйте с программой вводя различные значения.

Замечание. В данной программе отсутствует часть `else` и поэтому, если ввести вместо рассматриваемых арифметических знаков, ввести любой символ, то программа будет работать, но будет работать неверно. Устраните этот недочет.

Задачи для самостоятельного решения

Вариант 1

1. Дано целое число в диапазоне 1–7. Вывести строку — название дня недели, соответствующее данному числу (1 — «понедельник», 2 — «вторник» и т. д.).
2. Дано целое число К. Вывести строку-описание оценки, соответствующей числу КК (1 — «плохо», 2 — «неудовлетворительно», 3 —

«удовлетворительно», 4 — «хорошо», 5 — «отлично»). Если КК не лежит в диапазоне 1–5, то вывести строку «ошибка».

3. Написать программу перевода числовой оценки в текстовую

Вариант 2

1. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 — февраль и т. д.). Вывести название соответствующего времени года («зима», «весна», «лето», «осень»).

2. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 — февраль и т. д.). Определить количество дней в этом месяце для невисокосного года.

3. Ввести число. Определить четное или нечетное это число

Вариант 3

1. Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия NN (целое число в диапазоне 1–4) и вещественные числа AA и BB (BB не равно 0). Выполнить над числами указанное действие и вывести результат.

2. Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины (целое число в диапазоне 1–5) и длина отрезка в этих единицах (вещественное число). Найти длину отрезка в метрах.

3. Создать простейший калькулятор

Вариант 4

1. Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы (целое число в диапазоне 1–5) и масса тела в этих единицах (вещественное число). Найти массу тела в килограммах.

2. Даны два целых числа: DD (день) и MM (месяц), определяющие правильную дату невисокосного года. Вывести значения DD и MM для даты, предшествующей указанной.

3. Написать программу перевода римских цифр в арабские

Вариант 5

1. Даны два целых числа: DD (день) и MM (месяц), определяющие правильную дату невисокосного года. Вывести значения DD и MM для даты, следующей за указанной.

2. Робот может перемещаться в четырех направлениях («С» — север, «З» — запад, «Ю» — юг, «В» — восток) и принимать три цифровые команды: 0 — продолжать движение, 1 — поворот налево, -1 — поворот направо. Дан символ СС — исходное направление робота и целое число NN - посланная ему команда. Вывести направление робота после выполнения полученной команды.

3. Написать программу «Гороскоп»

Вариант № 6

1. Вводится число экзаменов $N \leq 20$. Напечатать фразу "Мы успешно сдали N экзаменов", согласовав слово "экзамен" с числом N. Предусмотреть ситуацию ввода $N < 0$.

2. Вводится число - номер месяца. Вывести количество дней в месяце (год невисокосный). Предусмотреть неправильный ввод номера месяца.

3. Вводится число лет ($N \leq 25$). Напечатать фразу "Мне N лет", согласовав слово "лет" с числом N. Предусмотреть ситуацию ввода $N < 0$.

Вариант № 7

1. Вводится число книг $N \leq 10$. Вывести фразу "Я взял из библиотеки N книг", согласовав слово "книга" с числом N. Предусмотреть ситуацию ввода $N < 0$.

2. Вводится число карандашей $N \leq 10$. Вывести фразу "Я купил N карандашей", согласовав слово "карандаш" с числом N. Предусмотреть ситуацию ввода $N < 0$.

3. Вводится число версий $N \leq 10$. Вывести фразу "Следователь проверил N версий", согласовав слово "версия" с числом N. Предусмотреть ситуацию ввода $N < 0$.

Вариант № 8

1. Вводится число программ $N \leq 10$. Напечатать фразу "Я разработал N программ", согласовав слово "программа" с числом N. Предусмотреть ситуацию ввода $N < 0$.

2. Вводится целое число C . Если $-9 \leq C \leq 9$ вывести величину числа в словесной форме с учетом знака, в противном случае - предупреждающее сообщение и повторный ввод.

3. Вводится целое число K . вывести фразу «Я посадил K цветов», согласовав слово "цветы" с числом K . Предусмотреть ситуацию ввода $K < 0$.

Вариант № 9

1. Вводится число мешков $N \leq 10$. Вывести фразу "Я перенес N мешков", согласовав слово "мешок" с числом N . Предусмотреть ситуацию ввода $N < 0$.

2. Дано целое число в диапазоне 1–7. Вывести строку — название дня не дели, соответствующее данному числу (1 — «понедельник», 2 — «вторник» и т. д.).

3. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 февраль и т. д.). Вывести название соответствующего времени года («зима», «весна», «лето», «осень»).

Вариант № 10

1. Написать программу перевода числовой оценки в текстовую

2. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 февраль и т. д.). Определить количество дней в этом месяце для невисокосного года.

3. Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины (целое число в диапазоне 1–5) и длина отрезка в этих единицах (вещественное число). Найти длину отрезка в метрах.

Вариант № 11

1. Даны два целых числа: DD (день) и MM (месяц), определяющие правильную дату невисокосного года. Вывести значения DD и MM для даты, предшествующей указанной

2. Вводится число экзаменов $N \leq 20$. Напечатать фразу "Мы успешно сдали N экзаменов", согласовав слово "экзамен" с числом N . Предусмотреть ситуацию ввода $N < 0$

3. Вводится число книг $N \leq 10$. Вывести фразу "Я взял из библиотеки N книг", согласовав слово "книга" с числом N . Предусмотреть ситуацию ввода $N < 0$.

Вариант № 12

1. Написать программу перевода римских цифр в арабские
 2. Даны два целых числа: DD (день) и MM (месяц), определяющие правильную дату невисокосного года. Вывести значения DD и MM для даты, следующей за указанной
 3. Вводится целое число C. Если $-9 \leq C \leq 9$ вывести величину числа в словесной форме с учетом знака, в противном случае - предупреждающее сообщение и повторный ввод.
-

Контрольные вопросы

1. Что такое оператор case?
2. Что позволяет выполнять оператор выбора?
3. С какими типами данных может работать оператор выбора?
4. Сколько строк может быть записано в списке выбора?
5. Может ли в операторе выбора отсутствовать часть else?
6. формулируйте, что может являться ключом выбора?
7. Можно ли оператор выбора заменить условным оператором if ... then?

Практическая работа № 8, №9, № 10

Тема: Циклы в Turbo Pascal. Решение задач на циклы

Цель работы : Изучить структуру оператора цикла с параметром языка Pascal, научиться составлять циклические алгоритмы, создавать программы, используя полученные знания.

Теоретический материал :

В программах, связанных с обработкой данных или вычислениями, часто выполняются циклически повторяющиеся действия.

Цикл представляет собой последовательность операторов, которая выполняется неоднократно. В языке Pascal различают три вида операторов цикла: цикл с предусловием (while), цикл с постусловием (repeat) и цикл с параметром (for).

Следует знать:

- подавляющее большинство задач с циклами можно решить разными способами, используя при этом любой из трех операторов цикла;
- часто решения, использующие разные операторы цикла, оказываются равноценными;
- в некоторых случаях все же предпочтительнее использовать какой-то один из операторов;
- самым универсальным из всех операторов цикла считается while, поэтому в случае затруднений с выбором можно отдать предпочтение ему;
- цикл repeat имеет очень простой и понятный синтаксис, поэтому с него удобно начинать изучение циклов;
- цикл for обеспечивает удобную запись циклов с заранее известным числом повторений;
- при неумелом использовании циклов любого типа возможна ситуация, когда компьютер не сможет нормально закончить цикл. При работе в среде Turbo Pascal для выхода из подобной ситуации используется комбинация клавиш <Ctrl>+<Break>.
- если это не помогает, есть и крайнее средство – < Ctrl >+<Alt>+. Одновременное нажатие этих трех клавиш или кнопки Reset, расположенной на

системном блоке, позволяет перезагрузить компьютер, при этом данные, относящиеся к работающей программе, будут утеряны.

- процедура `continue` позволяет прервать выполнение тела любого цикла и передает управление на его заголовок, заставляя цикл немедленно перейти к следующему выполнению.

Циклы с параметром

Если число требуемых повторений заранее известно, то используется оператор, называемый оператором цикла с параметром или циклом со счетчиком.

*цикл по <счетчик> от <нач.знач.> до <конечн.знач.>

** действие

* конец цикла

Оператор цикла с параметром имеет два варианта записи:

1) *вариант с увеличением счетчика*

```
for <счетчик> := <начальное значение> to <конечное значение> do  
<тело цикла>
```

2) *вариант с уменьшением счетчика*

```
for <счетчик> := <нач. значение> downto <конечное значение> do  
<тело цикла>
```

Счетчик – параметр цикла, простая переменная целого типа; <тело цикла> - операторы или оператор. Цикл повторяется до тех пор пока значение параметра лежит в интервале между начальным и конечным значениями. В первом варианте при каждом повторении цикла значение параметра увеличивается на 1, во втором - уменьшается на 1.

При первом обращении к оператору `for` вначале определяются начальное и конечное значения, и присваивается параметру цикла начальное значение. После этого циклически повторяются следующие действия.

1. Проверяется условие: параметр цикла \leq конечному значению.
2. Если условие выполняется, то оператор продолжает работу (выполняется оператор в теле цикла), если условие не выполняется, то оператор завершает работу и управление в программе передается на оператор, следующий за циклом.

3. Значение параметра изменяется (увеличивается на 1 или уменьшается на 1).

Если в теле цикла располагается более одного оператора, то они заключаются в операторные скобки `begin ... end`;

Следует знать:

- оператор `For` используется для организации циклов с фиксированным, заранее известным числом повторений;
- количество повторений цикла определяется начальным и конечным значениями переменной-счетчика. Оператор `For` обеспечивает выполнение тела цикла до тех пор, пока не будут перебраны все значения параметра цикла: от начального до конечного;
- переменная счетчик должна быть порядкового типа: чаще `integer`, реже – `char`, `boolean`. Использование вещественного типа недопустимо;
- начальное и конечное значения параметра цикла могут быть константами, переменными, выражениями и должны принадлежать к одному и тому же типу данных. Начальное и конечное значение параметра цикла нельзя изменять во время выполнения цикла;
- параметр цикла `For` может изменяться (увеличиваться или уменьшаться) каждый раз при выполнении тела цикла только на единицу. Если нужен другой шаг, предпочтительнее использовать циклы с предусловием или с постусловием.



Задача 1. Вывести на экран натуральные числа от 1 до 9 в обратном порядке.

```
Program z1;  
Var i:integer;  
Begin  
For i:=9 downto 1 do  
    Writeln(i);  
End.
```



Задача 2.

Вычисление $p = n!$ (n факториал).

```
program z2;
var p, i, n: integer;
begin
write ('введите целое n = ');
readln (n);
p: = 1;
for i: = 1 to n do p: = p * i;
write (n, '! = ',p);
readln
end.
```



Задача 3.

Составить программу вычисления значения выражения $y=1+1/2+1/3+\dots +1/20$.

{В данном случае целесообразно организовать цикл с параметром, изменяющимся от 1 до 20, то есть шаг изменения параметра равен +1. Обозначим: y - очередное значение суммы дробей; n - параметр цикла. Учитывая это, составим программу: }

```
program z3;
uses crt;
var n: integer;
y: real;
begin
clrscr;
{Задаем начальное значение, равное нулю.}
y:=0;
{Организовываем цикл с параметром от 1 до 20.}
for n:=1 to 20 do
begin
{Находим очередную сумму.}
```



```
y:=y+1/n;  
{Выводим на экран очередную сумму.}  
writeln('y=',y);  
end.;  
end.
```

Оператор цикла с предусловием

Если число повторений заранее неизвестно, а задано лишь условие его повторения (или окончания), то используются операторы **while** и **repeat**. Оператор While часто называют оператором цикла с предусловием. Так как проверка условия выполнения цикла производится в самом начале оператора.

*цикл пока <условие>

**<действие>

* конец цикла

Общий вид: While <условие продолжения повторений> do
 <тело цикла>;

Тело цикла – простой или составной оператор или операторы. Если операторов в теле цикла несколько, то тело цикла заключается в операторные скобки begin...end. Ключевые слова While и do означают соответственно "пока" и "выполнять". Когда программа в процессе выполнения впервые достигает оператора while, осуществляется проверка истинности условия. Если условие истинно, то выполняется тело цикла (оператор). После этого происходит возврат к началу фрагмента while do, где проверка условия осуществляется вновь. Цикл будет выполняться до тех пор, пока логическое выражение будет истинным. Как только логическое выражение станет ложным, управление передается следующему за циклом оператору. Если при первом выполнении цикла значение логического оператора будет "ложь", то цикл не станет выполняться, а управление сразу же передается следующему за while оператору.

Следует знать:

- число повторений операторов цикла while определяется в ходе работы программы и, как правило неизвестно;

- после слова `while` записывается условие продолжения выполнения инструкций цикла;
- условие – это выражение логического типа: простое выражение отношения или сложное выражение отношения, которое может принимать одно из двух значений: `true` или `false`;
- для успешного завершения цикла `while` в его теле обязательно должны присутствовать инструкции, оказывающие влияние на условие выполнения инструкций цикла.



Задача 1. Найти сумму 10 произвольных чисел.

```

Program z1;
Const
N=10;
Var k, x, s: integer;
Begin
k:=0; s:=0; {k- количество введенных чисел}
while k < n do
  begin
    k:=k+1;
    write('Введите число');
    readln(x);
    s:=s+x;
  end;
writeln('Сумма чисел равна', s);
end.

```



Задача 2.

Дано натуральное число n . Посчитать количество цифр в числе. *{Подсчет количества цифр начнем с последней цифры числа. Увеличим счетчик цифр на единицу. Число уменьшим в 10 раз (тем самым мы избавляемся от последней цифры числа). Далее с получившимся числом сделаем ту же последовательность*

действий и т. д., пока число не станет равным нулю. *Примечание: в теле цикла обязательно должен быть оператор, влияющий на соблюдение условия, в противном случае произойдет заикливание.* }

```
program z2;
  uses crt;
  var m, n: longint;
  k: integer;  {счетчик цифр}
begin
  clrscr;
  {Вводим целое число.}
  writeln('Введите N='); readln(n);
  {Запоминаем его и счетчику цифр присваиваем начальное значение.}
  m:=n;k:=0;
  {Пока m<>0 делать цикл.}
  while m <>0 do
  begin
    {"Уменьшаем" число на последнюю цифру, т. е. в 10 раз.}
    k:=k+1; m:= m div 10; {Inc(k) }
  end;
  {Вывод количества цифр.}
  writeln('В числе ',n,'- ',k, ' цифр!');
  readln;
end.
```



Задача 3.

Вычислить наибольший общий делитель двух натуральных чисел А и В.

```
program z3;
  var a, b: integer;
begin
  write ('введите два натуральных числа');
  readln (a, b);
```

```

while a<>b do
if a>b then a: = a - b
else b: = b - a;
write ('НОД = ', a);
readln;
end.

```



Задача 4.

Составление таблицы значений функции $y = \sin x$ отрезке $[0;3.14]$ с шагом 0,1.

```

program z4;
var x, y: real;
begin
x: = 0;
writeln ('x' : 10,'sinx' : 10);
while x <= 3.14 do
begin
y: = sin(x);
writeln (x : 10,' ',y : 7 : 5);
x: = x + 0.1
end;
readln
end.

```

Циклы с постусловием

Оператор цикла `repeat` аналогичен оператору `while`, но отличается от него, во-первых, тем, что условие проверяется после очередного выполнения операторов тела цикла и таким образом гарантируется хотя бы однократное выполнение цикла. Во-вторых, тем, что критерием прекращения цикла является равенство выражения константе `true`. За это данный оператор часто называют циклом с постусловием, так как он прекращает выполняться, как только условие, записанное после слова `until`, выполнится. Оператор цикла `repeat` состоит из заголовка, тела и условия окончания.

*цикл с постусловием

**<действие>

* конец цикла, если <условие>

Общий вид: Repeat

<оператор>

... ..

<оператор>

until <условие окончания цикла>

Вначале выполняется тело цикла, затем проверяется условие выхода из цикла. В любом случае этот цикл выполняется хотя бы один раз. Если условие не выполняется, т.е. результатом выражения является False, то цикл активизируется еще раз. Если условие выполнено, то происходит выход из цикла. Использование операторных скобок, в случае, если тело цикла состоит из нескольких операторов, не требуется.

