Ханнанова С.Т.

Учимся программировать Pascal

Пособие для учащихся

СОДЕРЖАНИЕ

Введение 3
§ 1. Структура программы в Паскале 4
§ 2. Особенности записи программы на языке Паскаль 5
§ 3. Типы данных 6
§ 4. Оператор присваивания 7
§ 5. Ввод и вывод данных. Линейные алгоритмы 9
§ 6. Математические операции 11
§ 7. Работа с переменными. Линейные алгоритмы 13
§ 8. Целочисленная арифметика 15
§ 9. Алгоритмы ветвления 16
§ 10. Условный оператор. Решение задач 18
§ 11. Логические операции и выражения 19
§ 12. Целочисленная арифметика и условный оператор 22
§ 13. Операторы цикла в Паскале 23
§ 14. Последовательная детализация алгоритма 27

ВВЕДЕНИЕ

Языки программирования, их назначение, особенности

Центральным понятием программирования является **алгоритм**. С него начинается работа над программой, а от качества алгоритма зависит ее успешное завершение. Поэтому учится программировать, прежде всего, означает учиться разрабатывать хорошие алгоритмы и применять те, что уже известны.

Алгоритм необходимо записать. Это можно сделать на русском языке, на языке графических схем, наконец, на алгоритмическом языке.

Существует множество языков программирования, они обладают разными достоинствами и недостатками, некоторые имеют специфическую направленность.

Языки программирования можно разделить на две группы – языки высокого уровня и низкого уровня (машинные).

К языкам низкого уровня относятся язык Ассемблер, в котором программа пишется в основном на уровне машинных кодов. Языки высокого уровня позволяют писать программу с помощью условных обозначений, близких к языку человека.

Мы начнем знакомство с современным и пригодным для профессиональной работы языком Паскаль (Pascal). Он был создан швейцарским профессором **Никлаусом Виртом** в 1971 году специально для обучения программированию и был назван в честь французского математика и физика Блеза Паскаля, впервые создавшего механическое вычислительное устройство.

Компьютеры работает на двоичных кодах. Следовательно, алгоритм, введенный в компьютеры на каком-либо языке программирования, должен быть преобразован в специальные коды. Для этого в состав языка программирования входят специальная программа – **транслятор**, которая и выполняет эту задачу.

Трансляторы можно разделить на две группы по их работе – **компиляторы** и **интерпретаторы.**

Интерпретатор читает строку программы, транслирует ее в коды ЭВМ и немедленно выполняет, затем переходит к следующей строке и вновь повторяет все действия. При этом происходит проверка правильности написания строк программы с точки зрения правил языка. При обнаружении ошибок специальный отладчик сообщает об ошибке, исполнения программы.

Компилятор действует иначе. Он сначала просматривает программу, отмечает все ошибки и только после того, как все ошибки исправлены, - компилирует программу, т.е. в памяти или на диске создает программу в машинных кодах и после этого выполняет.

Pascal ABC и PascalABC.NET. История развития

Система создавалась на факультете математики, механики и компьютерных наук ЮФУ как учебная среда программирования (автор — доцент С. С. Михалкович).

Система представляла из себя фактически интерпретатор языка программирования Паскаль с интегрированной оболочкой. Язык, в основном, соответствовал входному языку Object Pascal. СистемаPascal ABC разрабатывалась в среде Delphi для операционных систем, использующих Win32 API.

Несмотря на неполную реализацию языка, система стала удачной заменой уже давно устаревшей системе Turbo Pascal в первоначальном обучении программированию.

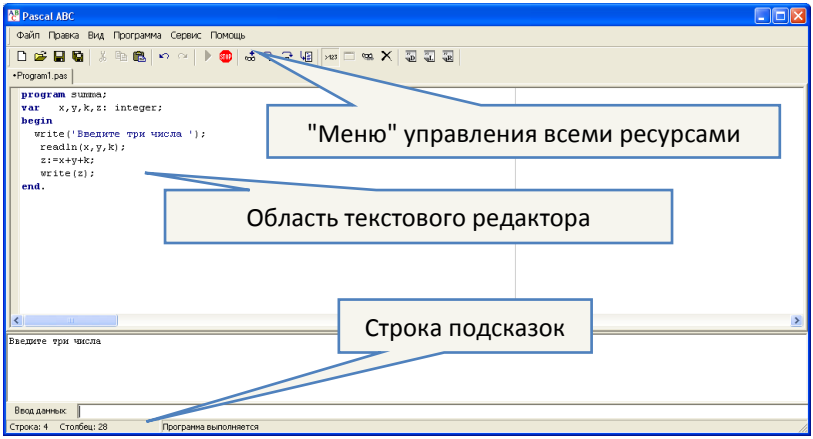
В 2005—2006 система была полностью переработана: изменён язык реализации — на C#, и изменена её архитектура — на полноценный компилятор языка Object Pascal с расширениями, связанными с платформой .NET. Новая система получила название PascalABC.NET.

Pascal ABC и PascalABC.NET всех версий является бесплатным (freeware) программным обеспечением.

§ 1. СТРУКТУРА ПРОГРАММЫ В ПАСКАЛЕ

1.1 Структура программы в Паскале

program (имя программы); заголовок программы
label (список меток);
const (список постоянных значений);
type (описание сложных типов данных);
var (описание данных программы); описание переменных
begin (начало программы)
(алгоритм)
end. (конец программы) терминатор



1.2 Создание программы

PascalABC.exe – файл программы. При запуске системы появляется окно редактора текстов программ (его можно использовать и как текстовый редактор). Если экран пуст, то можно сразу набирать текст программы, делая такие же отступы, какие имеются в примере. Эти отступы облегчают чтение текста и поиск ошибок. Если на экране после запуска системы находится ненужная программа, то следует войти в пункт меню **Файл** и выполнить команду **Новый**. Набор каждой строки программы завершается нажатием клавиши **<Enter>.**

Начнем знакомство с Паскалем с программы, которая считывает два числа с клавиатуры, складывает их и выводит сумму на экран.

Текст программы	**Комментарий**
program summa;	заголовок программы
var x,y,z: integer;	описание переменных
begin	начало программы
write('введите два числа ');	вывод на экран
readln(x,y);	ввод значений x и y
z:=x+y;	присваивание суммы
write(z);	вывод результата
end.	Конец программы (терминатор)

1.3 Запуск программы

Для выполнения программы надо выйти в меню и в пункте **Программа** выполнить команду **Выполнить** (или нажать клавишу **F9)**. Система сначала запускает транслятор, который переводит программу с Паскаля на язык машинных кодов и ищет синтаксические ошибки в программе. Если они найдены, то программа не будет выполняться, произойдет возврат в редактор. В строке состояния появляется красная строка с сообщением об ошибке. Когда все ошибки исправлены, программа начинает выполняться.

Лабораторная работа.

1. Запустите **PascalABC** и наберите текст, приведенной программы. Для перехода с русского на английский и наоборот используйте настройки Windows**.**
2. Запустите программу на выполнение. Если нет ошибок, то всплывает окно с текстом «Введите два числа» и полем ввода.
3. Наберите на клавиатуре через пробел два целых числа и нажмите **<Enter>.**
4. После выполнения программы на экране внизу появится окно редактора, где можно посмотреть полученный результат**.**

Текст программы **:**

```
program summa;
var     x,y,z: integer;
begin
        write('введите два числа ');
        readln(x,y);
        z:=x+y;
        write(z);
end.
```

5. Сохраните программу. Нажмите **<Ctrl+S>,** в появившемся окне введите имя файла (например, PRIM1_1).

6. Составьте программу для нахождения суммы трех чисел. Сохраните ее.

§ 2. ОСОБЕННОСТИ ЗАПИСИ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ

2.1 Основные правила записи

Вашу программу будет выполнять машина, а она все делает буквально. Поэтому правила написания программы очень жестки, но их немного. Одно утешает, что правила русского языка намного сложнее.