Следует знать:

- число повторений операторов цикла repeat определяется в ходе работы программы и, как правило неизвестно;
- инструкции цикла repeat будут выполняться, пока условие, стоящее после until, будет оставаться ложным;
- после слова until записывается условие завершения цикла;
- условие – это выражение логического типа: простое выражение отношения или сложное выражение отношения, которое может принимать одно из двух значений: true или false;
- для успешного завершения цикла repeat в его теле обязательно должны быть инструкции, выполнение которых влияет на условие завершения цикла, иначе цикл будет выполняться бесконечно – программа заикнется. Другими словами, переменная, которая участвует в условии выхода из цикла, обязательно должна изменяться в теле цикла.



Задача 1. Составить программу, которая вводит и суммирует целые числа. Если введено значение 999, то на экран выводится результат суммирования.

Program z1;

Var x, s:integer;

```

Begin
S:=0;
Repeat
  Write('Ввести число');
  Readln(x);
  If x<>999 then s:=s+x;
Until x=999;
Writeln('Сумма введенных чисел', s);
End.

```



Задача 2.

Использование цикла repeat для подсчета суммы вводимых чисел до первого отрицательного числа

```

Program z2;
var
sum, a: real; {sum - для накопления суммы, a - для очередного числа}
begin
sum:=0;
{Обнуляем сумму}a:=0;
repeat
sum:=sum+a; {Добавляем введенное число к сумме}
write('Введите число');
{Ввод очередного числа}
readln(a)
until a < 0 {Проверяем введенное число на отрицательность}
{При выходе из цикла выполняется этот оператор:}
writeln ('',sum:5:3);
readln
end.

```



Задача 3.

Составить программу планирования закупки товара в магазине на сумму, не превышающую заданную величину.

{Обозначим через x , k - соответствующую цену и количество товара, через p - заданную предельную сумму, через s - общую стоимость покупки. Начальное значение общей стоимости покупки (s) равно нулю. Значение предельной суммы считывается с клавиатуры. Необходимо повторять запрос цены и количества выбранного товара, вычислять его стоимость, суммировать ее с общей стоимостью и выводить результат на экран до тех пор, пока она не превысит предельную сумму p . В этом случае на экран нужно вывести сообщение о превышении: }

```
program z3;
uses crt;
var c, k, p, s: integer;
begin
clrscr;
{Вводим наличие денежных средств.}
writeln('Пред. сумма - ');
readln(p);
{Начальное значение стоимости покупки равно нулю.}
s:=0;
repeat
{Вводим цену товара и его количество.}
writeln('цена тов. и его кол ');
readln (c, k);
{Находим стоимость приобретенных товаров.}
s:=s+c*k;
{Выводим на экран стоимость приобретенных товаров.}
writeln('Стоимость покупки = ', s);
{Повторяем до тех пор, пока стоимость товара не превысит наличия денежных средств.}
until s>p
```

```
writeln('Суммарная стоимость покупки превысила предельную сумму');  
readln;  
end.
```



Задача 4.

Написать программу нахождения наибольшего общего делителя (НОД) двух натуральных чисел.

```
program z4;  
uses crt;  
var x, y: integer;  
begin  
  clrscr;  
  {Вводим два целых неотрицательных числа.}  
  writeln('x:=,y: = ');readln(x,y);  
  repeat  
    if x>y then x:=x mod y else y:=y mod x;  
    {До тех пор, пока одно из чисел не станет равно нулю.}  
  until (x=0) Or (y=0);  
  {Вывод НОД - без условного оператора, так как одно из чисел равно нулю.}  
  writeln('НОД=',x+y));  
  readln;  
end.
```

Вложенные циклы

В теле любого оператора цикла могут находиться другие операторы цикла. При этом цикл, содержащий в себе другой, называют внешним, а цикл, находящийся в теле первого – внутренним (вложенным). Правила организации внешнего и внутреннего циклов такие же, как и для простого цикла.

При программировании вложенных циклов необходимо соблюдать дополнительное условие: все операторы внутреннего цикла должны полностью располагаться в теле внешнего цикла.



Задача 1.

Даны натуральные числа n и k . Составить программу вычисления выражения $1^k+2^k+\dots+n^k$.

{Для вычисления указанной суммы целесообразно организовать цикл с параметром i , в котором, во-первых, вычислялось бы очередное значение $y=i^k$ и, во-вторых, осуществлялось бы накопление суммы прибавлением полученного слагаемого к сумме всех предшествующих ($s = s + y$).}

```
program z1;
uses crt;
var n, k, y, i, s, m: integer;
begin
clrscr;
writeln ('n= k='); readln(n, k);
s:=0;
for i:=1 to n do
begin
y:=1;
for m:=1 to k do begin
{Нахождение степени k числа i.}
y: = y*i;
end;
{Нахождение промежуточной суммы.}
s:=s+y;
end;
writeln(' Ответ: ',s);
readln;
end.
```



Задача 2.

Вычислить сумму $1^1+2^2+\dots+n^n$.

```
program z2;
uses crt;
var n, k, y, i, s, m: integer;
```

```

begin
clrscr;
writeln ('n= k='); readln(n, k);
s:=0;
for i:=1 to n do
begin
y:=1;
for m:=1 to i do
begin
{Нахождение степени k числа i.}
y: = y*i;
end;
{Нахождение промежуточной суммы.}
s:=s+y;
end;
writeln(' Ответ: ',s);
readln;
end.

```



Задача 3.

Написать программу, которая находит и выводит на печать все четырехзначные $abcd$, числа a, b, c, d - различные цифры, для которых выполняется: $ab-cd=a+b+c+d$.

{Задачу можно решать несколькими способами. Одним из возможных способов является перебор всех четырехзначных чисел и проверка для каждого из них выполнения условий. Попробуем сократить перебор, для этого преобразуем второе условие:

$$10a+b-(10c+d)=a+b+c+d;$$

$$9(a-c)=2(c+d);$$

$$(a-c)/(c+d)=2/9$$

Проанализировав первое условие, получаем, что $a=c+2$, $d=9-c$, следовательно $0 \leq c \leq 7$.)

```
program z3;
  uses crt;
  var a,b,c,d: integer;
  begin
    clrscr;
    for c:=0 to 7 do
      begin
        a:=c+2; d:=9-c;
        for b:=0 to 9 do
          begin
            if (b<>c)and(b<>a)and(b<>d)
            then
              write(a,b,c,d);
              writeln
            end;
          end;
        readln;
      end.
```



Задача 4.

Если мы сложим все цифры какого-либо числа, затем все цифры найденной суммы и будем повторять много раз, мы, наконец, получим однозначное число (цифру), называемое цифровым корнем данного числа. Например, цифровой корень числа 34697 равен 2 ($3+4+6+9+7=29$; $2+9=11$; $1+1=2$). Составим программу для нахождения цифрового корня натурального числа.

```
program z4;
  uses crt;
  var n, k, s: longint;
  begin
    clrscr;
```

```

writeln(' число='); readln(n);
s:=n;
{Пока сумма является двузначным числом.}
while s>9 do
begin
k:=s;s:=0;
{Вычисляем сумму цифр числа.}
repeat
S:=s+k mod 10; k:=k div 10;
until k=0;
end;
writeln(' цифр. корень числа ',n, ' равен ',s);
readln;
end.

```

Задачи для самостоятельного решения:

Вариант № 1

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{1+i}{2+i^2}$ (три различные программы)

2. Вычисление $p = n!$ (n факториал)

3. Приближенное вычисление суммы бесконечно убывающего ряда $1+x/1!+x^2/2!+x^3/3!+ \dots$

Вариант № 2

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{1}{4+i^3}$ (три различные программы)

2. Использование цикла repeat для подсчета суммы вводимых чисел до первого отрицательного числа

3. Вычислить сумму $1+2+...+nn$.

Вариант № 3

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{3i^2}{24+i}$ (три различные программы)

2. Составить программу вычисления значения выражения $y=1+1/2+1/3+\dots+1/20$.

3. Составление таблицы значений функции $y = \sin x$ отрезке $[0;3.14]$ с шагом 0,1

Вариант № 4

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{8+3i^2}{12+i^2}$ (три различные программы)

2. Составить программу, которая вводит и суммирует целые числа. Если введено значение 999, то на экран выводится результат суммирования

3. Написать программу, которая находит и выводит на печать все четырехзначные abcd, числа a, b, c, d - различные цифры, для которых выполняется: $ab-cd=a+b+c+d$

Вариант № 5

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{8}{16+i^2}$ (три различные программы)

2. Из чисел от 10 до 99 вывести те, сумма цифр которых равна S ($0 < S < 18$).

3. Вычислить наибольший общий делитель двух натуральных чисел A и B

Вариант № 6

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{3+i}{20+i}$ (три различные программы)

2. Составить программу планирования закупки товара в магазине на сумму, не превышающую заданную величину

3. Даны натуральные числа n и k . Составить программу вычисления выражения $lk+2k+\dots+nk$.

Вариант № 7

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{12-i^2}{6+i}$ (три различные программы)

2. Вывести на экран натуральные числа от 1 до 9 в обратном порядке.

3. Найти сумму 10 произвольных чисел

Вариант № 8

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{3-i}{2+i^2}$ (три различные программы)

2. Написать программу нахождения наибольшего общего делителя (НОД) двух натуральных чисел

3. Если мы сложим все цифры какого-либо числа, затем все цифры найденной суммы и будем повторять много раз, мы, наконец, получим однозначное число (цифру), называемое цифровым корнем данного числа. Например, цифровой корень числа 34697 равен 2 ($3+4+6+9+7=29$; $2+9=11$; $1+1=2$). Составим программу для нахождения цифрового корня натурального числа

Вариант № 9

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{2i^2}{4+i}$ (три различные программы)

2. Дано натуральное число n ($1000 \leq n \leq 9999$). Определить, является ли оно палиндромом ("перевертышем"), с учетом четырех цифр. Например, палиндромами являются числа: 2222, 6116, 1441

3. Дано натуральное число n . Посчитать количество цифр в числе

Вариант № 10

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{2i}{3+i}$ (три различные программы)

2. Составить программу, которая вводит и суммирует целые числа. Если введено значение 999, то на экран выводится результат суммирования

3. Вычислить сумму $11+22+\dots+nn$.

Вариант № 11

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{1+i}{2+i^2}$ (три различные программы)

2. Использование цикла repeat для подсчета суммы вводимых чисел до первого отрицательного числа

3. Составить программу вычисления значения выражения $y=1+1/2+1/3+\dots+1/20$.

Вариант № 12

1. Составить 3 варианта программ циклической структуры типа for, while и repeat, откомпилировать их, ввести исходные данные, сравнить полученные результаты. $\sum_{i=1}^N \frac{6}{4+i^3}$ (три различные программы)

2. Из чисел от 10 до 99 вывести те, сумма цифр которых равна $S(0 < S < 18)$.

3. Найти сумму 10 произвольных чисел

Контрольные вопросы

1 Что такое циклический алгоритм?

2 Какие обозначения используются в графическом представлении алгоритма для обозначения цикла?

3 Какие бывают циклы?

4 Что такое и когда используется цикл с предусловием?

5 Какой оператор в языке Паскаль используется для представления цикла с предусловием?

- 6 Как в блок-схемах изображаются цикл с предусловием?
- 7 Особенности использования цикла с предусловием.
- 8 Что такое и когда используется цикл с постусловием?
- 9 Какой оператор в языке Паскаль используется для представления цикла с постусловием?
- 10 Как в блок-схемах изображаются цикл с постусловием?
- 11 Особенности использования цикла с постусловием.
- 12 Что такое и когда используется цикл с параметром?
- 13 Какой оператор в языке Паскаль используется для представления цикла с параметром?
- 14 Как в блок-схемах изображаются цикл с параметром?

Практическая работа №11, №12

Тема: Решение задач с использованием одномерных массивов.

Теория.

В повседневной и научной практике часто приходится встречаться с информацией, представленной в табличной форме. Вот, например, таблица, содержащая значения температуры, за определенный год.

Месяц										0	1	2
Температура	21	18	7,5	,6	0	8	2,2	4	7	,4	7	18

Такую таблицу называют линейной. Она представляет собой последовательность упорядоченных чисел. Если требуется какая-то математическая обработка данных, то для их обозначения вводят индексную символику.

Например:

T_1 – температура января;

T_5 – температура мая;

В общем виде множество значений, содержащихся в таблице, можно обозначить так: $\{ T_i \}$, $i = 1, 2, 3, \dots, 12$.

Порядковые номера элементов называются индексами. Индексированные величины удобно использовать для записи их математической обработки:

$T_{cp} = \frac{1}{12} \sum_{i=1}^{12} T_i$. По данной формуле можно подсчитать среднегодовую температуру.

В языке Паскаль для записи линейных таблиц используют структурированный тип данных, который называется **одномерным массивом**. Одномерный массив представляет собой совокупность пронумерованных однотипных значений, имеющих общее имя. Элементы массива обозначаются переменными с индексами. Индексы записывают в квадратных скобках после имени массива.

Например: $T[1]$, $T[5]$, $T[I]$ и т.п.

Описание одномерных массивов:

Переменная типа одномерный массив описывается в разделе описания переменных в следующем виде:

var имя массива: **array[1..n] of** тип;

n – количество элементов массива.



Пример: var T: array [1..12] of real;

Обработка массивов в программах производится поэлементно. Обращаются к элементам массива, указывая индекс.

Ввод массива (чтение массива):

```
writeln('введите ',n,' элементов массива');  
for i:=1 to 12 do  
  readln (t[i]);
```

Вывод массива:

```
write ln('массив');  
for i:=1 to 12 do  
  write ln ('t[',i,']= ', t[i]:0:2);
```



Примеры задач.

1. Известны данные о среднемесячной температуре за год. Требуется вычислить среднегодовую температуру, а также ежемесячные отклонения от этой величины.

```
Program Example;  
Const n=12;  
Var t, dt: array[1..n] of real;  
ts: real;  
i:integer;
```

```

Begin {ввод исходных данных}
write ln ('введите температуру за ',n,' месяцев');
for i:=1 to n do
read ln (t[i]);
{вычисление средней температуры}
ts:=0;
for i:=1 to n do
ts:=ts+t[i];
ts:=ts/n;
{вычисление таблицы отклонений от среднего}
for i:=1 to n do
dt[i]:=t[i]-ts;
{вывод результатов}
write ln ('среднегодовая температура =' , ts:0:2);
write ln ('отклонения от среднегодовой температуры:');
for i:=1 to n do
write ln (i,' - месяц: ', dt[i]:0:2);
End.

```



2. Дан массив из 8 элементов целого типа. Найти сумму положительных элементов массива.

```

Program summa;
Const n=8;
Var a:array[1..n] of integer;
i,s: integer;
Begin {ввод исходных данных}
writeln ('введите ',i,' целых чисел');
for i:=1 to n do
readln (a[i]);
s:=0;
for i:=1 to n do

```

```

if a[i]>0
then s:=s+a[i];
writeln ('сумма положительных чисел =',s);
End.

```



3. Задать массив В из 12 элементов по следующей формуле:

$$b_i = \begin{cases} i^2 - 15, & i < 6 \\ i + 8, & i \geq 6 \end{cases} .$$


Вывести полученный массив. Найти максимальный элемент

массива.

```


Program maximum;
Const n=12;
Var b:array[1.. n] of integer;
i,max: integer;
Begin {получение массива}
for i:=1 to n do
if i<6
then b[i]:=sqr(i)-15
else b[i]:=i+8;
{Вывод массива}
writeln('массив');
for i:=1 to n do
writeln ('b[',i,']= ', b[i]);
{поиск максимального элемента массива}
max:=b[1];
for i:=1 to n do
if b[i]>max
then max:=b[i];
writeln ('максимум=',max);
End.

```

4.  Получить массив С из 10 элементов следующего вида: $c_i = \frac{i^3 - 10i}{3}$.