Для начала рассмотрим программу, которая должна была у вас получиться при выполнении 3 задания предыдущего §.

```
program summa;
var     x,y,k,z: integer;
begin
        write('Введите три числа ');
        readln(x,y,k);
        z:=x+y+k;
        write(z);
end.
```

На этом примере видны ***основные особенности записи*** программы:

1) всякая программа начинается с заголовка – слова **PROGRAM** и следующего за ним названия программы; (в последних версиях Паскаля это предложение писать не обязательно)
2) после заголовка располагается описательная часть программы;
3) между служебными словами **BEGIN** и **END** записывается алгоритм решения задачи;
4) команды (операторы) разделяются точкой с запятой;
5) текст который выводится на экран (русский текст) заключается в апострофы ‘ … ‘ ;
6) перечисляемые объекты разделяются запятой.

2.2 Имена и зарезервированные слова

Тексты программы записывается при помощи латинских букв, цифр и знаков. Буквы допускаются прописные и строчные, причем для компилятора записи **VAR**, **vAR**, **VaR** идентичны.

Особую роль в тексте программы играют имена (индикаторы) и зарезервированные слова. Имена применяются для обозначения программы и ее объектов. Имя может состоять из любого количества букв или цифр, но должно начинаться с буквы. В имя можно включать знак подчеркивания (например Prim_1). В программе **summa** пять имен: **summa, x, y, k, z.** Программисты часто используют осмысленные имена в своих программах. Это всегда полезно, а в сложных программах совершенно необходимо.

Зарезервированные слова применяют для обозначения операторов (команд) и других элементов языка Паскаль. Их нельзя использовать в качестве имен и во всех программах они имеют одинаковый смысл. зарезервированными словами в нашем примере являются слова: **program** (программа), **var** (переменные), **begin** (начало), **read** (читать), **writeln** (писать), **end** (конец).

2.3 Константы и переменные

Данные, которыми оперирует программа, могут быть определены в ней как неизменные, либо как способные изменять свое значение в ходе выполнения программы. Первые называются константами, а вторые переменными. И переменные и константы размещаются в памяти компьютера (в так называемой «ячейке памяти»). В программе переменные должны быть описаны в предложении **var** (от слова VARIABLE-переменная), а константы – в предложении **const**. У всякой ***величины*** имеется три основных свойства: ***имя, значение, тип***.

§ 3. ТИПЫ ДАННЫХ

3.1. Тип INTEGER (целый)

И в жизни и в программировании очень полезно использовать понятие типа.

Любая константа или переменная, использованная в программе, принадлежит к определенному типу. Тип задает множество допустимых значений переменных, внешний вид констант, возможные операции над значениями.

Значения величины типа **INTEGER** не может быть меньше **–32768** или больше **32767.**

Переменные должны быть перечислены в описательной части программы в предложении

var имя _переменной: integer;

Над величинами целого типа допустимы ***арифметический*** операции:

+ (сложение), **–** (вычитание), ***** (умножение),
div (деление нацело),
mod (нахождение остатка от целочисленного деления).

Все операции вырабатывают результат **целого** типа. Например,

15 div 4 = 3,
25 mod 4 = 1.

Над целыми разрешено и обычное деление, оно обозначается косой чертой «/» и дает результат ***вещественного*** типа.

3.2. Тип REAL(вещественный)

Константы вещественного типа (числа с дробной частью) изображаются с десятичной ***точкой***:

12.3, –1.5, –0.75 или в ***показательной (экспоненциальной)*** форме: -0.45E5, 6.7E-10, 0.355E6

Для получения числа в обычном виде надо перенести запятую на число разрядов указанных после **E** вправо, если число положительное, влево, если отрицательное.

Например: 6.7E-10=0.00000000067.

Вещественные переменные требуют описания предложением

var имя: real;

Над величинами вещественного типа допустимы арифметические операции:

+ (сложение), **-** (вычитание), ***** (умножение), **/** (деление).

3.3. Диапазоны допустимых значений

Вы познакомились с представителями вещественного и целого типов. На самом деле и тот и другой имеет несколько видов отличающихся диапазоном допустимых значений.

В следующей таблице приведены 5 стандартных **целых** типов

тип	Значение	формат
shortint	-128..127	знаковый
integer	-32768..32767	знаковый
longint	-2147483648.. –2147483647	знаковый
byte	0..255	без знаковый
word	0..65535	без знаковый

и 5 стандартных **вещественных** типов

тип	Значение	Число значащих чисел
real	$2.9*10^{-39}..1.7*10^{38}$	11..12
single	$1.5*10^{-45}..3.4*10^{38}$	7..8
double	$5.0*10^{-324}..1.7*10^{308}$	15..16
extended	$3.4*10^{-4932}..1.1*10^{4932}$	19..20
comp	$-2*10^{63}+1..+2*10^{63}-1$	19..20

§ 4. ОПЕРАТОР ПРИСВАИВАНИЯ

Оператор присваивания придает переменной конкретное значение, например:

x:=2; y:=5,

одновременно уничтожая старое.

Редкая программа обходится без оператора присваивания.

Присваивать можно значение другой переменной или результат вычисления арифметического выражения:

```
a:=b;
a:=b+c;
x:=y+2-z.
```

Формат команды:

<имя переменной>:=<выражение>

исполнение команды присваивания происходит в таком порядке: сначала вычисляется <выражение>, затем полученное значение присваивается переменной.

Пример 1: Пусть переменная А имела значение 6. Какое значение получит переменная А после выполнения команды: А:=2 * А – 1?

Решение:

Вычисление выражения 2* А – 1 при А=6 даст число 11.

Значит новое значение переменной А будет равно 11.

Пример 2: ***Поменяйте между собой значения двух переменных А и В***, воспользовавшись третьей переменной R для временного хранения значения.

Решение:

```
program  prim2_2;
var    a, b, r: real;
begin
        write('Введите два числа ');readln(a,b);
        b:=r;
        r:=a;
        a:=b;
        write(' a=' , a, '  b=' , b);
end.
```

Задание для тренировки.

1). Найдите значения переменных, если это возможно. Учтите, что число 7.0 является вещественным, т.к. оно имеет дробную часть, хотя и равную нулю. Операции MOD и DIV можно выполнять только над целыми числами:

a) a:=21 div 5 b)a:= 2 mod 3
b:= 20 mod 5 b:= 36.0 mod 6
c:= 14 div 6.0 c:= 81 div 0
d:= 14 mod 0 d:= 38 div 6
e:= 5 mod 13 e:= 3 div 2

2). Определить конечное значение переменных X и Y в результате выполнения следующих алгоритмов:

а)
X:=2
X:=X * X
X:=X * X * X
X:=X * X* X * X

б)
X:=1.5
X:=2 * X + 1
Y:=X/2
Y:=X + Y
X:=X – Y

3). Поменяйте между собой значения трех переменных X, Y и Z по схеме тройного квартирного обмена:

$$X \to Y \to Z \to X.$$

4). Присвойте переменной N ее собственное значение, увеличенное в N раз.

5). Чему равно X в результате выполнения программы

```
X:=2;
X:=X+X;
X:=X-X
```

6). Указать значения величин a и b после выполнения следующих операторов присваивания:

а)
a:=5.8
b:=-7.9
b:= a
a:=b

б)
a:=0
b:=-9.99
b:=a
a:=b

7). Переменной А присвоить ее значение, увеличенное в N раз.

8). Чему станет равна переменная А в результате выполнения следующего алгоритма:
1) A := 2; 2) A := 3 * A - A; 3) A := 3 * A - A.

§ 5. ВВОД И ВЫВОД ДАННЫХ. ЛИНЕЙНЫЕ АЛГОРИТМЫ

Мало программ обходится без ввода данных, и совсем нет таких, которые не выводят полученные результаты. Написать такую программу можно, но кому она понадобится?

5.1 Ввод

Для сообщения данных компьютеру служит оператор ввода. Он помещает вводимое значение переменной в отведенную для него ячейку. Оператор ввода:

read (список переменных),

Где список переменных – последовательность имен переменных, разделенных запятыми.