Вывести полученный массив. Сосчитать количество отрицательных элементов массива.

```
Program kolvo;
Const n=10;
Var c:array[1.. n] of real;
i,k:integer;
Begin {получение массива}
for i:=1 to n do
c[i]:= (i*sqr(i)-10*i)/3;
{вывод массива}
writeln('массив');
for i:=1 to n do
writeln ('c[',i,']= ', c[i]:0:2);
{подсчет количества отрицательных}
k:=0;
for i:=1 to n do
if c[i]<0
then k:=k+1;
writeln ('количество отрицательных чисел=',k);
End.
```

5.  Дан массив из 10 элементов целого типа. Заменить отрицательные элементы массива на их квадрат, положительные оставить без изменения. Вывести полученный массив.

```
Program zamena;
Const n=10;
Var a:array[1.. n] of integer;
```

```

i:integer;
Begin { ввод исходных данных }
writeln ('введите ',i,' целых чисел');
for i:=1 to n do
readln (a[i]);
{замена отрицательных чисел}
fori:=1 tondo
if a[i]<0
then a[i]:=sqr(a[i]);
{вывод массива}
writeln('массив');
for i:=1 to n do
writeln ('a[' ,i,']=', a[i]);
End.

```

Задания для самостоятельного решения:

Вариант № 1

1. Дан массив из 10 целых чисел. Найти количество отрицательных элементов массива.
 2. Дан массив из 8 чисел. Заменить все положительные числа на 0. Отрицательные числа оставить без изменения. Вывести элементы массива.
 3. Дано натуральное число n. Задать массив из n элементов по формуле $a_i = \frac{i^2 - 3}{2}$. Вывести элементы массива. Найти минимальный элемент массива
 4. Составить программу, которая формирует случайным образом одномерный массив из N элементов и выводит те его элементы, которые встречаются только один раз
-

Вариант № 2

1. Дан массив из 8 чисел. Найти максимальный элемент массива.

2. Дан массив из 6 целых чисел. Умножить все отрицательные числа на 5. Вывести элементы массива.
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = \cos i$. Вывести элементы массива. Найти сумму положительных элементов массива
 4. Составить программу, которая формирует случайным образом одномерный массив из N элементов и выводит повторяющиеся элементы (примечание: повторяющийся элемент должен выводиться **один** раз независимо от того, сколько раз он встречается в массиве)
-

Вариант № 3

1. Дан массив из 10 целых чисел. Найти минимальный элемент массива.
 2. Дан массив из 7 чисел. Заменить все отрицательные числа на их модуль. Вывести элементы массива.
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = e^{i+2} - 4$. Вывести элементы массива. Найти количество отрицательных элементов массива
 4. Составить программу, которая формирует случайным образом два одномерных массива из N элементов и выводит элементы первого массива, которые не встречаются во втором
-

Вариант № 4

1. Дан массив из 8 чисел. Найти сумму отрицательных элементов массива.
2. Дан массив из 9 целых чисел. Умножить все отрицательные числа на -3. Вывести элементы массива.
3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = \frac{|i-7|}{4}$. Вывести элементы массива. Найти минимальный элемент массива
4. Составить программу, которая формирует случайным образом два одномерных массива из N элементов и выводит элементы, встречающиеся как в первом, так и во втором массивах (примечание:

элемент должен выводиться один раз независимо от того, сколько раз он встречается в массивах)

Вариант № 5

1. Дан массив из 14 целых чисел. Найти количество отрицательных элементов массива.
 2. Дан массив из 8 чисел. Заменить все отрицательные числа на 1. Положительные числа оставить без изменения. Вывести элементы массива.
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = \frac{i^2 - 6}{2}$. Вывести элементы массива. Найти максимальный элемент массива.
 4. Составить программу, которая формирует случайным образом два одномерных массива из N элементов и формирует третий массив из элементов, которые встречаются только в одном из массивов
-

Вариант № 6

1. Дан массив из 12 целых чисел. Найти количество положительных элементов массива.
 2. Дан массив из 9 чисел. Заменить все отрицательные числа на 0. Положительные числа оставить без изменения. Вывести элементы массива.
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = \frac{i^2 - 5}{4}$. Вывести элементы массива. Найти минимальный элемент массива
 4. Найти среднее арифметическое всех элементов массива
-

Вариант № 7

1. Дан массив из 18 чисел. Найти минимальный элемент массива.

2. Дан массив из 8 целых чисел. Умножить все отрицательные числа на 6. Вывести элементы массива.
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = 2 \sin i$. Вывести элементы массива. Найти сумму отрицательных элементов массива.
 4. Определить, сколько раз встречается число 7 среди элементов массива
-

Вариант № 8

1. Дан массив из 12 целых чисел. Найти максимальный элемент массива.
 2. Дан массив из 9 чисел. Заменить все отрицательные числа на их модуль. Вывести элементы массива.
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = e^{i+2} - 14$. Вывести элементы массива. Найти количество положительных элементов массива.
 4. Определить, сколько элементов массива меньше, чем число 6
-

Вариант № 9

1. Дан массив из 17 чисел. Найти сумму отрицательных элементов массива.
 2. Дан массив из 8 целых чисел. Умножить все отрицательные числа на -5. Вывести элементы массива.
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = \frac{|i-7|}{5}$. Вывести элементы массива. Найти минимальный элемент массива
 4. Определить, сколько элементов массива больше, чем число 3
-

Вариант № 10

1. Дан массив из 14 целых чисел. Найти минимальный элемент массива.
2. Дан массив из 10 чисел. Заменить все отрицательные числа на их модуль. Вывести элементы массива.

3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = e^{i+2} - 12$. Вывести элементы массива. Найти количество отрицательных элементов массива
 4. Найти сумму всех неотрицательных элементов массива
-

Вариант № 11

1. Дан массив из 10 целых чисел. Найти количество отрицательных элементов массива
 2. Дан массив из 6 целых чисел. Умножить все отрицательные числа на 5. Вывести элементы массива
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = e^{i+2} - 4$. Вывести элементы массива. Найти количество отрицательных элементов массива
 4. Найти наименьший по модулю элемент массива
-

Вариант № 12

1. Дан массив из 8 чисел. Найти сумму отрицательных элементов массива
 2. Дан массив из 8 чисел. Заменить все отрицательные числа на 1. Положительные числа оставить без изменения. Вывести элементы массива
 3. Дано натуральное число n . Задать массив из n элементов по формуле $a_i = \frac{i^2 - 5}{4}$. Вывести элементы массива. Найти минимальный элемент массива
 4. Найти наибольший из элементов массива, имеющих нечетные индексы
-

Контрольные вопросы

1. Как осуществляется вывод массива Паскаля?
2. Какие операции можно производить с элементами массива?
3. Что такое одномерный массив?
4. Какие характеристики имеет массив?

Практическая работа № 13

Тема: Двумерные массивы

Цель: Изучить структуру двумерных массивов

Ход работы:

Теоретическое обоснование

Двумерный массив в Паскале трактуется как одномерный массив, тип элементов которого также является массивом (массив массивов). Положение элементов в двумерных массивах Паскаля описывается двумя индексами. Их можно представить в виде прямоугольной таблицы или матрицы.

Рассмотрим двумерный массив Паскаля размерностью 3*3, то есть в ней будет три строки, а в каждой строке по три элемента:

Каждый элемент имеет свой номер, как у одномерных массивов, но сейчас номер уже состоит из двух чисел – номера строки, в которой находится элемент, и номера столбца. Таким образом, номер элемента определяется пересечением строки и столбца. Например, а 21 – это элемент, стоящий во второй строке и в первом столбце.

Описание двумерного массива Паскаля.

Существует несколько способов объявления двумерного массива Паскаля.

Мы уже умеем описывать одномерные массивы, элементы которых могут иметь любой тип, а, следовательно, и сами элементы могут быть массивами. Рассмотрим следующее описание типов и переменных:

Пример описания двумерного массива Паскаля

Type

```
Vector = array [1..5] of <тип_элементов>;
```

```
Matrix= array [1..10] of vector;
```

```
Var m: matrix;
```

Мы объявили двумерный массив Паскаля m, состоящий из 10 строк, в каждой из которых 5 столбцов. При этом к каждой i -й строке можно обращаться m [i], а каждому j -му элементу внутри i -й строки – m [i , j].

Определение типов для двумерных массивов Паскаля можно задавать и в одной строке:

Type

```
Matrix= array [1..5] of array [1..10] of <тип элементов>;
```

или еще проще:

```
type
```

```
matrix = array [1..5, 1..10] of<тип элементов>;
```

Обращение к элементам двумерного массива имеет вид: $M [i , j]$. Это означает, что мы хотим получить элемент, расположенный в i -й строке и j -м столбце. Тут главное не перепутать строки со столбцами, а то мы можем снова получить обращение к несуществующему элементу. Например, обращение к элементу $M [10, 5]$ имеет правильную форму записи, но может вызвать ошибку в работе программы.

Основные действия с двумерными массивами Паскаля

Все, что было сказано об основных действиях с одномерными массивами, справедливо и для матриц. Единственное действие, которое можно осуществить над однотипными матрицами целиком – это присваивание. Т.е., если в программе у нас описаны две матрицы одного типа, например,

```
type
```

```
matrix= array [1..5, 1..10] of integer;
```

```
var
```

```
a , b : matrix ;
```

то в ходе выполнения программы можно присвоить матрице a значение матрицы b ($a := b$). Все остальные действия выполняются поэлементно, при этом над элементами можно выполнять все допустимые операции, которые определены для типа данных элементов массива. Это означает, что если массив состоит из целых чисел, то над его элементами можно выполнять операции, определенные для целых чисел, если же массив состоит из символов, то к ним применимы операции, определенные для работы с символами.

Ввод двумерного массива Паскаля.

Для последовательного ввода элементов одномерного массива мы использовали цикл `for`, в котором изменяли значение индекса с 1-го до последнего. Но положение элемента в двумерном массиве Паскаля определяется двумя

индексами: номером строки и номером столбца. Это значит, что нам нужно будет последовательно изменять номер строки с 1-й до последней и в каждой строке перебирать элементы столбцов с 1-го до последнего. Значит, нам потребуется два цикла `for`, причем один из них будет вложен в другой.

Рассмотрим пример ввода двумерного массива Паскаля с клавиатуры:

Пример программы ввода двумерного массива Паскаля с клавиатуры

```
type
matrix= array [1..5, 1..10] of integer;
var
a, : matrix;
i, j: integer; { индексы массива }
begin
fori :=1 to 5 do {цикл для перебора всех строк}
forj :=1 to 10 do {перебор всех элементов строки по столбцам}
readln( a [ i , j ]); {ввод с клавиатуры элемента, стоящего в i -й строке и j -м
столбце}
```

Двумерный массив Паскаля можно заполнить случайным образом, т.е. использовать функцию `random (N)`, а также присвоить каждому элементу матрицы значение некоторого выражения. Способ заполнения двумерного массива Паскаля выбирается в зависимости от поставленной задачи, но в любом случае должен быть определен каждый элемент в каждой строке и каждом столбце.

Вывод двумерного массива Паскаля на экран.

Вывод элементов двумерного массива Паскаля также осуществляется последовательно, необходимо напечатать элементы каждой строки и каждого столбца. При этом хотелось бы, чтобы элементы, стоящие в одной строке, печатались рядом, т.е. в строку, а элементы столбца располагались один под другим. Для этого необходимо выполнить следующую последовательность действий (рассмотрим фрагмент программы для массива, описанного в предыдущем примере):



Пример программы вывода двумерного массива Паскаля

```
fori :=1 to 5 do {цикл для перебора всех строк}
```

```

begin
forj :=1 to 10 do {перебор всех элементов строки по столбцам}
write( a [ i , j ]:4); {печать элементов, стоящих в i -й строке матрицы в одной
экранной строке, при этом для вывода каждого элемента отводится 4 позиции}
writeln ; {прежде, чем сменить номер строки в матрице, нужно перевести
курсор на начало новой экранной строки}
end ;

```

Замечание (это важно!): очень часто в программах студентов встречается ошибка, когда ввод с клавиатуры или вывод на экран массива пытаются осуществить следующим образом: `readln (a)`, `writeln (a)`, где `a` – это переменная типа массив. При этом их удивляет сообщение компилятора, что переменную этого типа невозможно считать или напечатать. Может быть, вы поймете, почему этого сделать нельзя, если представите N кружек, стоящих в ряд, а у вас в руках, например, чайник с водой. Можете вы по команде «налей воду» наполнить сразу все кружки? Как бы вы ни старались, но в каждую кружку придется наливать отдельно. Заполнение и вывод на экран элементов массива также должно осуществляться последовательно и поэлементно, т.к. в памяти ЭВМ элементы массива располагаются в последовательных ячейках.

Представление двумерного массива Паскаля в памяти

Элементы абстрактного массива в памяти машины физически располагаются последовательно, согласно описанию. При этом каждый элемент занимает в памяти количество байт, соответствующее его размеру. Например, если массив состоит из элементов типа `integer`, то каждый элемент будет занимать по два байта. А весь массив займет S^2 байта, где S – количество элементов в массиве.

А сколько места займет массив, состоящий из массивов, т.е. матрица? Очевидно: $S_i \times S_j$, где S_i - количество строк, а S_j – количество элементов в каждой строке. Например, для массива типа

```
Matrix = array [1..3, 1..2] of integer ;
```

потребуется 12 байт памяти.

Как будут располагаться в памяти элементы этого массива? Рассмотрим схему размещения массива `M` типа `matrix` в памяти.

Под каждый элемент $M [i,j]$ типа `integer` выделяется две ячейки памяти. Размещение в памяти осуществляется «снизу вверх». Элементы размещаются в порядке изменения индекса, что соответствует схеме вложенных циклов: сначала размещается первая строка, затем вторая, третья... Внутри строки по порядку идут элементы: первый, второй и т.д.

Как мы знаем, доступ к любой переменной возможен, только если известен адрес ячейки памяти, в которой хранится переменная. Конкретная память выделяется для переменной при загрузке программы, то есть устанавливается взаимное соответствие между переменной и адресом ячейки. Но если мы объявили переменную как массив, то программа «знает» адрес начала массива, то есть первого его элемента. Как же происходит доступ ко всем другим элементам массива? При реальном доступе к ячейке памяти, в которой хранится элемент двумерного массива, система вычисляет ее адрес по формуле:

$$\text{Addr} + \text{SizeElem} * \text{Cols} * (I - 1) + \text{SizeElem} * (J - 1),$$

где `Addr` – фактический начальный адрес, по которому массив располагается в памяти; `I`, `J` – индексы элемента в двумерном массиве; `SizeElem` – размер элемента массива (например, два байта для элементов типа `integer`); `Cols` – количество элементов в строке.

Выражение $\text{SizeElem} * \text{Cols} * (I - 1) + \text{SizeElem} * (J - 1)$ называют смещением относительно начала массива.

Сколько памяти выделяется для массива?

Рассмотрим не столько вопрос о том, сколько памяти выделяется под массив (это мы разобрали в предыдущем разделе), а о том, каков максимально допустимый размер массива, учитывая ограниченный объем памяти.

Для работы программы память выделяется сегментами по 64 Кбайт каждый, причем как минимум один из них определяется как сегмент данных. Вот в этом-то сегменте и располагаются те данные, которые будет обрабатывать программа. Ни одна переменная программы не может располагаться более чем в одном сегменте. Поэтому, даже если в сегменте находится только одна переменная, описанная как массив, то она не сможет получить более чем 65536 байт. Но почти наверняка, кроме массива в сегменте данных будут описаны еще некоторые переменные,

поэтому реальный объем памяти, который может быть выделен под массив, находится по формуле: $65536 - S$, где S – объем памяти, уже выделенный под другие переменные.

Зачем нам это знать? Для того чтобы не удивляться, если при компиляции транслятор выдаст сообщение об ошибке объявления слишком длинного массива, когда в программе встретит описание (правильное с точки зрения синтаксиса):

```
Type myArray= array [1..50000] of integer;
```

Вы уже знаете, что, учитывая двухбайтовое представление целых чисел, реально можно объявить массив с количеством элементов равным $65536/2 - 1 = 32767$. И то лишь в том случае, если других переменных не будет. Двумерные массивы должны иметь еще меньшие границы индексов.

Задания для самостоятельного решения

Вариант № 1

1. Сформировать с помощью датчика случайных чисел и вывести на экран матрицу, размером $M \times N$. Элементы задаются на интервале $[-20, 25]$.
2. В двумерном массиве, состоящем из n целых чисел, найти сумму элементов в каждой строке. Размер произвольный.
3. Найти наименьший элемент двумерного массива. Размер $M \times N$. Элементы задаются на интервале $[-30, 45]$.