Например, read (x,y,z);
read (beta);

Оператор **read** останавливает работу программы и ждет, пока пользователь наберет на клавиатуре число и нажмет <Enter>. Введенное число помещается в оперативную память, в отведенную ячейку, имеющую имя указанное в операторе. Если список ввода содержит несколько имен, то для каждого надо ввести свое значение. Вводимые числа разделяют пробелами или нажатием клавиши <Enter>.

Заканчивается ввод всегда клавишей <Enter>. После работы этого оператора курсор располагается за последним введенным символом, но не переводится на новую строку. Для перевода курсора на новую строку экрана дисплея после ввода данных, используется оператор

readln (список переменных).

Оператор **readln** отличается от **read** еще и тем, что, введя необходимое количество данных, пропускает все остальные, набранные до нажатия клавиши <Enter>.

5.2. Вывод

Для вывода результатов работы программы служит оператор

write(список вывода).

Список вывода может содержать имена переменных, числовые и текстовые константы, выражения. Элементы в списке разделяются запятыми. Если указана переменная, то на экран выводится ее значение, константа выводится без изменения, значения выражений вначале вычисляются, а затем высвечиваются на экране.

Вслед за выражением после двоеточия можно указать ширину поля экрана, в котором разместится выводимое значение.

Например, оператор **write(10:3, 55:6)**

высветит на экране .**10…..55** (точка означает пробел, пустую позицию экрана).

Вывод происходит в том месте экрана, где находится курсор.

При выводе вещественных значений можно указать, сколько десятичных цифр следует сохранить в дробной части числа. Количество цифр указывается вслед за шириной поля после двоеточия.

Например, если **X=3.14159, а Y=2.71468**,

то оператор write(**x:6:2,y:8:3**) высветит на экране **..3.14….2.715**.

Чтобы прокомментировать выводимые значения, в список вывода можно помещать строки любых символов, заключенные в апострофы (одинарные кавычки). Например,

write(‘Ответ:’, X:4,’км/сек.’).

Эти строки появятся на экране без кавычек. Так при X=3.5 этот оператор выведет:

Ответ: 3.5 км/сек.

Перевод курсора на новую строку осуществляется оператором пустого вывода **writeln**;

Если надо перевести курсор после вывода, то применяется оператор

writeln (список вывода).

Пример: Пусть требуется найти сумму, произведение и разность двух данных чисел. Для каждого из чисел надо придумать имя переменной и указать ее тип. Затем ввести эти числа в отведенные ячейки и, используя возможности оператора вывода напечатать результаты.

При решении задач имена присваиваются не только исходным данным, но и результатам, а также получаемым промежуточным значениям. Поскольку в рассматриваемом примере надо получить три результата, введем для них переменные X,Y,Z. В программе этим переменным будут присвоены значения суммы, произведения и разности двух вводимых чисел.

```
program prim_4;
var a,b,x,y,z:real;
begin
  write('Введите два числа через пробел, затем нажмите <enter>');
  readln(a,b);
  x:=a+b;
  y:=a*b;
  z:=a-b;
  writeln('a+b=',x);
  writeln('a*b=',y);
  writeln('a-b=',z);
end.
```

Задание для тренировки.

9). Даны катет и гипотенуза прямоугольного треугольника. Определить его площадь.
10). Даны катеты прямоугольного треугольника. Найти его периметр.
11). Даны два числа. Найти их среднее арифметическое.

5.3.Целочисленная арифметика. Задачи на целочисленное деление

В программировании существует целый класс задач, где действия производятся только с целыми числами. При решении подобных задач обычно используются операции над целыми ***mod*** и ***div***. Сегодня мы познакомимся с задачами на целочисленное деление.

Задача. Дано расстояние в сантиметрах. Найти число полных метров в нем.

Код программы:

```
program prim_4;
var a,b:integer;
begin
        write('введите расстояние в сантиметрах');
        readln(a);
        b:=a div 100;
writeln(b,' метров ');
end.
```

Задание для тренировки.

12). Дана масса в килограммах. Найти число полных центнеров в ней.
13). Дана масса в килограммах. Найти число полных тонн в ней.
14). Дано расстояние в метрах. Найти число полных километров в нем.
15). Дан прямоугольник с размерами 543x130 мм. Сколько квадратов со стороной 130 мм можно отрезать от него?
16). *Дано целое число k ($1 \le k \le 365$). Присвоить целочисленной величине n значение 1, 2,…,6 или 0 в зависимости от того, на какой день недели (понедельник, вторник, …, суббота или воскресенье) приходится k-й день года, в котором 1 января – понедельник.

§ 6. МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ

6.1 Арифметические выражения

Арифметические выражения строятся из имен переменных, констант, знаков операций, функций и круглых скобок так, как это принято в математике.

При вычислении их значений операции выполняются в порядке приоритета:

***, /, DIV, MOD**, а затем **+** и **-**.

Операции одинакового старшинства выполняются слева направо.

В Паскале имеются следующие стандартные функции

Функция	Назначение	Тип результата
abs(x)	Абсолютное значение X (модуль X)	Тип аргумента
arctan(x)	Арктангенс X	Вещественный
cos(x)	Косинус X	Вещественный
exp(x)	e^x	Вещественный
frac(x)	Дробная част X	Вещественный
int(x)	Целая часть X, обнуление дробной части	Вещественный
ln(x)	Натуральный логарифм	Вещественный
pi	Значение PI=3.1415926535897932385	Вещественный
round(x)	Округление до ближайшего целого	Целый
sin(x)	Синус X	Вещественный
sqr(x)	Квадрат X	Тип аргумента
sqrt(x)	Квадратный корень X	Вещественный
trunc(x)	Отбрасывание дробной части	Целый

Наряду с переменными и константами в арифметические выражения можно включать функции. При определении значения выражения, прежде всего, вычисляются значения входящих в него функций.

Аргумент функции обязательно заключается в скобки.

Выражение на Паскале, как впрочем, и на других языках программирования, записывается в одну строчку, а для сохранения порядка действий используются скобки. Все действия должны быть указаны. Например, 2x + xy надо записать как **2*x + x*y**.

В Паскале нет оператора возведения в произвольную степень. Возведение в степень осуществляется с использованием логарифмов. Вместо $\mathbf{a^x}$ пишется **exp(ln(a) * x).**

В Паскале существует только стандартная функция вычисления натурального логарифма, поэтому используется следующее математическое тождество: $\mathbf{\log_a b = \ln b / \ln a}$

6.2 Действия над данными разных типов. Преобразование типов

Все решения задач на Паскале связаны с вычислением различных математических функций, решением уравнений, вычислением формул, неравенств. В программе могут одновременно встречаться переменные разных типов. Как совместить переменные целого и вещественного типа?

Результат операций +, - и * зависит от типа аргументов. Если хоть один из них имеет тип **real**, то и результат будет иметь тип **real**. Это объясняется тем, что у данных типа **real** есть дробная часть, а у **integer** — нет. Даже если в вещественной переменной хранится целое число, оно все равно имеет дробную часть, только она равна нулю. То есть если хотя бы у одного из аргументов есть дробная часть, то в результате выполнения операции она никуда не исчезает. Поэтому результат тоже имеет дробную часть (**real**).

Операции целочисленного деления **div** и **mod** определены только для целых чисел. Поэтому результат тоже всегда целое число.

Если **a<b,** то при выполнении операции **a *mod* b** результат равен **a.**

Выражение может включать в себя и целые и вещественные члены. Наличие хотя бы одного вещественного члена или знака / приводит к тому, что значение результата будет вещественным.

Результат операции вещественного деления по определению всегда имеет дробную часть. Функции **int** и **frac** предназначены для выделения целой части и дробной части при этом результат будет вещественного типа

В программе могут одновременно встречаться переменные разных типов. Как их совместить? Функции **trunc** и **round** предназначены для преобразования типов. Они явно указывают на то, что сделать с дробной частью числа. Поэтому это возможность получить на Паскале из дробного числа целое.