Вариант № 2

1. В двумерном массиве, состоящем из целых чисел, найти наименьший элемент и номер строки, в которой он находится. Элементы вводятся с клавиатуры. Размер $M \times N$.
 2. Найти сумму элементов в каждой строке двумерного массива, состоящего из целых чисел. Размер $M \times N$. Элементы задаются на интервале $[-19, 30]$.
 3. Подсчитать количество положительных элементов в каждой строке матрицы размером $M \times N$, элементы которой вводятся с клавиатуры.
-

Вариант № 3

1. Сформировать матрицу типа
- | | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

2. Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество отрицательных элементов. Элементы вводятся с клавиатуры.

3. Упорядочить каждый столбец матрицы по возрастанию. Массив размером $M \times N$, элементы которого задаются датчиком случайных чисел на интервале $[-17; 26]$.

Вариант № 4

1. Сформировать матрицу
- | | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |

2. Найти наибольшее нечетное число в матрице размером $M \times N$, элементы которой задаются датчиком случайных чисел на интервале $[-27, 38]$.

3. Найти наименьший элемент двумерного массива. Размер $M \times N$. Элементы задаются на интервале $[-60, 15]$.

Вариант № 5

1. Сформировать с помощью датчика случайных чисел и вывести на экран матрицу, размером $M \times N$. Элементы задаются на интервале $[-10, 45]$.

2. Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество положительных элементов. Элементы вводятся с клавиатуры.

3. Подсчитать количество отрицательных элементов в каждой строке матрицы размером $M \times N$, элементы которой вводятся с клавиатуры.

Вариант № 6

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и выводит его в виде матрицы

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и упорядочивает по возрастанию каждую строку массива

Дан двумерный массив. Определить сумму всех элементов массива

Вариант № 7

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и вычисляет сумму элементов каждой строки

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и упорядочивает по убыванию каждую строку массива

Дан двумерный массив. Определить сумму квадратов всех элементов массива

Вариант № 8

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и находит максимальные элементы каждого столбца

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и упорядочивает по убыванию каждый столбец массива

Дан двумерный массив. Определить произведение всех отрицательных элементов

Вариант № 9

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и вывод элементы под главной диагональю

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и упорядочивает по возрастанию каждый столбец массива

Дан двумерный массив. Определить произведение всех положительных элементов

Вариант № 10

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и выводит его в виде матрицы

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и упорядочивает по возрастанию каждую строку массива

Дан двумерный массив. Определить сумму всех элементов массива

Вариант № 11

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и вычисляет сумму элементов каждой строки

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и упорядочивает по убыванию каждую строку массива

Дан двумерный массив. Определить сумму квадратов всех элементов массива

Вариант № 12

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и находит максимальные элементы каждого столбца

Составить программу, которая формирует случайным образом массив $N \times N$ целых чисел и упорядочивает по убыванию каждый столбец массива

Дан двумерный массив. Определить произведение всех отрицательных элементов

Контрольные вопросы

1. Что такое двумерный массив?
2. Что такое индекс?
3. Как объявляются двумерные массивы?
4. Как можно заполнить двумерный массив?
5. Каким образом можно вывести двумерный массив на экран?
6. Как найти сумму, произведение и количество элементов двумерного массива?
7. Как найти минимальный элемент в двумерном массиве?
8. Как найти максимальный элемент в двумерном массиве?
9. Как вывести двумерный массив в виде таблицы?

Практическая работа № 14

Тема: Строки в Pascal

Цель: Изучить строковый тип в Паскале

Ход работы:

Теоретическое обоснование

Строковый тип данных в Паскале

Строки в Паскале – это данные типа `string`. Они используются для хранения последовательностей символов. В Паскале длина стандартной строки ограничена 255 символами. Под каждый символ отводится по одному байту, в котором хранится код символа. Кроме того, каждая строка содержит еще дополнительный байт, в котором хранится длина строки.

Если заранее известно, что длина строки будет меньше 255 символов, то программист может сам задать максимальную длину строки.

Примеры описания строк:

```
type
    str_type = string[12];
const
    n = 50;
var
    s1: string;
    s2, s3: str_type;
    s4: string[n];
    s5, s6, s7: string[7];
...
```

Длина строки хранится в первом ее байте, индекс которого равен 0.

Объявление типизированной константы для типа `string` осуществляется так:

```
const
    s: string = 'FreePascal'
...
```

Существует понятие пустой строки, т.е. строки, которая не имеет элементов.

Пустая строка обозначается двумя рядом стоящими апострофами (например, `st := ''`).

Операции над строками

Строки можно присваивать друг другу. Если максимальная длина переменной слева меньше длины присваиваемой строки, то лишние символы справа отбрасываются.

```
...  
s1 := 'this is text';  
s2 := s1;
```

...

Строки можно объединять с помощью операции конкатенации, которая обозначается знаком +.

```
...  
s1 := 'John';  
s2 := 'Black';  
s1 := s1 + ' ' + s2;
```

...

Строки можно сравнивать друг с другом с помощью операций отношения. При сравнении строки рассматриваются посимвольно слева направо, при этом сравниваются коды соответствующих пар символов. Строки равны, если они имеют одинаковую длину и посимвольно эквивалентны. В строках разной длины существующий символ всегда больше соответствующего ему отсутствующего символа. Меньшей будет та строка, у которой меньше код первого несовпадающего символа (вне зависимости от максимальных и текущих длин сравниваемых строк).

```
'abc' > 'ab' (true)  
'abc' = 'abc' (true)  
'abc' < 'abc ' (false)
```

Имя строки может использоваться в процедурах ввода-вывода. При вводе в строку считывается из входного потока количество символов, равное длине строки или меньшее, если символ перевода строки (который вводится нажатием клавиши Enter) встретится раньше. При выводе под строку отводится количество позиций, равное ее фактической длине.

...

```
readln (s1);  
write (s1);  
...
```

К отдельному символу строки можно обращаться как к элементу массива символов, например `s1[3]`. Символ строки совместим с типом `char`, их можно использовать в выражениях одновременно, например:

```
...  
s1[3] := 'h';  
writeln (s2[3] + 'r');  
...
```

Можно осуществлять коррекцию любого символа строковой переменной, для чего в соответствующем операторе достаточно указать имя переменной типа `string`, вслед за которым в квадратных скобках задается номер ее элемента (например, `str[3]:= 'j'`).

Элементы строки нумеруются с единицы, т.к. в каждой строковой переменной имеется элемент с номером 0, в котором в виде символа хранится длина текущей строки. Чтобы узнать текущую длину, достаточно применить функцию `ord` к нулевому элементу строки. Например:

```
...  
writeln(ord(st[0]))  
...
```

Нулевой элемент строковой переменной можно корректировать. При этом будет изменяться текущая длина строки.

Например, выражение `str[0]:= #50` устанавливает текущую длину равной 50.

Процедуры и функции для работы со строками

При работе со строками, как правило, возникает необходимость выполнять их копирование, вставку, удаление или поиск. Для эффективной реализации этих действий в Паскале предусмотрены стандартные процедуры и функции. Они кратко описаны ниже.

Функция `Concat (s1, s2, ..., sn)` возвращает строку, являющуюся слиянием строк `s1, s2, ..., sn`.

Функция Copy (s, start, len) возвращает подстроку длиной len, начинающуюся с позиции start строки s.

Процедура Delete (s, start, len) удаляет из строки s, начиная с позиции start, подстроку длиной len.

Процедура Insert (subs, s, start) вставляет в строку s подстроку subs, начиная с позиции start.

Функция Length (s) возвращает фактическую длину строки s, результат имеет тип byte.

Функция Pos (subs, s) ищет вхождение подстроки subs в строку s и возвращает номер первого символа subs в s или ноль, если subs не содержится в s.

Процедуры преобразования типов

Процедура Str (x, s) преобразует числовое значение x в строку s, при этом для x может быть задан формат, как в процедурах вывода write и writeln. Например:

```
x := 123;
```

```
s := str(x:6,s);
```

Результат: s = ' 123'.

Процедура Val (s, x, errcode) преобразует строку s в значение числовой переменной x, при этом строка s должна содержать символьное представление числа. В случае успешного преобразования переменная errcode равна нулю. Если же обнаружена ошибка, то errcode будет содержать номер позиции первого ошибочного символа, а значение x не определено.



Задача: (Символьные величины)

Составить программу, которая определяет количество букв «а» в заданном тексте

```
program fff;  
var a,b:string;  
i,n,k:integer;  
begin  
  readln(a);  
  n:=length(a);  
  for i:=1 to n do
```

```

begin
if a[i]='a' then k:=k+1;
end;
writeln('в слове 'a,' буква А встречается 'k,' раз');
readln;
end.

```



Задача: (Символьные величины)

Составить программу, которая определяет количество слов в заданном тексте при условии, что слова разделены пробелом.

```

program fff;
var a,b:string;
i,n,k:integer;
begin
readln(a);
n:=length(a);
for i:=1 to n do
begin
if a[i]=' ' then k:=k+1;
end;
writeln('в тексте: 'a,' количество слов='k);
readln;
end.

```



Задача: (Массив с символьными переменными)

Составить массив из пяти фамилий, и вывести на экран столбиком, начиная с последней.

```

program fff;
var v:array[1..5] of string;
i:integer;

```



```

begin
writeln('введи пять фамилий');
for i:=1 to 5 do readln(v[i]);
writeln(' фамилии наоборот:');
for i:=5 downto 1 do writeln(v[i]);
readln;
end.

```



Задача: (Массив с символьными переменными)

Составить массив из пяти фамилий, и вывести на экран те из них, которые начинаются с определённой буквы, которая вводится с клавиатуры.

Длину строки можно указать в разделе описания переменных:

<имя переменной, ...>:string[n]

```

program fff;
var v:array[1..5] of string;
k:string[1];
i:integer;
begin
writeln('введи букву, с которой будет начинаться фамилия');
readln(k);
writeln('введи пять фамилий');
for i:=1 to 5 do readln(v[i]);
writeln('интересующие Вас фамилии:');
for i:=1 to 5 do
if (v[i])[1]=k then writeln(v[i]);
readln;
end.

```



Задача: (Массив с символьными переменными)

Из вводимого с клавиатуры слова вырезать каждую третью букву.

```

program aa;
var a:string;

```

```
k,x: integer;
begin
readln(a);
k:=length(a);
x:=3;
while x<=k do
begin
a[x]:=' ';
x:=x+3;
end;
writeln(a);
readln;
end.
```



Задача: (Массив с символьными переменными)

В вводимом слове с клавиатуры заменить все буквы «a» на букву «o».

```
program aa;
var a:string;
k,x: integer;
begin
readln(a);
k:=length(a);
for x:=1 to k do
if a[x]='a' then a[x]:='o';
writeln(a);
readln;
end.
```



Задача: (Массив с символьными переменными)

Заданы фамилия, имя, отчество учащегося, разделённые пробелами.

Напишите программу, печатающую фамилии ученика и его инициалы.

```

program aaa;
uses crt;
var d,r:string;
k,i:integer;
begin
clrscr;
writeln('введи Ф.И.О. ');
readln(d);
k:=length(d);
for i:=1 to k do
begin
if d[i]=' ' then d:=copy(d,1,i)+d[i+1]+'.';
end;
writeln(d);
readln;
end.

```

Все структурированные типы данных, с которыми мы уже познакомились, представляют собой совокупности однотипных величин.

Комбинированный тип данных – это структурированный тип, состоящий из фиксированного числа компонент (полей) разного типа. Комбинированный тип имеет еще и другое название – запись.

Обычно запись содержит совокупность разнотипных атрибутов, относящихся к одному объекту. Например, анкетные сведения о студенте могут быть представлены в виде информационной структуры.

Анкета студента:

Фамилия, Имя, Отчество;

Пол;

Дата рождения;

Адрес;

Курс;

Группа;

В Паскале эта информация может храниться в одной переменной типа Запись. Задать тип и описать соответствующую переменную можно следующим образом:

```
Типе имя типа=record;
```

```
поле 1:тип;
```

```
поле 2:тип;
```

```
...
```

```
поле n:тип;
```

```
End;
```

```
Пр. Type anketa=Record;
```

```
FIO:String[50];
```

```
Pol:Char;
```

```
Dat:String[16];
```

```
Adres:String[50];
```

```
Curs:integer;
```

```
Grup:string[5];
```

```
End;
```

```
Var student:anketa;
```

К каждому элементу записи можно обратиться используя составное имя, которое имеет следующую структуру: имя переменной.имя поля

Например: student.fio, student.dat и т.п. Если, например полю курс присвоить значение 3, то это можно сделать следующим образом:

```
Student.Curs:=3;
```

Поля записи могут иметь любой тип, в частности сами могут быть записями. Такая возможность используется в том, случае, когда требуется представить многоуровневое дерево (более 2 уровней).

Любая обработка записей, в том числе ввод и вывод, производится поэлементно.

Например, ввод сведений о 500 студентах можно организовать следующим образом:

```
For i:=1 to 500 do
```

```
With Student [i] do
```

Begin

Write ('Ф.И.О. '); Readln (FIO);

Write ('Пол (м/ж) '); Readln (Pol);

Write ('Дата рождения '); Readln (Dat);

Write ('Адрес '); Readln (Adres);

Write ('Курс '); Readln (Curs);

Write ('Группа '); Readln (Grup);

End;

В этом примере использован оператор присоединения, который имеет следующий вид:

With имя переменной типа запись do

Begin

Действия

End;

Он позволяет, один раз указав имя переменной типа запись после слова With, работать в пределах оператора с именами полей как с обычными переменными, т.е. не писать громоздких составных имен.

Примеры задач.



В группе 30 студентов. О каждом студенте известно: фамилия и оценки по 8 предметам. Найти среднюю оценку каждого студента и выдать на экран среднюю оценку и фамилию лучшего студента.

Program zapis;

Const n=30;

Type sved=record;

 fio:string[50];

 p1, p2, p3, p4, p5, p6, p7, p8:integer;

 ocsr:real;

End;

Var s:array[1..n] of sved;

```

i,k: integer;
max: real;
Begin
writeln('введите данные о ',n,' студентах');
For i:=1 to n do
  With spisok [i] do
    Begin
      Writeln ('введите фамилию ученика');
      Readln (famil);
      Writeln ('введите оценки по восьми предметам');
      Readln (p1, p2, p3, p4, p5, p6, p7, p8);
      Clrscr;
    End;
  {нахождение среднего балла }
  for i:=1 to n do
    with s[i] do
      ocsr:=(p1+p2+p3+p4+p5+p6+p7+p8)/8;
    {нахождение максимального среднего балла}
    max:=s[1].ocsr;
    for i:=1 to n do
      if s[i].ocsr>max
      then
        begin
          max:=s[i].ocsr;
          k:=i
        end;
    {печать сведений о лучшем ученике}
    writeln('Лучший ученик')
    writeln ('ФИО - ',s[k].fio,' средний балл - ',s[k].ocsr);
end.

```

Составить программу, которая позволит ввести данные о 10 автомобилях предприятия (марка, цвет, пробег, год выпуска, водитель) и выдаст сведения об автомобилях, старше 10 лет.

```
Program Avto;
Const n=10;
Type mash=record;
    marka:string[25];
    cvet:string[15];
    probeg:real;
    god_vip:integer;
    voditel:string[25];
End;
Var s:array [1.. n] of mash;
    i:integer;
Begin
    writeln ('Введите данные о ',n,' автомобилях');
    for i:=1 to n do
        with s[i] do
            begin
                write('введите марку автомобиля'); readln(marka);
                write('введите цвет автомобиля'); readln(cvet);
                write('введите пробег автомобиля'); readln(probeg);
                write('введите год выпуска'); readln(god_vip);
                write('введите фамилию водителя'); readln(voditel);
            end;
        end;
    writeln('автомобили старше 10 лет');
    for i:=1 to n do
        begin
            with s[i] do
                begin
                    if 2011-god_vip>10
```

```

then
  begin
    writeln ('марка машины - ',marka);
    writeln ('цвет машины - ',cvet);
    writeln ('пробег - ',probeg);
    writeln ('водитель - ',voditel);
  end;
end;
end;
End.

```



Составить программу, которая позволяет ввести данные о 15 деталях, находящихся на складе (название, количество, стоимость одной детали). Вывести название и количество самой дорогой детали.

```

Program detali;
Const n=15;
Type detal=record;
  nazv:string[25];
  kol:integer;
  cena:real;
End;
Var s:array [1.. n] of detal;
  i,k:integer;
  max:real;
Begin
  writeln ('Введите данные о ',n,' деталях');
  for i:=1 to n do
    with s[i] do
      begin
        write('введите название детали'); readln(nazv);
        write('введите количество этой детали на складе'); readln(kol);
        write('введите цену 1-й детали'); readln(cena);

```



```

    end;
max:=s[1].cena;
k:=1;
for i:=1 to n do
    begin
        with s[i] do
            begin
                if cena>max
                then
                    begin
                        max:=cena;
                        k:=i
                    end;
                end;
            end;
        writeln('самая дорогая деталь');
        writeln ('название - ',s[k].nazv);
        writeln ('количество - ',s[k].kol);
    end.