Пример. Одновременное использование вещественных и целых типов

```
program mix;
var n,k: integer;
    a,b: real;
begin
n: =4;
a:=3.6;
b:=n; {в переменную типа real можнозаписать целое число}
writeln('b=',b); {в переменную типа integer нельзя записать вещественное число! чтобы
                  все-таки поместить число типа real в переменную типа integer, нужно явно указать, что делать с дробной частью числа. есть два варианта; }
n:=trunc(a); {функция trunc(x) возвращает целую часть числа x, то есть отбрасывает
                  дробную часть}
writeln('trunc (3.6)=',n);
k:=round(a); { функция round(x) округляет до ближайшего целого }
writeln ('round(3.6)=' ,к) ;
end.
```

при запуске программа выведет на экран следующее:

```
в:= 4.0000000000E+00
trunc(3.6)=3
round(3.6)=4
```

В фигурные скобки {…} в программе можно заключать комментарий, компилятор пропускает текст, заключенный в такие скобки, а комментарий позволяет вспомнить о чем программа если вы позднее возвращаетесь к ее тексту.

Задание для тренировки.

17). Напишите программу, которая запрашивает два числа, находит остаток от деления первого на второе и выводит результат.

18). Найдите значения переменных, если это возможно:

a) a:=round(6.9)	b)a:=round(15.39)
b:=ROUND(6.48)	b:=ROUND(15.8)
c:=trunc(9.5)	c:=trunc(-39)
d:=frac(9.5)	d:=frac(39)
e:=int(9.5)	e:=INT(39)
f:=TRUNC(-17)	f:=TRUNC(5.6)
g:=frac(17)	g:=frac(-0.3)
h:=INT(-17)	h:=int(1.25

19). Запишите на Паскале следующие выражения

a) $\frac{A+B}{C}+\frac{\sqrt{C}}{A-B}$ c) $\frac{A-B}{C}:\frac{A+B}{D}$ e) $\frac{A\cdot B}{C}-\frac{D}{A\cdot B}$

b) $\frac{X+Y}{3}\bullet\frac{7}{X-Y}$ d) $3\frac{1}{2}+\frac{C}{(A+B)^2}$ f) $\frac{X^2}{Y}+\frac{Y^2}{X}$

20). Найдите значения переменных, если это возможно:

a) A:=SQR(100)
B:=sqrt(100)
C:=SQR(-10)
D:=sqrt(-10)
E:=SQR(0.9)
F:=SQRT(0)

b) A:=sqrt(9)
B:=SQR(9)
C:=SQRT(-9)
D:=sqr(-9)
E:=SQRT(0.0)
F:=SQR(0.1)

21). Запишите по правилам языка программирования следующие выражения:

a) $\sqrt{x_1^2+x_2^2}$ d) $\sqrt{1-\sin^2 x}$ g) $mg\cos a^2$

b) $\frac{1}{1+\frac{1}{2+\frac{1}{2+\frac{3}{5}}}}$ e) $\frac{-b+\frac{1}{a}}{\frac{2}{c}}$ h) $\frac{1}{1+\frac{a+b}{2}}$

c) $\frac{-b+\sqrt{b^2-4ac}}{2a}$ f) $\frac{\sqrt{x+1}+\sqrt{x-1}}{2\sqrt{x}}$ i) $\frac{ab+bc}{ad}$

22). Составьте программу нахождения периметра квадрата, если задана его площадь.
23). Заданы стороны треугольника. Определить его площадь.
24). Заданы стороны прямоугольника. Определить его площадь и длину диагонали.
25). Заданы радиус основания и высота цилиндра. Определить площадь его поверхности и объем.
26). Найти площадь кольца по заданным внешнему и внутреннему радиусам.
27). *Поменять местами значения переменных X и Y, не используя дополнительной переменной.
28). Заданы длина, ширина и высота параллелепипеда. Определить его объем и площадь поверхности.

§ 7. РАБОТА С ПЕРЕМЕННЫМИ. ЛИНЕЙНЫЕ АЛГОРИТМЫ

7.1 Обмен значениями без дополнительной переменной

Начнем с разбора 28-го задания предыдущего § . Первая мысль, приходящая в голову, это написать программу, похожую на эту:

```
A := B;
B := A;
```

Но эта программа работать не будет (в обеих переменных будет значение B). Теперь поищем правильное решение.

Обозначим начальное значение A за A1, B за B1. Тогда необходимо, чтобы по окончании работы программы A равнялось B1, а B - A1.

1) A=A1; B=B1;

2) Занесем в переменную А результат суммирования А и В (А := А + В):
А = А1 + В1; В = В1;
3) Занесем в переменную В разность А и В (В := А – В):
т.к. А = А1 + В1; то В=(А1+В1)-В = А1;
4) Занесем в переменную А разность А и В (А := А – В):
А = В1; В = А1;

Код программы:

```
program prim_4;
var a,b:integer;
begin
        write('Введите два числа ');
        readln(a,b);
        a:=a+b;
        b:=a-b;
        a:=a-b;
        writeln('a=',a);
        writeln('b=',b);
end.
```

7.2 Эффективные алгоритмы

В Паскале отсутствует возможность возведения в степень, не считая квадрата. Поэтому для получения a^{20} нужно a*a*a*a…*a 19 раз. Но если учесть, что результат умножения можно сохранить в промежуточной переменной, ответ можно найти за 5 действий.

```
program prim_4;
var a,b:real;
begin
   write('введите число'); readln(a);
   b:=a*a;      {получаем а во 2 }
   b:=b*b;      { получаем а в 4 }
   b:=a*b;      { получаем а в 5 }
   b:=b*b;      { получаем а в 10 }
   b:=b*b;      { получаем а в 20 }
   writeln('а в 20 степени=',b:0:2);
            {вместо указания ширины поля для вывода числа лучше ставить 0, тогда компилятор отведет столько позиций под целую часть, сколько получилось в ответе, а для дробной части я посчитала возможным оставить 2 позиции}
end.
```

Задание для тренировки.

29). Дано вещественное число А. Не пользуясь никакими арифметическими операциями, кроме умножения, получить:
1. A^7 за четыре операции;
2. A^8 за три операции;
3. A^9 за четыре операции;
4. A^{13} за пять операции;
5. A^{15} за пять операции;
6. A^{19} за пять операции;
7. A^{21} за шесть операции;
8. A^{28} за шесть операции;

30). Переменным А и В присвоить значения: А := А+В, В := А - В. Третью переменную не использовать. Записать алгоритм.

§ 8. ЦЕЛОЧИСЛЕННАЯ АРИФМЕТИКА

Очень часто необходимо чтобы программа определила, из каких цифр состоит число, или определила разряд заданной цифры или наоборот цифру в заданном разряде. Если вы попросите человека решить эту задачу, проблем не возникнет, а как быть с компьютером. Для него любое число это набор нулей и единиц, т.е. двоичный код. В отличие от нас компьютер все действия выполняет в двоичной системе. Например, число 27 хранится в его памяти как 11011. И где здесь 2 и 7?

Для решения этих задач надо вспомнить, что собой представляет любое число десятичной системы счисления (т.е. той, к которой мы с вами привыкли). В позиционной системе счисления (к которой относится десятичная система) величина, обозначаемая цифрой в записи числа, зависит от ее позиции. Например, в числе 333 первая цифра обозначает три сотни, вторая - три десятка, третья – три единицы. Любое число можно записать в виде:

$$32478=3*10000+2*1000+4*100+7*10+8=3*10^4+2*10^3+4*10^2+7*10^1+8*10^0$$

Поэтому для обработки десятичных чисел используется 10 в соответствующей степени.