```

End.

Составить программу, которая позволяет ввести данные о 20 кубиках (цвет, материал, длина ребра). Сосчитать количество кубиков с длиной ребра меньше 5 см.

```

Program kubiki;

```

```

Const n=20;

```

```

Type kub=record;

```

```

    cvet:string[25];

```

```

    material:string[25];

```

```

    dlina_rebra:real;

```

```

End;

```

```

Var s:array [1.. n] of kub;

```

```

    i,k:integer;

```

```

Begin

```

```

writeln ('Введите данные о ',n,' кубиках');
for i:=1 to n do
  with s[i] do
    begin
      write('введите цвет кубика'); readln(cvet);
      write('введите материал, из которого изготовлен'); readln(material);
      write('введите длину ребра'); readln(dlina_rebra);
    end;
  k:=0;
  for i:=1 to n do
    with s[i] do
      begin
        if dlina_rebra<5
          then k:=k+1
        end;
      writeln('количество кубиков с длиной ребра меньше 5 см. =' ,k);
    End.

```

Даны две строки, выяснить совпадают ли их длины.

Дана строка, подсчитать количество вхождений слова «кот» в данную строку.

Найти номер позиции, с которой в заданной строке в первый раз встретится некоторое введенное слово.

Подсчитать количество запятых в заданном тексте с помощью массива символов.

Заменить в заданной строке все вхождения «ку» на «за» с помощью операции копирования, удаления и вставки.

Задается последовательность символов(строка, состоящая только из букв латинского алфавита). Определить, являются ли все символы строки уникальными. Уникальность символа означает, что он встречается в строке только один раз. Малые и большие буквы алфавита считаются разными

Задается последовательность символов(строка). Заменить все плюсы на минусы, а все минусы на плюсы.

Задается последовательность символов(строка). Определить количество пробелов

в левой части строки, (от начала строки до первого пробела). Пример: ;.....bnnb',

Дана последовательность символов (строка). Заменить на символ звездочки (*) все буквы английского алфавита, которые повторяются больше, чем один раз. Малые и большие буквы считаются эквивалентными.

Дана последовательность символов (строка). Удалить все знаки "+", если за ними непосредственно следует цифра. Вывести новый текст и количество удаленных символов.

Дан двумерный массив размером $p \times t$, заполненный случайными целыми числами в диапазоне $[-70,30]$ (после заполнения массива случайными числами массив вывести на экран). Определить, есть ли в данном массиве столбец, в котором имеются одинаковые элементы.

Дана последовательность символов (строка). Словом текста является последовательность цифр и букв алфавита (латинского). Будем называть особым такое слово, в котором количество цифр и букв одинаково. Найти количество особых слов

Определить количество слов во введенном тексте, начинающихся с заданной буквы. Считать, что слова в тексте разделены пробелами.

Задания для самостоятельного решения

Вариант 1

1. Известны данные о 5 пассажирах: ФИО, количество вещей и общий вес вещей. Вывести ФИО и количество вещей пассажира с максимальным общим весом вещей.

2. Известны данные о 5 студентах: ФИО, рост и вес. Вывести ФИО и вес всех студентов, которые ниже 180 см.

3. Известны данные о 5 кубиках: цвет, материал и длина ребра. Сосчитать количество красных кубиков.

Вариант 2

1. Известны данные о 5 мячиках: цвет и радиус. Вывести цвет самого маленького мячика.

2. Известны данные о 5 спортсменах: ФИО, возраст и вид спорта. Вывести ФИО и возраст тех спортсменов, которые занимаются плаванием.

3. Известны данные о 5 рабочих: ФИО, стаж, оклад. Сосчитать количество рабочих с окладом больше 15 000 рублей.

Вариант 3

1. Известны данные о 5 сортах конфет: название, цена и фабрика изготовитель. Вывести название и фабрику изготовитель самых дорогих конфет.

2. Известны данные о 5 студентах: ФИО, год рождения и специальность. Вывести ФИО и специальность тех студентов, которые родились после 1994 года.

3. Известны данные о 5 пассажирах: ФИО, количество вещей и общий вес вещей. Сосчитать количество пассажиров с количеством вещей больше 2.

Вариант 4

1. Известны данные о 5 кубиках: цвет, материал и длина ребра. Вывести цвет и материал самого маленького кубика.

2. Известны данные о 5 учениках музыкальной школы: ФИО, класс, инструмент. Вывести ФИО и класс тех из них, кто занимается игрой на скрипке.

3. Известны данные о 5 спортсменах: ФИО, возраст, вид спорта. Сосчитать количество спортсменов младше 20 лет.

Вариант 5

1. Даны две строки, выяснить совпадают ли их длины.

2. Дана строка, подсчитать количество вхождений слова «кот» в данную строку.

3. Найти номер позиции, с которой в заданной строке в первый раз встретится некоторое введенное слово.

Вариант 6

1. Удалить из введенной строки все гласные буквы и вывести результат на экран
 2. Удалить пробелы между первым и вторым вопросительным знаком
 3. Заменить в самом длинном слове строки буквы 'a' на 'b'
-

Вариант 7

1. Введенную строку вывести в обратном порядке (с последней буквы начинается и первой заканчивается)
Посчитать количество согласных букв в введенной строке
 2. Заменить три последних символа у слов, больших определенной длины
 3. Составить программу определяющую, какая из двух фамилий длиннее.
Фамилии имеют разную длину
-

Вариант 8

1. Посчитать количество гласных букв в введенной строке
 2. Даны две строки, выяснить совпадают ли их длины
 3. Определить длины кратчайшего и самого длинного слов в строке
-

Вариант 9

1. Удалить из введенной строки все согласные буквы и вывести результат на экран
 2. Дана строка, подсчитать количество вхождений слова «мама» в данную строку
 3. Поиск самой длинной строки в массиве
-

Вариант 10

1. Найти номер позиции, с которой в заданной строке в первый раз встретится некоторое введенное слово
 2. Заменить пробел и группы пробелов символом
-

3. Составить программу определяющую, какая из двух фамилий длиннее.
Фамилии имеют разную длину
-

Вариант 11

1. Введенную строку вывести в обратном порядке (с последней буквы начинается и первой заканчивается)
 2. Определить длину самого короткого слова в строке
 3. Подсчет количества слов в строке
-

Вариант 12

1. Посчитать количество гласных букв в введенной строке
 2. Найти слово под заданным номером
 3. Удаление из строки слов заданной длины
-

Контрольные вопросы

1. Что такое строковый тип?
2. С какими типами данных может работать строковый тип?
3. Что такое процедура Val?

Практическая работа № 15, 16

Тема: Процедуры и функции

Цель: научиться применять подпрограммы в решении задач на Паскале, а также научиться понимать какой вид подпрограммы необходим при решении определенной задачи и рассмотреть основные приемы использования подпрограмм

Ход работы:

Теоретическое обоснование:

Иногда в разных местах программы приходится выполнять практически одни и те же последовательности действий с разными исходными данными. Такие последовательности действий можно оформить в виде так называемых подпрограмм (от англ. subroutine) – сгруппировать операторы в блок, к которому можно обратиться по имени, причем неоднократно.

Подпрограммы сокращают текст программы, существенно уменьшают время их исполнения, облегчают жизнь программистам, которые могут создавать программы модульно, т. е. собирая сложную программу из законченных кусочков более простых составляющих. Это позволяет группе программистов создавать большие программы, а группе школьников разрабатывать и реализовывать какие-либо глобальные проекты

Подпрограммы делятся на процедуры и функции.

Встроенные (стандартные) процедуры и функции являются частью языка и могут вызываться по имени без предварительного описания. Например, `abs`, `sqrt`, `ln`, `sin...`- функции (возвращают результат), `readln`, `write...` – процедуры (не возвращают результат). Их наличие существенно облегчает разработку прикладных программ. Однако в большинстве случаев некоторые специфичные для данной программы действия не находят прямых аналогов в библиотеках Turbo Pascal, и тогда программисту приходится разрабатывать свои нестандартные процедуры и функции.

Процедуры пользователя пишутся самим программистом в соответствии с синтаксисом языка в разделе описания подпрограмм.

Структура процедуры повторяет структуру программы, это "программа в миниатюре" — она также представлена заголовком и телом.

Заголовок состоит из зарезервированного слова `procedure`, идентификатора (имени) процедуры.

```
VAR ... // раздел описания переменных главной программы
procedure Имя Процедуры;
var ...
begin
...
end;
begin
//тело главной программы
end.
```

Вызов процедуры для последующего выполнения записывается в теле главной программы.



Пример 1. Программа вычисления площади и периметра.

Открыть в Паскале файл. Оформим повторяющуюся часть программы в виде процедуры (программа внутри главной программы).

```
{***Вычисление***}
procedure tx;
begin
writeln;
writeln('Вычисление: ');
write('Введите стороны a, b: ');
end;
{*****}
```

Вызов в нужном месте:

```
tx;
```

При вызове процедуры работа главной программы приостанавливается и начинает выполняться вызванная процедура. Когда процедура выполнит свою задачу, программа продолжится с оператора, следующего за оператором вызова процедуры.

Т.е. мы “научили” ПК новой команде tx. Ею можно пользоваться только в этой программе и, причем, много раз.

Достоинства подпрограмм:

Программы, написанные с участием подпрограмм, легче тестировать и отлаживать, у них более четкая логическая структура.

Самостоятельный характер подпрограмм позволяет поручать их составление различным программистам. Так осуществляется разделение работы по программированию и, тем самым, ускоряется ее завершение;

Использование подпрограмм позволяет экономить память. Память для хранения переменных, используемых в подпрограмме, выделяется только на время ее работы и высвобождается, как только ее выполнение заканчивается.



Пример 2. Пользователь вводит две стороны трех прямоугольников.

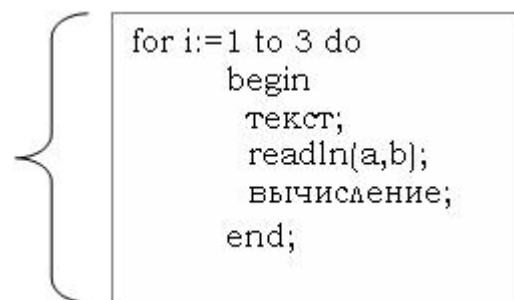
Вывести их площади.

Можно решить задачу так:

```
for i:=1 to 3 do
begin
writeln('Введите a и b:');
readln(a,b);
writeln('Площадь=',a*b);
end;
```

Хорошим стилем программирования считается использование процедур. Необходима процедура, которая будет вычислять площадь прямоугольника. Вот как схематично будет выглядеть главная программа:

```
текст
readln (a,b);
вычисление
текст
readln (a,b);
вычисление
текст
readln (a,b);
```



вычисление

Процедура текста уже есть (см пример1). Создадим вторую процедуру, которая вычисляет площадь. Но для того чтобы вычислить S, надо знать 2 стороны, поэтому процедуре надо показать какие стороны она должна перемножать.

другие переменные (ненастоящие),
формальные
Определяются новые ячейки памяти

```
procedure pl (c,d: integer);  
var S:integer;  
begin  
S:=c*d;  
writeln('площадь прямоугольника со сторонами ',c, ' ',d, '= ',S);  
end;
```

Параметр – это переменная, которой присваивается некоторое значение. Существуют формальные параметры, определенные в заголовке подпрограммы, и фактические параметры – выражения, задающие конкретные значения при обращении к подпрограмме.

Процедура выполнится, если вызвать ее по имени и указать фактические параметры,

отделенные
друг от
друга

настоящие, реальные, фактические. Должны быть разные, т.к
позволяют выполнять процедуру с различными значениями.

запятymi и заключенных в круглые скобки:

```
pl(4,5);  
pl(a,b);
```

Фактические параметры должны совпадать по типу и количеству с формальными.

Итак, главная программа:

```
for i:=1 to 3 do  
begin  
tx;  
readln(a,b);
```

```
pl(a,b);
```

```
end;
```

Замечание. При решении этой задачи необходимо проверять введенные пользователем числа (они не должны быть отрицательными, иначе работа программы прервется).

Составим процедуру проверки:

```
procedure error (f,g:integer);
```

```
begin
```

```
if (f<0) or (g<0) then begin
```

```
writeln('стороны прямоугольника не могут быть отрицательными');
```

```
halt; //прерывание программы
```

```
end;
```

```
end;
```

Итак формат процедуры:

```
Procedure <имя> (формальные параметры);
```

```
Const ...;
```

```
Type ...;
```

```
Var ...;
```

```
Begin
```

```
<операторы>;
```

```
end;
```



Пример 3. Составить программу обмена местами двух чисел $c=5$ и $d=7$.

```
program obmenDan;
```

```
uses wincrt;
```

```
var c,d:integer;
```

```
{ обмен }
```

```
procedure obmen ( a,b:integer);
```

```
var m:integer;
```

```
begin
```

```
m:=a; a:=b; b:=m;
```

```
writeln(a,b);
```

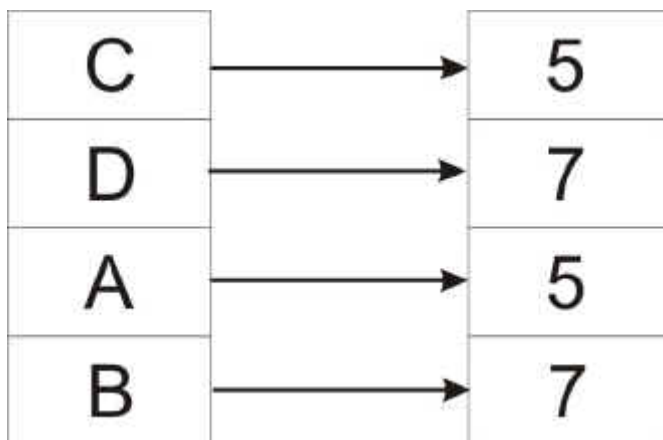
```

end;
begin
writeln ('Введите 2 числа: ');
readln(c,d);
obmen(c,d);
writeln(c, ', ',d);
end.

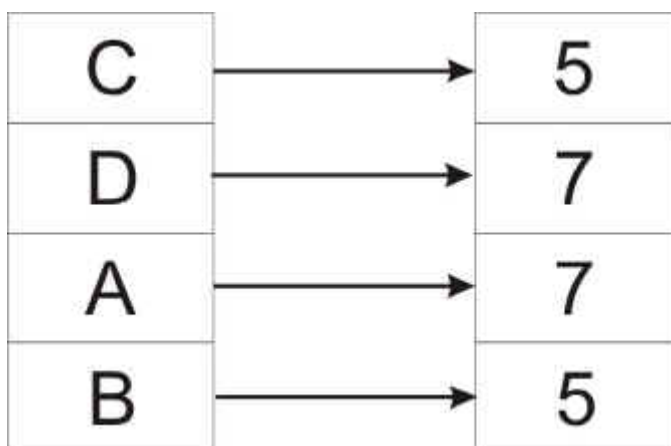
```

После запуска программы видно, что поменялись местами формальные параметры (в процедуре), а фактические (которые используются в лавной программе) – не поменялись. Рассмотрим рисунок, на котором приведена часть оперативной памяти:

1) при вызове процедуры обмен с двумя параметрами 5 и 7, в переменные а и в помещаются тоже числа 5 и 7 соответственно:



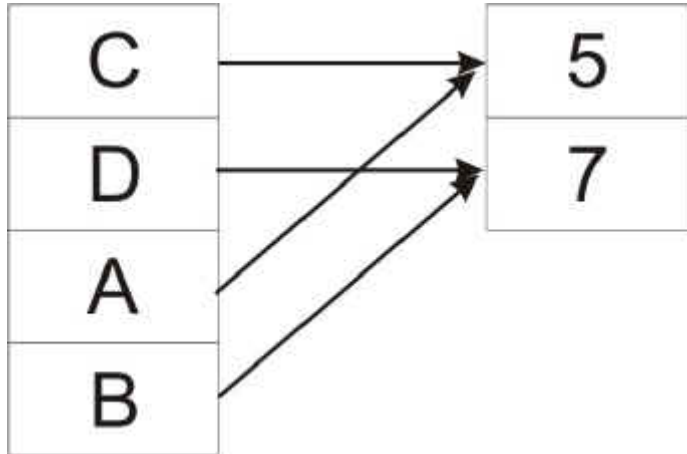
2) далее в процедуре осуществляется перестановка значений ячеек памяти а и в:



3) но в переменных с и d данные не поменялись, т.к. они находятся в других ячейках памяти.