Например. Надо получить число, образованное при перестановке цифр заданного числа.

```
program prim_6;
var n,x1,x2,m:integer;
begin
    write('введите двузначное число');
    readln(n);
    x1:=n mod 10;{выделяем из числа единицы}
    x2:=n div 10;  {получает число десятков в числе}
    m:=x1*10+x2; {число единиц умножаем на 10, получаем десятки}
writeln(m);
end.
```

Пусть дано n= 27.
x1:= 27 mod 10 =7
x2:= 27 div 10 = 2
m:=7*10+2 = 72, что и требовалось получить.

К сожалению, таким образом, мы можем определить лишь крайние цифры числа. А если цифра, которая нам нужна, стоит не с краю? Не беда, сделаем ее крайней. Например, дано трехзначное число, надо определить среднюю цифру числа.

Пусть a=246
b:=a div 10 =24
b:=b mod 10=4
или по другому:
b:=a mod 100=46
b:=b div 10 =4

Задание для тренировки.

31). Дано двузначное число. Найти:
   a. число десятков в нем;
   b. число единиц в нем;
   c. сумму его цифр;
   d. произведение его цифр.

32). Дано трехзначное число. Найти:
   a. Число единиц в нем;
   b. Число десятков в нем;
   c. Сумму его цифр;
   d. Произведение его цифр.

33). Дано трехзначное число. Найти число, полученное при прочтении его цифр справа налево.
34). Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее в конце. Найти полученное число.
35). Дано трехзначное число. В нем зачеркнули последнюю справа цифру и приписали ее в начале. Найти полученное число.
36). Дано трехзначное число. Найти число, полученное при перестановке первой и второй цифр заданного числа.
37). Дано трехзначное число. Найти число, полученное при перестановке второй и третьей цифр заданного числа.
38). *Дано вещественное число А, содержащее два знака до запятой и два после. Получить новое число, поменяв в числе А целую и дробную части.
39). *В кассе имеются купюры достоинством в К рублей и в 1 рубль. Выдать N рублей минимальным набором купюр заданного достоинства.

§ 9. АЛГОРИТМЫ ВЕТВЛЕНИЯ

В практике хорошо известны задачи, дальнейший ход решения которых зависит от выполнения какого-либо условия. В жизни часто приходится действовать в зависимости от обстоятельств, от каких-то условий. Но если в жизни мы часто ищем выход из ситуации только тогда, когда попали в неё, в программе необходимо предусмотреть все действия которые необходимо выполнить после проверки условия, как в случае его выполнения, так и в случае невыполнения. Чтобы изменять последовательность выполнения различных частей программы, применяют условный оператор.

9.1 Условный оператор

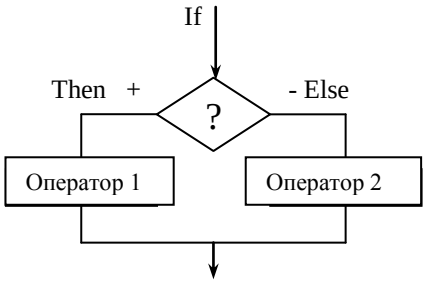
Условный оператор позволяет выполнять или пропускать операторы программы в зависимости от некоторого условия. Условный оператор может иметь две формы:

Полный оператор

Формат команды:

IF *условие* THEN *оператор_1* ELSE *оператор_2;*

Схема оператора

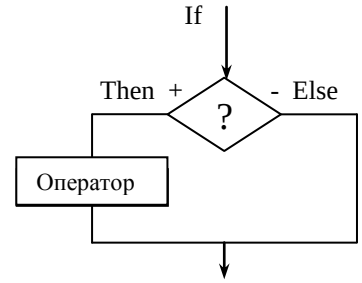


Неполный оператор

Формат команды:

IF *условие* THEN *оператор;*

Схема оператора



Если перевести на русский язык английские слова то получим

ЕСЛИ условие ТО оператор_1 ИНАЧЕ оператор_2; и ли
ЕСЛИ условие ТО оператор;

В качестве условия применяют операции сравнения: **=, <>,<=,>=,<,>.** Слева и справа от знака сравнения записывают арифметические выражения.

Например, оператор

```
If x<>o then z:=y/x
        else write(‘ Ошибка!‘);
```

присвоит переменной Z значение частного y/x, если x<>0, в противном случае высветит на экране слово «Ошибка!».

9.2 Составной оператор

В некоторых случаях после слов THEN и ELSE надо выполнить не один оператор, а несколько. Тогда эти операторы заключаются в так называемые операторные скобки, где BEGIN- открывающая скобка, END –закрывающая скобка. Все операторы находящиеся внутри операторных скобок называются составным оператором.

Перед словом ELSE точка с запятой никогда не ставится.

Формат команды: BEGIN оператор; оператор;…оператор END;

Например:

```
if a<b then  begin
                r:=a;
                a:=b;
                b:=r;
             end.
```

После выполнения такого оператора в переменной A будет большее, а в переменной B – меньшее из двух значений, находившихся там ранее.

В качестве выполняемого в условном операторе действия может быть другой условный оператор.

Например:

```
if sqr(x)+sqr(y)>1 then
          if x>y then z:=0
                 else z:=1;
```

При такой форме записи со сдвигом вправо для каждого внутреннего действия, легко понять, к какому из двух слов IF относится слово ELSE. Рассмотрим пример программы с использованием условного оператора. Пусть для двух целых чисел надо определить, являются они четными или нет. Для проверки четности используем условие: остаток от деления на 2 четного числа равен 0.

```
program prim_7;
var  a,b:integer;
begin
   write(‘введите два целых числа’);
   readln(a,b);
   if a mod 2 = 0 then writeln (‘  a - четное ‘)
                  else writeln (‘  a - нечетное ‘);
  if b mod 2 = 0 then writeln (‘  b - четное ‘)
                  else writeln (‘  b -нечетное ‘);
end.
```

Задания для тренировки.

40). Вычислить значение y при заданном значении x:

$$y=\begin{cases}\sin^2 x & при \quad x>o,\\ 1-2\sin(x)^2 & в \quad противном \quad случае.\end{cases}$$

41). Ввести два числа. Напечатать сначала меньшее, затем большее из них.
42). Даны числа x и y. Вычислите число z, равное x+y, если x≤y, и 1 – x + y в противном случае
43). Даны два числа. Выведите первое из них, если оно больше второго, и оба числа, если это не так.
44). Если данное число x меньше нуля, то z присвойте значение большего из двух чисел x и y, иначе z присвойте значение полусуммы этих чисел.
45). *Даны два числа. Меньшее из них замените полусуммой этих чисел, а большее – их произведением.
46). Даны радиус круга и сторона квадрата. У какой фигуры площадь больше?
47). Дано целое число. Определить:
   a. Является ли оно четным;
   b. Оканчивается ли оно цифрой 7;
   c. Делится ли оно на 13.
48). Написать программу нахождения

а) $$F(x)=\begin{cases} \dfrac{x}{x^4+6}, & \text{если } x<3. \end{cases}$$

б) $$F(x)=\begin{cases} 0, & \text{если } x\le 1; \\ \dfrac{1}{x+6}, & \text{если } x>1. \end{cases}$$

в) $$F(x)=\begin{cases} 3x-9, & \text{если } x\le 7; \\ \dfrac{1}{x^2-4}, & \text{если } x>7. \end{cases}$$

г) $$F(x)=\begin{cases} x^2+4x+5, & \text{если } x\le 2; \\ \dfrac{1}{x^2+4x+5}, & \text{если } x>2. \end{cases}$$

§ 10. УСЛОВНЫЙ ОПЕРАТОР. РЕШЕНИЕ ЗАДАЧ

Разберем одно из задания предыдущего § , так как её решение содержит некоторые подводные камни, на которые, как правило, натыкаются начинающие программисты.

Задача:*Даны два числа. Меньшее из них замените полусуммой этих чисел, а большее – их произведением.

На первый взгляд задача решается просто:

```
If a>b  then begin  b:=(a+b)/2; a:=a*b  end
        else  begin  a:=(a+b)/2; b:=a*b  end;
```

но если выполнить эту команду, то ответ получится неверным, почему?

Предположим a=10, B=20. Что же получится в результате выполнения оператора. Так как a<b, то условие, указанное в операторе не выполнено, следовательно, выполнится

else begin a:=(a+b)/2; b:=a*b end;

подставим значения и получим

a:=(a+b)/2=(10+20)/2=15

b:=a*b =15*20=300 вместо 10*20=200 потому, что к этому моменту первоначальное значение a=10 изменилось и стало равно 15. Следовательно надо как то сохранить первоначальные значения. Можно запомнить их в дополнительных переменных

```
x:=a;
y:=b;
If a>b   then begin  b:=(x+y)/2; a:=x*y  end
            else  begin  a:=(x+y)/2; b:=x*y  end;
```

а можно запомнить в дополнительных переменных полусумму и произведение:

```
var  a,b,x,y:real;
begin
      write('введите два  числа');
      readln(a,b);
      x:=a*b;                    {запомним  значение произведения}
      y:=(x+y)/2;     {запомним значение полусуммы}
      if a>b   then begin  b:=y; a:=x  end
            else  begin  a:=y;  b:=x  end;
writeln (' a= ',a:0:2,'   b=',b:0:2);
end.
```

Задание для тренировки.

49). Дано трехзначное число N. Проверить, будет ли сумма его цифр четным числом.
50). Определить, равен ли квадрат заданного трехзначного числа кубу суммы цифр этого числа.
51). Определить, является ли целое число N четным двузначным числом.
52). Определить, является ли треугольник со сторонами a, b, c равносторонним.
53). Определить, является ли треугольник со сторонами a, b, c равнобедренным.
54). Определить, имеется ли среди чисел a, b, c хотя бы одна пара взаимно противоположных чисел.
55). Подсчитать количество отрицательных чисел среди чисел a, b, c.
56). Подсчитать количество положительных чисел среди чисел a, b, c.
57). Подсчитать количество целых чисел среди чисел a, b, c.
58). Определить, делителем каких чисел a, b, c является число k.
59). Даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень — отрицательные.
60). Даны две точки A(x, y) и B(x, y). Составить алгоритм, определяющий, которая из точек находится ближе к началу координат.
61). Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник. Если да, то будет ли он прямоугольным.
62). Составить программу, осуществляющую перевод величин из радианной меры в градусную или наоборот. Программа должна запрашивать, какой перевод нужно осуществить, и выполнять указанное действие.
63). Написать программу нахождения суммы большего и меньшего из 3 чисел.
64). Найти max{min(a, b), min(c, d)}.

§ 11. ЛОГИЧЕСКИЕ ОПЕРАЦИИ И ВЫРАЖЕНИЯ

Для построения сложных условий в Паскале имеются четыре логических операции:

NOT- отрицание (***НЕТ***),
AND- логическое умножение (***И***),
OR- логическое сложение (***ИЛИ***),
XOR- исключающее «ИЛИ».

Если условие выполняется, то говорят, что соответствующее выражение истинно, если не выполняется – выражение ложно.

Приоритеты логических операций: 1) not; 2) and; 3) or; 4) xor.

Результаты логических операций для различных значений операндов приведены в таблице 1, где использованы обозначения: T- true (истина), F – false (ложь). Где A и B результат операции отношения.

Таблице 1

A	B	Not A	A and B	A or B	A xor B
T	T	F	T	T	F
T	F	F	F	T	T
F	F	T	F	F	F
F	T	T	F	T	T

Примеры логических выражений:

a) (0<x) AND (x<=1)
b) (a=0) OR (abs(x)<5)
c) NOT (x=y)

Операции отношений имеют более низкий приоритет, чем логические операции, поэтому их следует заключать в скобки при использовании с логическими операциями.

Из переменных, констант, сравнений, логических операций и скобок можно строить логические выражения.

Рассмотрим следующую задачу: Имеется прямоугольное отверстие со сторонами a и b и кирпич с ребрами x, y, z. Требуется определить пройдет ли кирпич в отверстие.

Решение.

Кирпич имеет три грани, каждую из которых мы можем повернуть на 90 градусов, т.е. для каждой грани надо проверит два случая. Итого шесть. Получаем условие:

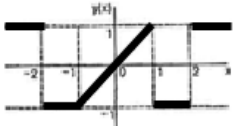
(a>x)and(b>y) or (b>x) and (a>y) or
(a>x) and(z>y) or (z>x) and(a>y) or
(b>x) and(z>y) or (z>x) and(b>y)

Код программы:

```
var  a,b,x,y,z:integer;
                f:boolean;
begin
     write('введите размеры отверстия');
     readln(a,b);
     write('введите размеры кирпича');
     readln(x,y,z);
     if (a>x)and(b>y) or (b>x) and (a>y) or (a>x) and(z>y) or (z>x)
and(a>y) or(b>x) and(z>y) or (z>x) and(b>y)
then  writeln ('кирпич пролезет в отверстие ')
                 else writeln ('кирпич не пролезет в отверстие ');
end.
```

Рассмотрим ещё задачу:

Заданы графики функции y(x), показанные на рис. Записать программы, которые на ввод аргумента выдают значение функции.



Идея решения: Для определенности будем считать, что обозначенные на оси x точки (точки -2, -1, 0, 1 и 2), принадлежат отрезкам прямой слева от точки. Рассмотрим команды с условиями программы решения примера.

```
if x <= -2 or  x > 2 then y = 1
if x > -2 and x <= -1 or x > 1 and x <= 2 then y = -1
if x > -1 and x <= 1 then y = x
```

Задание для тренировки.

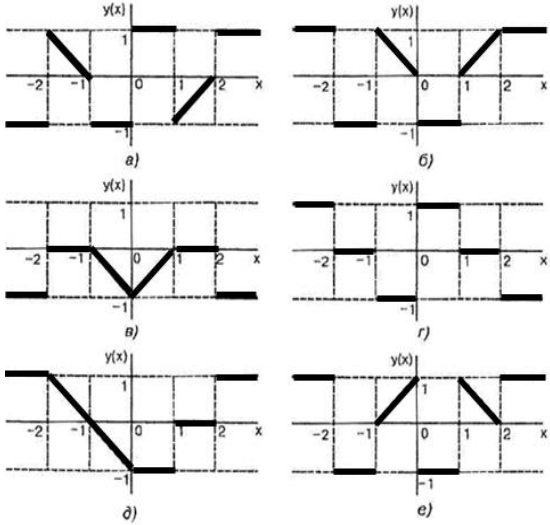
65). Установить, истинны или ложны следующие условия:

(A=0) and not (B=0) or not (a=0) and (B=0) при

a) A=0, B=0

b) A=0, B=1

66). Заданы графики функции y(x), показанные на рис. Записать программы, которые на ввод аргумента выдают значение функции.



67). Определить значение функции

$$y=\begin{cases} x^2, \text{ если } x<0; \\ x, \text{ если } 0\le x\le 1; \\ 1, \text{ если } x>1. \end{cases}$$

68). Проверьте, есть ли среди трех заданных чисел равные.

69). Известны площади круга и квадрата. Определить:

a). Уместится ли круг в квадрате;

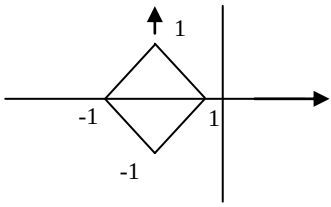
b). Уместится ли квадрат в круге.

70). Выяснить пройдет ли кирпич в круглое отверстие.

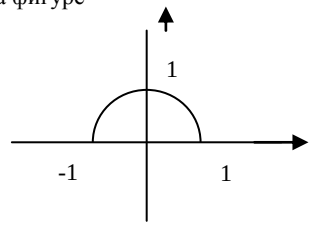
71). Даны три числа a, b, c. удвойте эти числа, если они являются упорядоченными по возрастанию.