Для того чтобы переменные *c* и *d*, *a* и *b* ссылались на одни и те же ячейки памяти (если изменятся значения *a* и *b*, то изменятся значения и *c*, *d*) необходимо при описании формальных параметров, перед нужными переменными добавить слово VAR:

```
procedure obmen (var a,b:integer);
```



Измените программу `obmen` на:

```
obmen (1,4);
```

⇒ ошибка из-за `var`. Числа – константы, которые нельзя изменять в процедуре.



Пример 4. Найти площадь круга с использованием процедуры, которая производит только вычисление, но не отображает результат на экране.

```
procedure circle (r:real);
```

```
var S:real;
```

```
begin
```

```
S:=pi*r*r;
```

```
end;
```

Процедура должна возвращать результат:

```
procedure circle (r:real; var S:real);
```

```
begin
```

```
S:=pi*r*r;
```

```
end;
```

```
begin
```

```
readln(a, e);
```

e и *S* - ссылаются на 1 ячейку памяти!

```
writeln(e);
```

```
end.
```

Замечание: Переменная в процедуре S используется для возвращения результатов работы процедуры в основную программу. При ее изменении, изменяется и фактический параметр в вызывающей программе, т.е. переменная e.

Чаще для этого в Паскале вместо процедур используют функции (подпрограммы, которые что-то возвращают).

Функция аналогична процедуре, но имеются два отличия.

Функция передает в программу результат своей работы – единственное значение, носителем которого является имя своей функции.

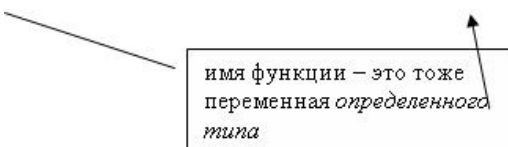
Имя функции может входить в выражение как операнд. Функция возвращает результат в точку своего вызова.

Например, $\text{sqr}(x)$ – возведет в квадрат значение x и возвратит в точку вызова вычисленное значение квадрата числа x : $y:=\text{sqr}(x)$;

Функция, определенная пользователем, состоит из заголовка и тела функции. Тело функции по структуре аналогично программе. Описание меток, констант, типов и т.д. действительны только в пределах данной процедуры.

Формат:

```
Function <имя> (формальные параметры) : <тип результата>;
```



имя функции – это тоже
переменная *определенного*
типа

```
Const ...;
```

```
Type ...;
```

```
Var ...;
```

```
Begin
```

```
<операторы>;
```

```
end;
```

В разделе операторов должен находиться хотя бы один оператор, присваивающий имени функции значение. В точку вызова возвращается результат последнего присваивания.



Пример 5. Переделаем задачу о площади круга.

```
function circle (r:real): real;
begin
circle:=pi*r*r;
end;
ВЫЗОВ:
a:=circle(5); (ОБЯЗАТЕЛЬНО присваиваем)
или
e:=circle(a);
writeln(e);
```

Пример 6. Найти $1!+2!+\dots+n!$

Используем функцию нахождения факториала, т.к подаем на вход и получаем результат.

```
function fact (a:integer): integer;
var i: integer;
begin
fact:=1;
for i:=1 to a do
fact:=fact*I;
end;
```

В строке `fact:=fact*I;`

компилятор найдет ошибку, т.к `fact` должна вызываться с параметрами.

Поэтому обячно вводят дополнительную переменную, в которую помещают результат. А потом в переменную `fact` присваивают этот результат:

```
program factorial;
uses wincrt;
var sum,n,j: integer;
function fact (a: integer): integer;
var i,d: integer;
begin
d:=1;
```

```
for i:=1 to a do
d:=d*i;
fact:=d;
end;
begin
sum:=0;
readln(n);
for j:=1 to n do
sum:=sum+fact(j);
writeln(sum);
end.
```

Задания для самостоятельного решения

Вариант № 1

1. Написать процедуру, которая складывает два любых числа (два параметра)
 2. С помощью подпрограммы заполнить матрицы случайными числами. Написать подпрограмму, вычисляющую сумму двух матриц. Вывести на экран две исходные матрицы и их сумму (используя процедуру)
-

Вариант № 2

1. Необходимо определить наибольший общий делитель двух введенных чисел, используя процедуру (два параметра-значения, один параметр-переменная)
 2. Найти разность двух матриц по модулю, т.е. из большего элемента одной матрицы вычесть меньший элемент второй матрицы, находящийся в той же позиции
-

Вариант № 3

1. Даны 3 различных массива целых чисел (размер каждого 15 элементов). В каждом массиве найти сумму элементов
 2. Преобразовать строку так, чтобы буквы каждого слова в ней были отсортированы по алфавиту
-

Вариант № 4

1. Создать процедуру для вывода десяти элементов массива (два параметра: количество элементов, массив)
2. Написать функцию, которая циклически сдвигает одномерный массив вправо или влево на указанное число позиций. Сдвиг также должен быть кольцевым, то есть те элементы, которые уходят вправо или влево за пределы массива, должны помещаться с другого его конца.

Например, дан массив:

1 2 3 4 5 6

Кольцевой сдвиг вправо на 2 единицы:

5 6 1 2 3 4

Вариант № 5

1. Создать процедуру для заполнения массива с использованием функции random
2. Получить десять массивов случайных чисел. Найти среди них тот, сумма элементов которого наибольшая.

Вариант № 6

1. Даны 3 различных массива целых чисел (размер каждого 15 элементов). В каждом массиве найти среднее арифметическое значение

Вариант № 7

1. Составить программу с процедурой для вычисления степени числа (входные параметры: число и степень). Для вывода результата использовать параметр-переменную
2. Дана квадратная матрица. Вычесть последнюю строку из каждой строки (реализовать с помощью подпрограмм).

Вариант № 8

1. Составить программу на Паскале с процедурой для вычисления факториала числа. ($0!=1$, $1!=1$, $2!=2$, $3!=6$...)
 2. Описать процедуру нахождения по заданной длине стороны треугольника и величинам двух прилежащих углов длин остальных сторон треугольника и величины третьего угла.
-

Вариант № 9

1. для 5 одномерных массивов определять произведение элементов каждого массива, используя процедуру с двумя параметрами — число элементов массива и параметр-переменная для вывода произведения
 2. В двумерном массиве случайных чисел $[1..10,1..10]$ вычислить сумму элементов побочной диагонали.
-

Вариант № 10

1. для 5 одномерных массивов определять минимальный элемент каждого массива, используя процедуру с двумя параметрами — число элементов массива и параметр-переменная для вывода минимального элемента
 2. Найти средние арифметические пяти массивов, состоящих их десяти целых чисел.
-

Вариант № 11

1. Необходимо определить наибольший общий делитель двух введенных чисел, используя процедуру (два параметра-значения, один параметр-переменная)
 2. Написать программу, которая переводит число из десятичной системы счисления в двоичную или восьмеричную.
-

Вариант № 12

1. Даны 3 различных массива целых чисел (размер каждого 15 элементов). В каждом массиве найти сумму элементов
 2. Двоичное число, введенное пользователем программы, преобразовать в десятичное число. Результат вывести на экран.
-

Контрольные вопросы:

1. Что входит в тело процедуры?
2. Отличия процедуры и функции.
3. Понятия локальной и глобальной переменной.

Практическая работа №17

Тема: Работа с файлами

Цель: Изучить способы работы с файлами в Паскале

Ход работы:

Теоретическое обоснование

Файл - это упорядоченная последовательность однотипных компонентов, расположенных на внешнем носителе. Файлы предназначены только для хранения информации, а обработка этой информации осуществляется программами. Использование файлов целесообразно в случае:

долговременного хранения данных ;

доступа различных программ к одним и тем же данным;

обработки больших массивов данных, которые невозможно целиком разместить в оперативной памяти компьютера.

В Паскале определены текстовые файлы, типизированные и не типизированные. Файл, не содержащий ни одного элемента, называется пустым. Создается файл путем добавления новых записей в конец первоначально пустого файла. Длина файла, т.е. количество элементов, не задается при определении файла.

Все файлы должны быть описаны в программе либо в разделе переменных VAR, либо в разделе типов TYPE. Под чтением файла понимают ввод данных из внешнего файла, находящегося на диске, в оперативную память машины. Запись в файл - вывод результатов работы программы из оперативной памяти на диск в файл.

Работа с файлами выполняется следующими процедурами:

Assign – устанавливает связь между именем файла в программе (файловой переменной) и физическим именем файла, принятым в ОС.

Reset - открывает существующий файл для чтения.

Rewrite – создает и открывает новый файл для записи на внешнем устройстве (если файл ранее существовал, вся предыдущая информация из него стирается).

Close - закрывает открытый файл.

Для определения конца файла используется стандартная встроенная функция EOF (файловая переменная), которая принимает значение True, если достигнут конец файла, и значение False в противном случае.

Текстовые файлы

Текстовые файлы – файлы на диске, состоящие из символов ASCII. Для разделения строк используются символы «конец строки». Текстовые файлы являются файлами с последовательным доступом. В любой момент времени доступна только одна запись файла. Другие записи становятся доступными лишь в результате последовательного продвижения по файлу. Текстовые файлы внутренне разделены на строки, длины которых различны. Для разделения строк используется специальный маркер конца строки. Объявляются текстовые файлы переменной типа `text`. Обработать их можно только последовательно и с помощью процедур и функций:

`Readln (f , st)` - чтение строки `st` из файла `f` и переход на начало следующей ;

`Writeln (f , st)` - запись строки `st` в файл `f` и маркера конца строки ;

`Append (f)` - процедура, открывающая файл `f` для добавления строк в конец файла;

`Eoln (st)` - логическая функция, результат выполнения которой равен `TRUE`, если достигнут маркер конца строки `st`.



Пример 1. Создать текстовый файл, в который записать 3 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.

```
Program File_text;
var
f1 : text;
st : string;
n: byte;
begin
assign (f1, 'file1.txt'); {связать с файлом file1.txt файловую переменную f1 }
rewrite (f1); { создать новый файл с именем file1.txt }
writeln ( f1, 'Очень полезно изучать'); { записать предложения в файл
writeln ( f1, ' всем студентам ');
writeln (f1, ' язык Pascal ');
```

```

close (f1); { закрыть файл для записи }
reset (f1); { открыть файл для чтения }
while not eof (f1) do { пока не конец файла f1 }
begin
readln (f1, st); { читаем строку из файла f1 }
writeln(st); { выводим на экран }
n:= length (st); { определяем длину строки }
writeln (' длина =' ,n);
end;
close (f1); { закрыть файл для чтения }
end .

```

Типизированные файлы

Типизированные файлы – это файлы, состоящие из нумерованной последовательности объектов (записей) любого типа. С такими файлами можно работать в режиме прямого доступа, при котором выполняется непосредственное обращение к любой записи файла. Каждая запись файла имеет свой номер, начиная с 0 и т.д.

Процедуры и функции обработки файлов:

1) Write и Read- записывают и читают информацию из указанного файла и перемещают указатель файла к следующей записи.

2) Seek (файловая переменная, номер записи); процедура перемещения указателя на запись файла с заданным номером.

3) Truncate (файловая переменная); процедура, усекающая файл по текущей позиции указателя файла, т.е. все записи, находящиеся после указателя файла, удаляются.

4) Функция Filesize (файловая переменная); имеет тип Integer и определяет размер файла, т.е. число записей.

5) Функция Filepos (файловая переменная); имеет тип Integer и возвращает текущую позицию указателя файла.

Для добавления записей в конец файла используются процедуры:

```
Readln (a );
```

Seek (f, filesize (f));

Write (f, a);

При этом указатель устанавливается за конец файла, т.к. нумерация записей начинается с нуля. После чего с помощью Write можно добавлять записи. Открывать файл можно только процедурой Reset (f). Для того, чтобы в режиме произвольного доступа считать, а затем изменить значение записи, следует выполнить два вызова процедуры Seek. Один вызов перед операцией Read, а другой - перед операцией Write (т.к. Read после чтения записи переместит указатель к следующей записи).



Пример: Создать файл из списка 10 студентов с их оценками (номер, Ф.И.О. и три оценки). Вывести его содержимое на экран, изменить фамилию студента с номером, введенным с клавиатуры, заново прочитать файл.

Program file;

Type

wed = record {Тип wed включает 3 поля: n, fio, bal}

n : byte ; fio : string[15] ;

bal : array [1..3] of byte; {Поле bal – массив из 3 оценок }

end;

Var spisok : wed ; {Запись spisok типа wed}

sp : file of wed; {Файл записей типа wed}

procedure vvod; { процедура создания файла}

var i,j:byte;

begin

{ оператор assing находится в основной прграмме }

rewrite (sp); {открытие файла для записи}

with spisok do

For i:=1 to 10 do begin

n:=i;

writeln (' Введите фамилию - ', i); readln (fio);

writeln (' Введите 3 оценки ', fio); For j:= 1 to 3 do readln (bal [j]);

write (sp , spisok); { запись в файл информации о студенте}

```

end;
close (sp);    { закрытие файла для записи }
end;
procedure print; { процедура чтения и печати всего файла }
var  j : byte
begin
reset ( sp);      { открытие файла для чтения }
writeln ( '      Список студентов: ');
while not eof (sp) do
with spisok do
begin
Read (sp, spisok); { чтение данных из файла }
write (n, ' ', fio); { вывод записи на экран }
For j:= 1 to 3 do write ( ' ', bal [j] );
writeln ;
end;
readln;
close (sp) ;
end;
procedure  work;
var  num: integer;
begin
reset ( sp);      { открытие файла для чтения }
writeln ('номер= ');  readln (num);
seek (sp, num-1); { поиск записи с указанным номером (нумерация записей с
0)}
read (sp,spisok); { чтение и перемещение указателя к сле д. записи }
write ('fio='); writeln (spisok.fio);
seek (sp,filepos(sp)-1); {возвращение к изменяемой записи }
writeln ( ' Введите новую фамилию' ); readln (spisok.fio);
write (sp, spisok); {запись в файл измененной записи }

```

```

close (sp);
end;

begin   {начало основной программы}
assign (sp,'Vedom.DAT'); {связать файловую переменную sp с файлом
Vedom.dat}vvod;
    print;
    {процедуры создания и чтения файла}
work;
    print;
    {корректировка и чтение измененного файла}
readln
end.

```

Задания для самостоятельного решения

Вариант № 1

1. Создать текстовый файл, в который записать 3 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.
2. Создать файл из списка 10 машин с их данными (номер, Ф.И.О. владельца, марка). Вывести его содержимое на экран, изменить фамилию владельца с номером, введенным с клавиатуры, заново прочитать файл.
3. й

Вариант № 2

1. Создать текстовый файл, в который записать 4 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.
2. Создать файл из списка 10 спортсменов с их данными (номер, Ф.И.О. и вид спорта, и их достижения). Вывести его содержимое на экран, изменить фамилию спортсмена с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 3

1. Создать текстовый файл, в который записать 5 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.
2. Создать файл из списка 10 стран с их данными (номер п/п, Ф.И.О. президента и столица, площадь). Вывести его содержимое на экран, изменить фамилию президента с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 4

1. Создать текстовый файл, в который записать 6 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.
2. Создать файл из списка 10 фильмов с их данными (номер п/п, Ф.И.О. режиссёра, главного героя, год создания, жанр). Вывести его содержимое на экран, изменить фамилию главного героя с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 5

1. Создать текстовый файл, в который записать 7 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.
2. а

Вариант № 6

1. Создать текстовый файл, в который записать 3 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.а
2. Создать файл из списка 15 фильмов с их данными (номер п/п, Ф.И.О. режиссёра, главного героя, год создания, жанр). Вывести его содержимое на экран, изменить фамилию главного героя с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 7

1. Создать текстовый файл, в который записать 4 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.а
2. Создать файл из списка 10 стран с их данными (номер п/п, Ф.И.О. президента и столица, площадь). Вывести его содержимое на экран, изменить фамилию президента с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 8

1. Создать текстовый файл, в который записать 5 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.а
2. Создать файл из списка 10 стран с их данными (номер п/п, Ф.И.О. президента и столица, площадь). Вывести его содержимое на экран, изменить фамилию президента с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 9

1. Создать текстовый файл, в который записать 6 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.а
2. Создать файл из списка 15 спортсменов с их данными (номер, Ф.И.О. и вид спорта, и их достижения). Вывести его содержимое на экран, изменить фамилию спортсмена с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 10

1. Создать текстовый файл, в который записать 7 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.
2. Создать файл из списка 17 спортсменов с их данными (номер, Ф.И.О. и вид спорта, и их достижения). Вывести его содержимое на экран, изменить фамилию спортсмена с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 11

1. Создать текстовый файл, в который записать 3 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения.
2. Создать файл из списка 15 стран с их данными (номер п/п, Ф.И.О. президента и столица, площадь). Вывести его содержимое на экран, изменить фамилию президента с номером, введенным с клавиатуры, заново прочитать файл.