72). Определить, принадлежит ли заданная точка фигуре

a)



b)



§ 12. ЦЕЛОЧИСЛЕННАЯ АРИФМЕТИКА И УСЛОВНЫЙ ОПЕРАТОР

Разберём типичные задачи, где необходимо применить условный оператор.

Задача: Дано трехзначное число. Определить:

a) Является ли сумма его цифр двузначным числом;
b) Является ли произведение его цифр трехзначным числом;
c) Больше ли числа А произведение его цифр;
d) Кратна ли пяти сумма его цифр;
e) Кратна ли сумма его цифр числу А.

*Код программы (*оформим задачу одним кодом*).*

```
var  a,x1,x2,x3,x,s,p:word;
begin
   write('введите трехзначное  число');
   readln(x);
   x1:=x div 100;
   x2:=(x div 10) mod 10;
   x3:= x mod 10;
   s:=x1+x2+x3;
   if (s>9) and (s<100) then writeln ('сумма двузначное число') {a}
            else writeln('сумма не двузначное число ');
   p:=x1*x2*x3;
   if (p>99) and (p<1000) then
                    writeln ('произведение трехзначное число') {b}
                          else
                 writeln('произведение не трехзначное число ');
   write('введите число a');
   readln(a);
   if  (p<a) then  writeln (' произведение больше ', a)          {c}
         else writeln('произведение не больше ', a);
   if  s mod 5 = 0 then  writeln (' сумма цифр кратна 5')        {d}
            else writeln('сумма цифр не кратна 5 ');
   if  s mod a = 0 then  writeln (' сумма цифр кратна ',a)       {e}
                 else writeln('сумма цифр не кратна ', a);
end.
```

Задание для тренировки.

73). Дано двузначное число. Определить:
   a. какая из его цифр больше, первая или вторая;
   b. одинаковы ли его цифры.

74). Дано трехзначное число.
   a. Верно ли, что все его цифры одинаковы?
   b. Определить, есть ли среди его цифр одинаковые.

75). Дано четырехзначное число. Определить:
   a. Равна ли сумма двух первых его цифр сумме двух его последних цифр;
   b. Кратна ли трем сумма его цифр;
   c. Кратно ли четырем произведение его цифр;
   d. Кратно ли произведение его цифр числу А.

76). Дано четырехзначное число N. Выяснить:
   a. Является ли число палиндромом?
   b. Верно ли, что все 4 цифры этого числа различны.

77). Дано натуральное число.
   b. Верно ли, что оно заканчивается нечетной цифрой?
   c. Верно ли, что оно заканчивается четной цифрой?
   d. Является ли число А делителем числа В? А наоборот?
78). Трамвайный билет имеет шестизначный номер. Выяснить, является ли билет «счастливым». Билет назовем «счастливым», если сумма первых трех цифр равна сумме последних трех цифр. Примечание. Так как шестизначное число больше 32767(тип Integer), необходимо номер билета определит как тип Longint(до 10 знаков).
79). Ввести три числа. Выбрать и напечатать наибольшее из них, их сумму и произведение.
80). Написать программу, которая требует ввода времени дня и, в зависимости от введенного значения, желает доброго утра, доброго дня, доброго вечера или спокойной ночи.
81). Дано четырехзначное число. Выяснить, является ли оно палиндромом («перевертышем»), т.е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево.
82). В небоскребе N этажей и всего один подъезд; на каждом этаже по 3 квартиры; лифт может останавливаться только на нечетных этажах. Человек садится в лифт и набирает номер нужной ему квартиры М. На какой этаж должен доставить лифт пассажира?
83). Составить программу, которая проверяла бы, не приводит ли суммирование двух целых чисел А и В к переполнению (т. е. к результату большему, чем 32 767). Если будет переполнение, то сообщить об этом, иначе вывести сумму этих чисел.
84). Два прямоугольника, расположенные в первом квадранте, со сторонами, параллельными осям координат, заданы координатами своих левого верхнего и правого нижнего углов. Для первого прямоугольника это точки {x1,y1) и (x2, 0), для второго — (x3, y3), (x4, 0). Составить программу, определяющую, пересекаются ли данные прямоугольники, и вычисляющую площадь общей части, если они пересекаются.

§ 13. ОПЕРАТОРЫ ЦИКЛА В ПАСКАЛЕ

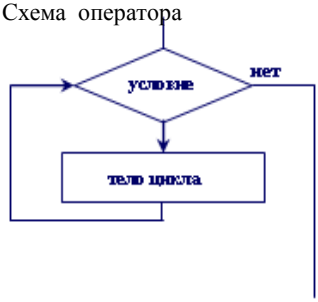
В своей практической деятельности человек постоянно сталкивается с задачами, при решении которых требуется многократно повторять одни и те же действия. Для составления алгоритмов решения таких задач используются команды повторения (циклы).

13.1 Разновидности циклов

Цикл – это замечательное изобретение, которое, в сущности, и делает компьютеры такими ценными. Он позволяет многократно повторить любую часть программы. Цикл не может выполняться вечно, он заканчивается по какому-либо условию. Проверка этого условия может производиться в начале каждого повторяющегося шага, в этом случае цикл называется ПОКА. При проверке условия в конце каждого шага цикл называется ДО. Разновидностью цикла ДО является цикл ПЕРЕСЧЕТ.

Цикл ПОКА

Схема оператора

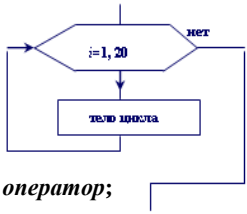


Формат команды:

WHILE ***логическое выражение*** **DO** ***оператор;***

Цикл ПЕРЕСЧЕТ

Схема оператора



Формат команды:

FOR ***переменная := выражение1*** **TO** ***выражение 2*** **DO** ***оператор*****;**

13.2 Оператор цикла WHILE (цикл ПОКА)

Формат оператора:

WHILE ***логическое выражение*** **DO;**

Оператор будет повторятся пока истинно логическое выражение. Перед каждым повторением оператора значение логического выражения вычисляется заново. Если необходимо повторить несколько операторов, их следует объединить в составной оператор, т.е. заключить в операторные скобки begin … end. Этот цикл может не выполниться ни разу, если условие при входе в него оказалось ложным. Таким образом, цикл ПОКА содержит условие повторения цикла.

Пример 1. Программа подсчета суммы S первых 1000 членов гармонического ряда

1+1/2+1/3+1/4+…1/N.

Код программы:

```
var      s:real; n:integer;
begin
        s:=0;    n:=0;
        while n<1000 do
             begin
             n:=n+1;
             s:=s+1/n
             end;
        writeln(s);
end.
```

Пример 2. Вычислить наибольший общий делитель двух натуральных чисел A и B. воспользуемся для этого алгоритмом Евклида: будем уменьшать каждый раз большее из чисел на величину меньшего до тех пор, пока оба числа не станут равны.

Код программы:

```
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;
     writeln('нод=',a);
end.
```

Пример 3. Начав тренировки, лыжник в первый день пробегал 10 км. Каждый следующий день он увеличивал длину пробега на 10% от предыдущего дня. Определить в какой день он пробежит больше 20 км, в какой день суммарный пробег за все дни превысит 100км.

Код программы:

```
var      s:real; n:integer;
begin
    s:=10;      n:=1;
    while s<20 do
              begin
              n:=n+1;
              s:=s*1.1
              end;
    writeln(‘ дневной пробег больше 20 км  на ’,n,’ день‘);
    s:=10;      n:=1;
    while s<100 do
              begin
              n:=n+1;
              s:=s+s*1.1      {накапливаем суммарный пробег}
              end;
    writeln(‘за ’,n,’ пробежит больше 100 км’);
end.
```

Задание для тренировки.

85). Даны целые числа a и b (a>b). Определить:
   a. Результат целочисленного деления a на b, не используя стандартную операцию целочисленного деления;
   b. Остаток от деления a на b не используя стандартную операцию вычисления остатка.

86). Известны оценки по информатике каждого из 20 учеников класса. В начале списка перечислены все пятерки, затем все остальные оценки. Сколько учеников имеют по информатике оценку «5»?. Условный оператор не использовать.

87). Напечатать минимальное число, большее 200, которое нацело делится на 17.

88). Гражданин 1 марта открыл счет в банке, вложив 1000 руб. Через каждый месяц размер вклада увеличивается на 2% от имеющейся суммы. Определить: за какой месяц величина ежемесячного увеличения вклада превысит 30 руб.; через сколько месяцев размер вклада превысит 1200 руб.

89). *В некоторой стране используются денежные купюры достоинством в 1, 2, 4, 8, 16, 32 и 64. дано натуральное число N. Как наименьшим количеством таких денежных купюр можно выплатить сумму N (указать количество каждой из используемых для выплаты купюр)? Предполагается, что имеется достаточно большое количество купюр всех достоинств.

13.3 Цикл ПЕРЕСЧЕТ (прямой)

Формат оператора:

FOR *переменная := выражение 1* TO *выражение 2* DO *оператор;*

Переменная должна быть порядкового типа. Порядковыми называются все простые типы, значения которых можно расположить в возрастающем порядке. Из известных - это: INTEGER, WORD, LONGINT, BYTE, CHAR. Выражение_1 и Выражение_2 должны быть того же типа, что и переменная. Чтобы цикл выполнился хотя бы раз выражение_1 должно быть не больше выражения_2.

Выполнение начинается с вычисления значений выражения_1 и выражения_2. затем переменная получает значение выражения_1 и делается проверка, не превышает ли значение переменной выражения_2. Если не превышает, выполняется оператор стоящий после служебного слова DO. После завершения оператора переменная получает следующее по порядку значение, и все повторяется, начиная с проверки.

Когда значение переменной становится равным выражению_2, оператор выполняется последний раз.

Пример 1. Напечатать ряд из повторяющихся чисел 20 в виде:

20 20 20 20 20 20 20 20 20 20

Код программы:

```
var i: byte;
begin
    for I:=1 to 10 do write(20,' ');
end.
```

Пример 2. Напечатать числа следующим образом:

```
10  10.4
11  11.4
...
12  25.4
```

Код программы (переменная используется не только для управления циклом но и для вывода на экран в качестве результата):

```
var  i: byte;
begin
    for I:=10 to 25 do write(I,' ',I+0.4:0:1);{при сложении целого I и вещественного 0.4 получаем вещественный результат, значит надо выполнить его форматирование при выводе на экран}
end.
```

13.4 Цикл ПЕРЕСЧЕТ (обратный)

Возможен вариант оператора, когда переменная принимает последовательно убывающие значения.

Формат оператора:

FOR ***переменная := выражение_1*** **DOWNTO** ***выражение_2*** **DO** ***оператор;***

В этом случае, чтобы цикл выполнился хотя бы раз, выражение 1 должно быть не меньше выражения2. например:

For c:='z' downto 'a' do writeln(c);

Тренировочные задания.

90). Напечатать столбиком:
- а) все целые числа от 20 до 35;
- б) квадраты всех целых чисел от 10 до b (значение d вводится с клавиатуры; b>=10);
- в) третьи степени всех целых чисел от a до 50 (значение a вводится с клавиатуры; a<=50);
- г) все целые числа от a до b (значения a и b вводятся с клавиатуры; d>=a).

91). Напечатать числа следующим образом

```
25.5  24.8
26.5  25.8
  ...
35.5  34.8
```

92). Распечатать в столбик таблицу умножения на 7.

93). Вывести столбиком следующие числа: 2,1 2,2 2,3 . . . , 2,8

94). Вывести столбиком следующие числа: 2,2 2,4 2,6 . . . , 4,0 4,2

95). Вывести столбиком следующие числа: 4,4 4,6 4,8 . . . , 6,2 6,4

96). Найти корень уравнения $y=2x^2-1$ на отрезке [0;1] с точностью до 0.1. Результат представьте в виде таблицы.

97). Найти значения функции x^3-5x+3 на отрезке [-2;5] с шагом 1. Результат представьте в виде таблицы.

98). Найти сумму членов ряда, предварительно определив формулу общего члена.

a) S = 3/2 + 4/5 + 5/8 + 6/11 + ….. + 22/59

b) S = 0 + 1/5 + 2/7 + 3/9 +… + 16/35

§ 14. ПОСЛЕДОВАТЕЛЬНАЯ ДЕТАЛИЗАЦИЯ АЛГОРИТМА

Алгоритмы для обработки последовательностей обычно относятся к одному из двух типов: поиск; проверка условий.

Для последовательностей характерно, что в каждый момент времени нам доступен только один элемент последовательности. Поэтому все алгоритмы строятся с учетом однократного последовательного просмотра. Рассмотрим несколько программ. В каждой из них одновременно рассматривается только очередной член последовательности. Алгоритмы для решения таких задач называются алгоритмами с линейным поиском.

Задача: Вводится последовательность из N целых чисел. Найти сумму всех отрицательных чисел.

Код программы:

```
var     i,n,x,sum:integer;
begin
    write('введите длину последовательности n=');readln(n);
    sum:=0;
    for i:=1 to n  do begin
                write('введите x=');
                readln(x);
                if x<0 then sum:=sum+x
                end;
    if sum=0 then writeln('отрицательных чисел нет')
          else writeln('сумма отрицательных чисел =', sum);
end.
```

Тренировочные задания.

99). Вводится последовательность из n произвольных чисел. Определить, сколько раз последовательность меняет знак.

100). Вводится последовательность чисел, 0-конец последовательности. Определить, содержит ли последовательность хотя бы два равных соседних числа. Найти наименьшее число.

101). Вводится последовательность из N целых чисел. Найти наибольшее из всех отрицательных чисел. Найти, сколько в ней нулей.

102). Сформировать последовательность из 22 элементов, значения которых лежат в пределах от -7 до 31 и определить максимальное значение элемента.

103). В 1985 году урожай ячменя составил 20 ц с гектара. В среднем каждые 2 года за счет применения передовых агротехнических приемов урожай увеличивается на 5%. Определить, через сколько лет урожайность достигнет 25ц с га.

104). Найдите наименьшее трехзначное число, сумма кубов цифр которого равна 730.

105). Сформировать последовательность из 14 элементов, значения которых лежат в пределах от -13 до 19, и определить произведение элементов с нечетными номерами.

106). Найти значение функции $y=x^2+1$ на отрезке [5;10] с шагом 1. Результат представьте в виде таблицы.

107). Составить программу вычисления площади фигуры ограниченной графиками функций F(x)=cos x ,x=-1.5,x=1.5,y=0 методом левых прямоугольников.

108). Составить программу вычисления площади фигуры ограниченной графиками функций F(x)=sin x , x=0, x=3 ,y=0 методом правых прямоугольников.

ЛИТЕРАТУРА

1. PascalABC.NET. Обучение современному программированию. [Электронный ресурс] URL: http://pascalabc.net (дата обращения 10.01.2013)
2. ИНФОРМАТИКА. Задачник практикум.1 том. Л.А. Залогова и другие. Под ред. И.Г.Семакин. Москва БИНОМ. Лаборатория знаний. 2006
3. М.Е. Фиошин, А.А. Рессин, С.М.Юнусов. Информатика и ИКТ. 10-11 кл. Профильный уровень. В 1 ч. Ч.2.-М.Дрофа, 2008
4. Окулов С.М. Основы программирования. – М.: ЮНИМЕДИАСТАЙЛ, 2006. – 424 с.
5. Система программирования Pascal ABC. [Электронный ресурс] URL: http://sunschool.math.sfedu.ru/pabc/(дата обращения 10.01.2013)
6. Электронный задачник по программированию. © М. Э. Абрамян, 1998–2012 [Электронный ресурс] URL: http://ptaskbook.com (дата обращения 10.01.2013)