Вариант № 12

1. Создать текстовый файл, в который записать 4 предложения. Прочитать этот файл, вывести его содержимое на экран. Определить длину каждого предложения. а
 2. Создать файл из списка 10 фильмов с их данными (номер п/п, Ф.И.О. режиссёра, главного героя, год создания, жанр). Вывести его содержимое на экран, изменить фамилию главного героя с номером, введенным с клавиатуры, заново прочитать файл.
-

Контрольные вопросы

1. Что такое файл?
2. Что такое файловый тип переменных?
3. Что такое типизированный файл?

Практическая работа № 18, 19, 20

Тема: Работа с графикой

Цель: Овладение начальными навыками работы с графическим режимом экрана, используя модуль GRAPH

Ход работы:

Теоретическое обоснование

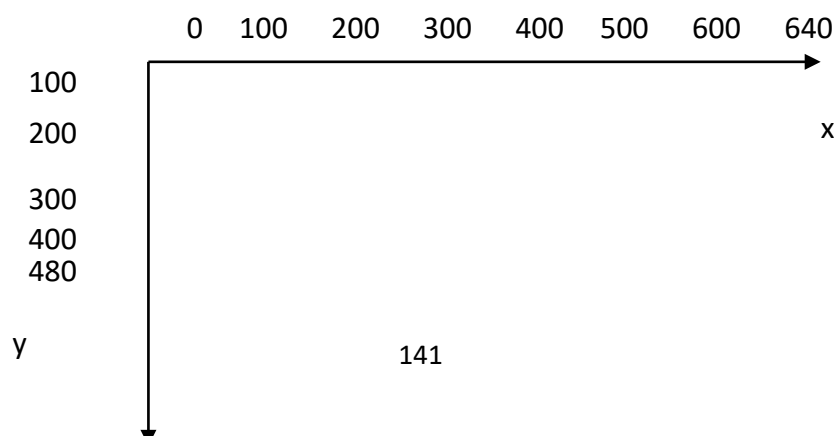
Компьютерная графика в Паскале - область информатики, изучающая методы и свойства обработки изображений с помощью программно-аппаратных средств.

Представление данных на компьютере в графическом виде впервые было реализовано в середине 50-х годов. Сначала, графика применялась в научно-военных целях. Сейчас эта область детально изучается в высшей технической школе.

Построение графических изображений в Паскале, исследование графиков функций, является одной из важных и интересных тем программирования.

Обычно экран компьютера находится в символьном режиме, а для переключения экрана в графический режим необходимо подключение модуля **graphABC**. В результате этого, экран очищается и переходит в цветной графический режим.

В графическом режиме экран представляет собой совокупность точек, каждая из которых может быть окрашена в один из 16 цветов. Каждая точка экрана имеет свои координаты (x,y) . Начало координат находится в верхнем левом углу экрана. Координаты точек возрастают слева направо и сверху вниз. Левая верхняя точка имеет координаты $(0,0)$, а правая нижняя имеет координаты $(640,480)$. Для того, чтобы программа могла выводить на экран графические примитивы (линии, окружности, прямоугольники и т.п.), необходимо инициализировать графический режим.



После запуска PascalABC, по умолчанию, запускается текстовый режим. Для работы с графикой служит отдельное графическое окно. Чтобы его открыть, необходимо подключить *модуль GraphABC*. В этом модуле содержится обширный набор процедур и функций, предназначенных для работы с графическим экраном, а также некоторые встроенные константы и переменные, которые могут быть использованы в программах с графикой. С их помощью можно создавать разнообразные графические изображения и сопровождать их текстовыми надписями.

Шаблон графической программы

```
Program<имя программы>;  
UsesgraphABC;  
Var<список переменных>:<тип переменных>;  
Begin  
  <здесь находится непосредственно  
  блок решения задачи>  
End.
```

UsesgraphABC - подключение модуля библиотеки графических процедур

2. Модуль GraphABC

Модуль GraphABC содержит процедуры и функции для рисования графическом окне. Они подразделяются на следующие группы:

- Графические примитивы
- Действия с цветом
- Действия с пером
- Действия с кистью
- Действия со шрифтом
- Действия с графическим окном

Графические примитивы

Формат записи	Как работает
Setpixel($x,y,color$);	Закрашивает один пиксель с координатами (x,y) цветом $color$.
Line($x1,y1,x2,y2$);	Вычерчивает линию от точки с координатами $(x1,y1)$ до точки $(x2,y2)$.
Rectangle($x1,y1,x2,y2$);	Рисует контур прямоугольника со сторонами, параллельными сторонам экрана. Точки с координатами $(x1,y1)$ и $(x2,y2)$ определяют диагональ прямоугольника.
RoundRect($x1,y1,x2,y2,w,h$);	Рисует прямоугольник со скругленными краями. Точки с координатами $(x1,y1)$ и $(x2,y2)$ определяют диагональ прямоугольника., а w и h — ширину и высоту эллипса, используемого для скругления краев.
Circle(x,y,R);	Рисует окружность радиуса R с центром в точке с координатами (x,y) .
Ellipse($x1,y1,x2,y2$);	Рисует эллипс, заданный своим описанным прямоугольником с координатами противоположных вершин $(x1,y1)$ и $(x2,y2)$.
Arc($x,y,r,a1,a2$);	Рисует дугу окружности с центром в точке (x,y) и радиусом r , заключенной между двумя лучами, образующими углы $a1$ и $a2$ с осью OX ($a1$ и $a2$ – вещественные, задаются в градусах и отсчитываются против часовой стрелки).
Pie($x,y,r,a1,a2$);	Рисует сектор окружности, ограниченный дугой (параметры процедуры имеют тот же смысл, что и в процедуре Arc).
Chord($x,y,r,a1,a2$);	Рисует фигуру, ограниченную дугой окружности и отрезком, соединяющим ее концы (параметры процедуры имеют тот же смысл, что и в процедуре Arc).
MoveTo(x,y);	Передвигает невидимое перо к точке с координатами

	(x,y) ; эта функция работает в паре с функцией <i>LineTo(x,y)</i> .
<i>LineTo(x,y)</i> ;	Рисует отрезок от текущего положения пера до точки (x,y) ; координаты пера при этом также становятся равными (x,y) .
<i>TextOut(x,y,s)</i> ;	Выводит строку <i>s</i> в позицию (x,y) (точка (x,y) задает верхний левый угол прямоугольника, который будет содержать текст из строки <i>s</i>).
<i>FloodFill(x,y,c)</i> ;	Заливает область одного цвета цветом <i>c</i> , начиная с точки (x,y) .
<i>FillRect(x1,y1,x2,y2)</i> ;	Заливает прямоугольник, заданный координатами противоположных вершин $(x1,y1)$ и $(x2,y2)$, цветом текущей кисти.

Действия с цветом

<i>clBlack</i> – черный	<i>clAqua</i> – бирюзовый
<i>clPurple</i> – фиолетовый	<i>clOlive</i> – оливковый
<i>clWhite</i> – белый	<i>clFuchsia</i> – сиреневый
<i>clMaroon</i> – темно-красный	<i>clTeal</i> – сине-зеленый
<i>clRed</i> – красный	<i>clGray</i> – темно-серый
<i>clNavy</i> – темно-синий	<i>clLime</i> – ярко-зеленый
<i>clGreen</i> – зеленый	<i>clMoneyGreen</i> – цвет зеленых денег
<i>clBrown</i> – коричневый	<i>clLtGray</i> – светло-серый
<i>clMedGray</i> – серый	<i>clDkGray</i> – темно-серый
<i>clSilver</i> – серебряный	<i>clBlue</i> – синий
<i>clCream</i> – кремовый	<i>clSkyBlue</i> – голубой
<i>clYellow</i> – желтый	

Действия с пером(стиль и цвет рисования)

Формат записи	Как работает																					
SetPenColor(color);	Устанавливает цвет пера, задаваемый параметром <i>color</i> .																					
SetPenWidth(w);	Устанавливает ширину пера, равную <i>w</i> пикселям.																					
SetPenStyle(ps);	Устанавливает стиль пера(<i>ps</i>), который задается следующими именованными константами:																					
	<table border="1"> <thead> <tr> <th>Значение</th> <th>Описание</th> <th>Вид линии</th> </tr> </thead> <tbody> <tr> <td>psSolid</td> <td>Сплошная линия</td> <td>_____</td> </tr> <tr> <td>psClear</td> <td>Отсутствие линии</td> <td></td> </tr> <tr> <td>psDash</td> <td>Штриховая линия</td> <td>- - - - -</td> </tr> <tr> <td>psDot</td> <td>Пунктирная линия</td> <td>.....</td> </tr> <tr> <td>psDashDot</td> <td>Штрихпунктирная линия</td> <td>-.-.-.-.-.-</td> </tr> <tr> <td>psDashDotDot</td> <td>Линия – штрих и два пунктира</td> <td>-.-.-.-.-.-</td> </tr> </tbody> </table>	Значение	Описание	Вид линии	psSolid	Сплошная линия	_____	psClear	Отсутствие линии		psDash	Штриховая линия	- - - - -	psDot	Пунктирная линия	psDashDot	Штрихпунктирная линия	-.-.-.-.-.-	psDashDotDot	Линия – штрих и два пунктира	-.-.-.-.-.-
Значение	Описание	Вид линии																				
psSolid	Сплошная линия	_____																				
psClear	Отсутствие линии																					
psDash	Штриховая линия	- - - - -																				
psDot	Пунктирная линия																				
psDashDot	Штрихпунктирная линия	-.-.-.-.-.-																				
psDashDotDot	Линия – штрих и два пунктира	-.-.-.-.-.-																				

Действия с кистью (стиль и цвет заливки фигур)

Формат записи	Как работает																		
SetBrushColor(color);	Устанавливает цвет кисти, задаваемый параметром <i>color</i> (смотри таблицу цветов).																		
SetBrushStyle(bs);	Устанавливает стиль кисти(<i>bs</i>), который задается следующими именованными константами:																		
	<table border="1"> <thead> <tr> <th>Константа</th> <th>Стиль заливки</th> </tr> </thead> <tbody> <tr> <td>bsSolid</td> <td>Сплошная заливка текущим цветом.</td> </tr> <tr> <td>bsClear</td> <td>Без заливки</td> </tr> <tr> <td>bsCross</td> <td>Штриховка в клетку.</td> </tr> <tr> <td>bsDiagCross</td> <td>xxx штриховка</td> </tr> <tr> <td>bsHorizontal</td> <td>Штриховка горизонтальными линиями</td> </tr> <tr> <td>bsVertical</td> <td>Штриховка вертикальными линиями</td> </tr> <tr> <td>bsBDiagonal</td> <td>/// штриховка</td> </tr> <tr> <td>bsFDiagonal</td> <td>\\\ штриховка</td> </tr> </tbody> </table>	Константа	Стиль заливки	bsSolid	Сплошная заливка текущим цветом.	bsClear	Без заливки	bsCross	Штриховка в клетку.	bsDiagCross	xxx штриховка	bsHorizontal	Штриховка горизонтальными линиями	bsVertical	Штриховка вертикальными линиями	bsBDiagonal	/// штриховка	bsFDiagonal	\\\ штриховка
Константа	Стиль заливки																		
bsSolid	Сплошная заливка текущим цветом.																		
bsClear	Без заливки																		
bsCross	Штриховка в клетку.																		
bsDiagCross	xxx штриховка																		
bsHorizontal	Штриховка горизонтальными линиями																		
bsVertical	Штриховка вертикальными линиями																		
bsBDiagonal	/// штриховка																		
bsFDiagonal	\\\ штриховка																		

Действия со шрифтом

Формат записи	Как работает																		
SetFontColor(color);	Устанавливает цвет шрифта (смотри таблицу цветов).																		
SetFontSize(sz);	Устанавливает размер шрифта в пунктах.																		
SetFontName(s);	Устанавливает наименование шрифта. По умолчанию установлен шрифт, имеющий наименование <i>MS SansSerif</i> . Наиболее распространенные шрифты – это <i>TimesNewRoman</i> , <i>Arial</i> и <i>CourierNew</i> . Наименование шрифта можно набирать без учета регистра.																		
SetFontStyle(fs);	Устанавливает стиль шрифта(<i>fs</i>), который задается следующими именованными константами:																		
	<table border="1"> <thead> <tr> <th>Константа</th> <th>Стиль шрифта</th> </tr> </thead> <tbody> <tr> <td>fsNormal</td> <td>обычный</td> </tr> <tr> <td>fsBold</td> <td>жирный</td> </tr> <tr> <td>fsItalic</td> <td>наклонный</td> </tr> <tr> <td>fsBoldItalic</td> <td>жирный наклонный</td> </tr> <tr> <td>fsUnderline</td> <td>подчеркнутый</td> </tr> <tr> <td>fsBoldUnderline</td> <td>жирный подчеркнутый</td> </tr> <tr> <td>fsItalicUnderline</td> <td>наклонный подчеркнутый</td> </tr> <tr> <td>fsBoldItalicUnderline</td> <td>жирный наклонный подчеркнутый</td> </tr> </tbody> </table>	Константа	Стиль шрифта	fsNormal	обычный	fsBold	жирный	fsItalic	наклонный	fsBoldItalic	жирный наклонный	fsUnderline	подчеркнутый	fsBoldUnderline	жирный подчеркнутый	fsItalicUnderline	наклонный подчеркнутый	fsBoldItalicUnderline	жирный наклонный подчеркнутый
Константа	Стиль шрифта																		
fsNormal	обычный																		
fsBold	жирный																		
fsItalic	наклонный																		
fsBoldItalic	жирный наклонный																		
fsUnderline	подчеркнутый																		
fsBoldUnderline	жирный подчеркнутый																		
fsItalicUnderline	наклонный подчеркнутый																		
fsBoldItalicUnderline	жирный наклонный подчеркнутый																		

Действия с графическим окном

Формат записи	Как работает
ClearWindow;	Очищает графическое окно кистью белого цвета.
SetWindowWidth(w);	Устанавливает ширину графического окна.
SetWindowHeight(h);	Устанавливает высоту графического окна.
SetWindowSize(w,h);	Устанавливает ширину и высоту графического окна.
CloseWindow;	Закрывает графическое окно.

Способы и стили заливки фигур



Замечание 1:

Процедуры `Setpencolor(color)`, `Setpenstyle(ps)`, `Setbrushcolor(color)`, `Setbrushstyle(bs)`, `SetPenWidth(w)` пишутся непосредственно перед объектом редактирования, если фигура рисуется одной графической процедурой, например: окружность, эллипс, прямоугольник и т.п..

Если же фигура является результатом использования нескольких графических процедур, например, треугольник, многоугольник и т.п., то все вышеперечисленные процедуры, кроме `Setpencolor(color)`, `Setpenstyle(ps)` пишутся после объекта редактирования



Замечание 2: Процедура `Floodfill(x,y,c)` используется для заливки фигур, являющихся результатом использования нескольких графических процедур, и пишется непосредственно после рисования объекта. Точка (x,y) должна находиться внутри построенного объекта, c – это цвет заливки.

Для заливки фигур, построенных одной графической процедурой используется процедура `Setbrushcolor(color)`, где $color$ – цвет заливки.

Выполнение готовых программ



Задача 1: нарисовать

```
Program pr;
usesGraphABC;
Begin
SetWindowSize(600,400);
Line (200,200,400,200); LineTO (300,140); lineTO (200,200);
FloodFill (300,170,clblue);
Line (200,200,400,200); LineTo (300,260); LineTo (200,200);
FloodFill (300,230,cllime);
circle (160,200,40);
FloodFill (160,200,clred);
circle (440,200,40);
```

```
FloodFill (440,200,clyellow);
```

```
End.
```

Поэкспериментируйте с программой меняя цвет и толщину линий



Задача2: построить график функции $y:=-\cos(x)$;

```
Program cosinus;
```

```
uses GraphABC;
```

```
var
```

```
x,y:real;
```

```
Begin
```

```
SetWindowSize(400,400);
```

```
Line (200,0,200,400);
```

```
Line (0,200,400,200);
```

```
x:= -10;
```

```
while x<=10 do
```

```
begin
```

```
y:=-cos(x);
```

```
circle
```

```
(trunc(x*20+200),trunc(y*20+200),2);
```

```
x:=x+0.01
```

```
end
```

```
End.
```



Задача 3.

```
program z1;  
uses graphABC;  
begin  
  setpencolor(clred);  
  setbrushcolor(clnavy);  
  ellipse(40,150,250,210);  
  setbrushcolor(clyellow);  
  ellipse(340,150,550,210);  
  setbrushcolor(clred);  
  circle(295,190,25);  
  setbrushcolor(clolive);  
  ellipse(265,225,325,425);  
  setbrushcolor(cllime);  
  ellipse(265,5,325,150);  
end.
```

Поэкспериментируйте с программой, меняя цвет и толщину линий



задача 4

```
program z2;  
uses graphABC;  
begin  
  setwindow(500,500);  
  setpenwidth(3);  
  setbrushcolor(clfuchsia);  
  circle(225,160,50);  
  line(225,160,225,180);  
  line(210,190,240,190);  
end.
```

```

line(225,210,225,250);
line(100,100,200,260);
line(200,260,400,260);
line(210,350,200,480);
line(240,350,250,480);
rectangle(200,230,250,350);
setbrushcolor(cllime);
fillrect(0,480,500,500);
setbrushcolor(clwhite);
circle(205,150,10);
circle(245,150,10);
end.

```

Поэкспериментируйте с программой, меняя цвет и толщину линий



Задача 4.

```

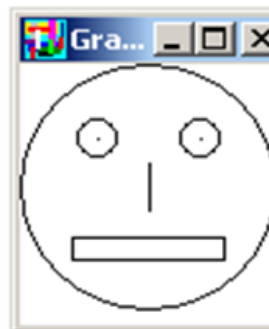
uses graphABC;

begin

Circle(50,50, 50);
Rectangle(20,70,80,80);
Circle(30,30, 8);
Circle(70,30, 8);
Line(50,40,50,60);
SetPixel(30,30,clblack);
SetPixel(70,30,clblack);

end.

```



задача 5 Наберите программу рисования окружности.

program z3;

```

usesgraphABC;
begin
  setpencolor(claqua);           {устанавливает цвет рисования окружности
- бирюзовый}
  setpenstyle(psdot);           {устанавливает стиль линии рисования
окружности –
  пунктирная линия}
  setpenwidth(10);              {устанавливает ширину линии рисования
окружности }
  setbrushstyle(bscross);       {устанавливает стиль заливки окружности}
  setbrushcolor(clred);         {устанавливает цвет заливки окружности}
  circle(225,160,50);          {рисует окружность}
end.

```



Задача 6

```

programz4;
usesgraphABC;
begin
  setpencolor(clpurple);        {устанавливает цвет рисования линий -
фиолетовый}
  line(100,100,400,300);       {рисует линию от точки (100,100) до точки
(400,300)}
  line(400,300,20,300);        {рисует линию от точки (400,300) до точки
(20,300)}
  line(100,100,20,300);        {рисует линию от точки (100,100) до точки
(20,300)}
  setbrushstyle (bsvertical);   {устанавливает стиль заливки окружности –
вертикальными линиями}
  floodfill(150,250,clbrown);   {заливает полученную замкнутую область (в
программе треугольник)коричневым цветом, начиная с точки

```

(150,250}

end.



Задача 7

Программа:

```
program qq;  
GraphABC;  
procedure Pifagor(x0, y0, a, L: real;  
N: integer);  
const k = 0.6;  
var x1, y1: real;  
begin  
if N > 0 then begin  
x1 := x0 + L*cos(a);  
y1 := y0 - L*sin(a);  
Line (round(x0), round(y0), round(x1), round(y1));  
Pifagor (x1, y1, a+pi/3, L*k, N-1);  
Pifagor (x1, y1, a+3*pi/30, L*k, N-1);  
Pifagor (x1, y1, a-3*pi/30, L*k, N-1);  
Pifagor (x1, y1, a-pi/3, L*k, N-1);  
end;  
end;  
begin  
Pifagor (250, 400, pi/2, 150,6);  
end.
```

Но если немного изменить программу:

```
program qq;  
uses GraphABC;  
procedure Pifagor(x0, y0, a, L: real;  
N: integer);
```

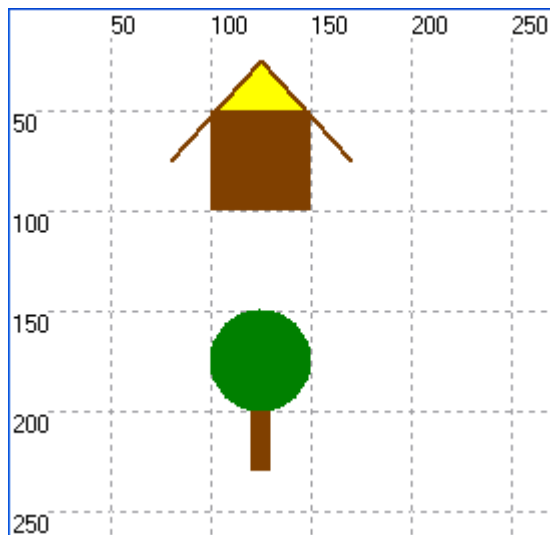
```

const k = 0.6;
var x1, y1: real;
begin
if N > 0 then begin
x1 := x0 + L*cos(a);
y1 := y0 - L*sin(a);
Line (round(x0), round(y0),round(x1), round(y1));
SetPenColor (clbrown);
Pifagor (x1, y1, a+pi/3, L*k, N-1);
SetPenColor (clyellow);
Pifagor (x1, y1, a+3*pi/30, L*k, N-1);
SetPenColor (clred);
Pifagor (x1, y1, a-3*pi/30, L*k, N-1);
SetPenColor (clgreen);
Pifagor (x1, y1, a-pi/3, L*k, N-1);
end;
end;
begin
Pifagor (250, 400, pi/2, 150,10);
end.

```



Задание 8. Определите координаты и составьте программу, выводящую на экран рисунок дома и дерева.



Программа будет иметь вид:

```

Program domik_2;
uses GraphABC;
var i: integer;
st: string;
begin

SetWindowSize(500, 500);
{Коричневая стена}
SetPenWidth(2);
SetBrushColor(clBrown);
FillRect(100, 50, 150, 100);
{Крыша желтого цвета}
SetPenColor(clBrown);
Line(125, 25, 80, 75);
Line(125, 25, 170, 75);
FloodFill(125, 30, clYellow);
{Крона дерева}
SetPenColor(clGreen);
SetBrushColor(clGreen);
Ellipse(100, 150, 150, 200);

```

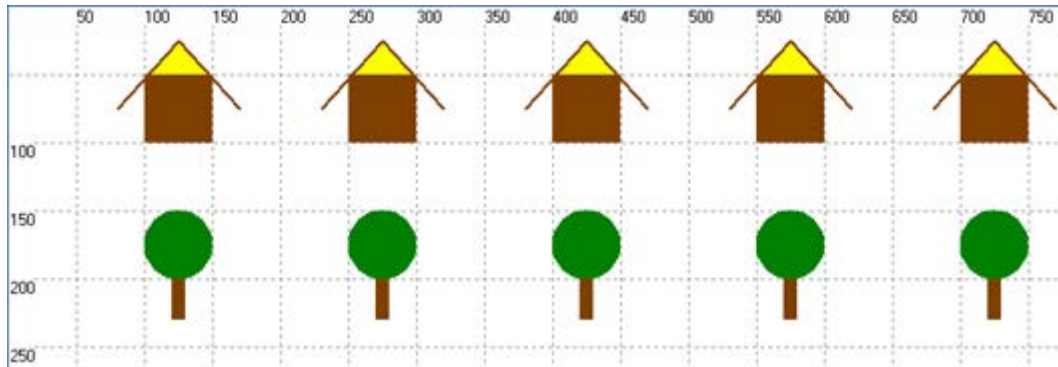
```

{стволдерева}
SetBrushColor(clBrown);
FillRect(120,200,130,230);
end.

```



Задание 9. Используя оператор цикла и введя переменную для пересчета координат по оси x , постройте “поселок”, состоящий из 5 домов. Внесите соответствующие дополнения и изменения в предыдущую программу.



Весь наш “поселок” выстроился вдоль горизонтальной оси экрана – оси X . Построение рисунка начинается с левого верхнего угла стены первого дома – точки с координатами $(100, 50)$. Координата Y не изменяется. Чтобы начать рисовать второй домик, нужно координату X увеличить на 150 (50 точек – ширина первого дома и 100 точек – расстояние между домиками).

Выберем в качестве параметра цикла целочисленную переменную X .

Для всех элементов нашего рисунка абсолютное значение координаты X заменим на относительное. Например, для стены дома процедура для рисования запишется следующим образом:

```
FillRect(x,50,x+50,100);
```

Сформулируем условие выполнения циклических действий для нашей задачи. Какие координаты имеет левый верхний угол пятого дома? Конечное значение выбранного нами параметра цикла $x = 700$. Тогда условие выполнения цикла записывается так: $x \leq 700$.

Словесное описание алгоритма коротко можно записать так:

Переменной цикла x присвоить начальное значение 100. Пока $x \leq 700$ выполнять серию действий “Нарисуй дом и дерево”, после каждого фрагмента рисунка увеличивать значение переменной цикла на 150.

Программа будет иметь вид:

```
ProgramDomik;
usesGraphABC;
var i: integer;
st:string;
x:integer;
begin
  SetWindowSize(800, 500);
  x:=100;
  While x<=700 do
  begin
    {Коричневая стена}
    SetPenWidth(2);
    SetBrushColor(clBrown);
    FillRect(x,50,x+50,100);

    {Крыша желтого цвета}
    SetPenColor(clBrown);
    Line(x+25,25,x-20,75);
    Line(x+25,25,x+70,75);
    FloodFill(x+25, 30,clYellow);
    {Крона дерева}
    SetPenColor(clGreen);
    SetBrushColor(clGreen);
    Ellipse(x,150,x+50,200);
    {ствол дерева}
```

```
SetBrushColor(clBrown);  
FillRect(x+20,200,x+30,230);  
x:=x+150;  
end;  
end.
```

Задания для самостоятельного решения

Вариант № 1

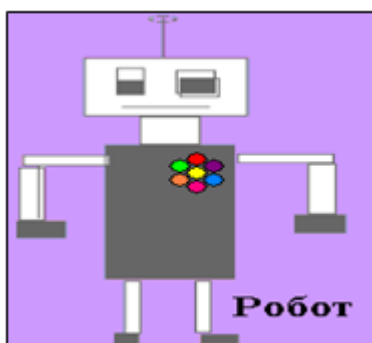
1. Построить график функции $y=\cos(x+5)$
2. а) Нарисовать весёлую рожицу. б) размножить ее



3. нарисовать

Вариант № 2

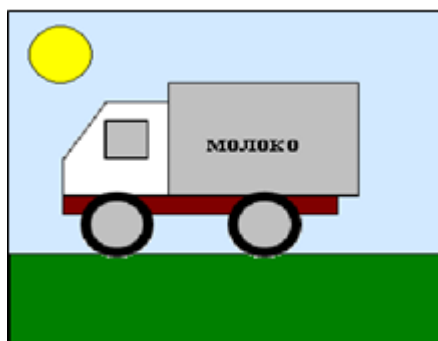
1. Построить график функции $y=\cos(2x-5)$
2. а) Нарисовать семицветную радугу б) размножить его



- 3.

Вариант № 3

1. Построить график функции $y=\sin(2x+7)$
2. а) Построить снеговика. б) размножить его



3.

Вариант № 4

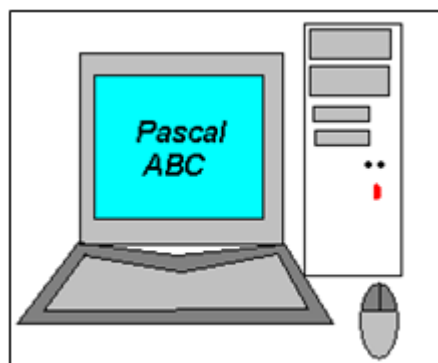
1. Построить график функции $y=\sin(x-3)$
2. а) Построить домик. б) размножить его



3.

Вариант № 5

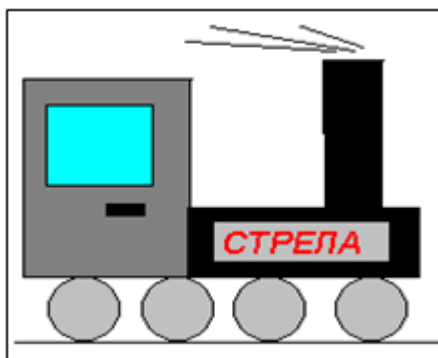
1. Построить график функции $y=\tan(2x-3)$
2. а) Нарисовать новогоднюю ёлку. б) размножить ее



3.

Вариант № 6

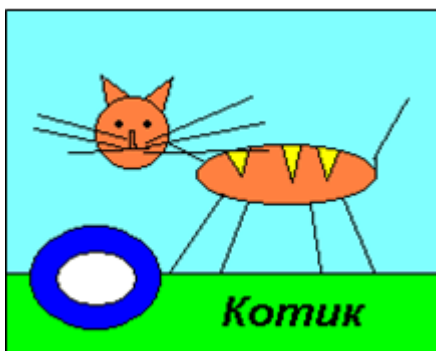
1. Построить график функции $y=\tan(3x-5)$
2. а) Построить кораблик. б) размножить его



3.

Вариант № 7

1. Построить график функции $y=\cos(3x-5)$
2. а) Составьте программу рисования часов типа «кукушка». б) размножить ее



3.

Вариант № 8

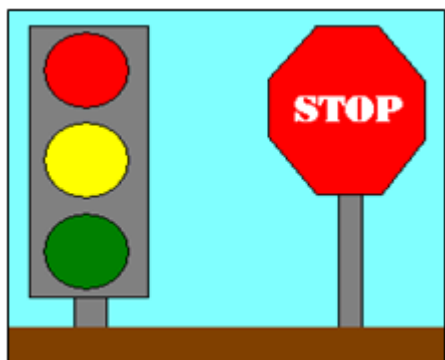
1. Построить график функции $y=\cos(-3x+5)$
2. а) Нарисуйте компьютер. б) размножить его



3.

Вариант № 9

1. Построить график функции $y=3\cos(-x+4)$
2. а) Постройте избушку на куриных ножках. б) размножить ее.



3.

Вариант № 10

1. Построить график функции $y=\text{ctg}(-x+4)$
2. а) Составьте программу рисования автомобиля. б) размножить ее.



3.

Вариант № 11

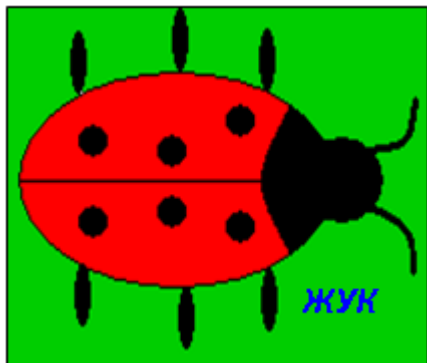
1. Построить график функции $y=\text{ctg}(x-4)$
2. а) Постройте куб и подпишите вершины. б) размножить ее.



3.

Вариант № 12

1. Построить график функции $y = \text{ctg}(x-4)$
2. а) Составьте программу рисования кошки. б) размножить ее.



3.

Контрольные вопросы

1. Что такое CloseGraph?
2. Что такое Setfillstyle?
3. Что такое GraphResult?