Использование переменных. Объявление переменной (тип, имя, значение). Локальные и глобальные переменные. Массивы

Канд. пед. наук, доцент И.В. Попова

Магнитогорск, 2012

Данные

Исходные (входные) — это данные, известные перед выполнением задачи из условия.

промежуточные – данные, которые не являются ни аргументом, ни результатом алгоритма, а используются только для обозначения вычисляемого промежуточного значения

Выходные— результат решения задачи.

Величина

— это элемент данных с точки зрения их семантического (смыслового) содержания или обработки.

Смысловое (семантическое) разбиение данных производится во время постановки задачи и разработки алгоритма ее решения (входные, выходные и промежуточные).

имя (идентификатор) — обозначение величины и её место в памяти

значение — динамическая характеристика, может меняться многократно в ходе исполнения алгоритма

тип — множество допустимых значений и множество применимых операций к ней

Постоянная и переменная

Типы данных (Pascal)

Целые типы

- определяют константы, переменные и функции, значения которых реализуются множеством целых чисел, допустимых в данной ЭВМ.

тип	диапазон значений	требуемая память
Shortint	-128 .. 127	1 байт
Integer	-32768 .. 32767	2 байта
Longint	-2147483648 .. 2147483647	4 байта
Byte	0 .. 255	1 байт
Word	0 .. 65535	2 байта

Арифметические операции над целыми

Операция	Символ	Примечания
сложение	+	Результат арифметической операции над целыми операндами есть величина целого типа. Результат выполнения операции деления целых величин есть целая часть частного.
вычитание	-	
умножение	*	
Получение целой части от деления	div	17 div 2 = 8, 3 div 5 = 0.
получение остатка от деления	mod	17 mod 2 = 1, 3 mod 5 = 3

Операции отношения

Операции отношения, примененные к целым операндам, дают результат логического типа TRUE или FALSE (истина или ложь).

В языке ПАСКАЛЬ имеются следующие операции отношения:

- равенство =,
- неравенство <>,
- больше или равно >=,
- меньше или равно <=,больше >,
- меньше < .

Стандартные функции

Функция	Значение
Abs(X), Sqr(X), Succ(X), Pred(X),	абсолютное значение X, X в квадрате, X+1, X-1.
Sin(X), Cos(X), ArcTan(X), Ln(X), Exp(X), Sqrt(X).	синус, косинус и арктангенс угла, заданного в радианах, логарифм натуральный, экспоненту и корень квадратный соответственно.
Odd(X)	значение истина, если аргумент нечетный, и значение ложь, если аргумент четный
Inc(X)	X:=X+1
Inc(X,N)	X:=X+N
Dec(X)	X:=X-1
Dec(X,N)	X:=X-N

Действительные типы данных

Определяют те данные, которые реализуются подмножеством действительных чисел, допустимых в данной ЭВМ.

Тип	Диапазон значений	Количество цифр мантиссы	Требуемая память, байт
Real	2.9e-39 .. 1.7e+38	11	6
Single	1.5e-45 .. 3.4e+38	7	4
Double	5.0e-324 .. 1.7e+308	15	8
Extended	3.4e-4932 .. 1.1e+4932	19	10
Comp	-9.2e+18 .. 9.2e+18	19	8

Операции над действительными числами

- Арифметические : сложение + , вычитание - , умножение * , деление / .
- Все операции отношения, дающие булевский результат.
- Функции, дающие действительный результат: Abs(X), Sqr(X), Sin(X), ArcTan(X), Ln(X),Exp(X), Sqrt(X), Frac(X), Int(X), Pi, Cos(X)
  - Frac(X) возвращает дробную часть X,
  - Int(X) – целую часть X.
- Trunc(X) целая часть действительного аргумента путем отсечения дробной части
- Round(X) – округление до ближайшего целого

Булевские операнды

- Not
- And
- Or
- Xor

False< True

Boolean – 1 байт
ByteBool, - 1 байт
WordBool - 2 байта
LongBool - 4 байта.

Символьный тип

- определяет упорядоченную совокупность символов, допустимых в данной ЭВМ.
- Значение символьной переменной или константы - это один символ из допустимого набора.
- Способы задания:
  - как один символ, заключенный в апострофы, например: 'A' 'a' 'Ю' 'ю';
  - с помощью конструкции вида #K, где K - код соответствующего символа, при этом значение K должно находиться в пределах 0..255;
  - с помощью конструкции вида ^C, где C - код соответствующего управляющего символа, при этом значение C должно быть на 64 больше кода управляющего символа.

Операции над величинами символьного типа

- все операции отношения.
- Ord(C) - определяет порядковый номер символа С в наборе символов;
- Chr(K) – определяет по порядковому номеру К символ, стоящий на К-ом месте в наборе символов;
- Pred(C) - определяет предыдущий символ Pred('F') = 'E' ;
- Succ(C) – определяет последующий символ Succ('Y') = 'Z'
- UpCase(C) – переводит буквы в верхний регистр

Переменная

- небольшая область в оперативной памяти компьютера, куда во время работы программы можно занести и хранить в закодированном виде некоторое значение (целое или действительное число либо символьный текст), которое в дальнейшем можно использовать или изменять.

Место переменной в общем объеме ОЗУ определяется адресом - ее порядковым номером в 16-ричной системе счисления. Например, в машинных кодах команда имеет вид:

K A B C,

где К - код операции; A, B, C - адреса ячеек памяти.

Имя переменной - это название ячейки ОЗУ, используемое в программе на алгоритмическом языке вместо указания адреса.

Виды переменных

- **Область видимости переменной** - место в программе, в котором доступно значение переменной.
  - **Статическая** переменная создаётся в момент запуска программы
  - **Динамическая** переменная создаётся в процессе её выполнения.
  - **Локальные** переменные доступны конкретной подпрограмме
  - **Глобальные** переменные доступны всей программе.

Простые переменные не имеют внутренней структуры, доступной для адресации.

Операции ввода данных (Pascal)

Процедура, которая в режиме диалога с клавиатуры присваивает значение для переменной величины, называется процедурой **ввода**

Read(список переменных)

```
Var
 A : Real; B : Integer; C : Char;
Begin
 Read(A, B, C)
End.
```

Читается: «Ввести вещественную A, целую B и символьную C».

ReadLn(список переменных)

```
ReadLn(A, B, C);
ReadLn(X);
ReadLn(LogPer);
```

ReadLn реагирует на конец строки, и в случае его обнаружения происходит сразу переход к следующей строке

В списке ввода значения разделяются между собой пробелом

Операции вывода данных (Pascal)

Процедура, которая выводит содержимое переменных на экран, называется процедурой **вывода** на экран

Write (список констант и/или переменных, разделенных запятой)

```
Write(A, B, C);
WriteLn('Корнем уравнения является ', X);
WriteLn(LogPer);
```

вывод значений выражений, приведенных в его списке, на текущую строку до ее заполнения

Для десятичного представления значения R применяется оператор с форматами вида WriteLn(R : N : M). N – это количество цифр до запятой, M – это количество цифр после запятой.

WriteLn (список констант и/или переменных, разделенных запятой)

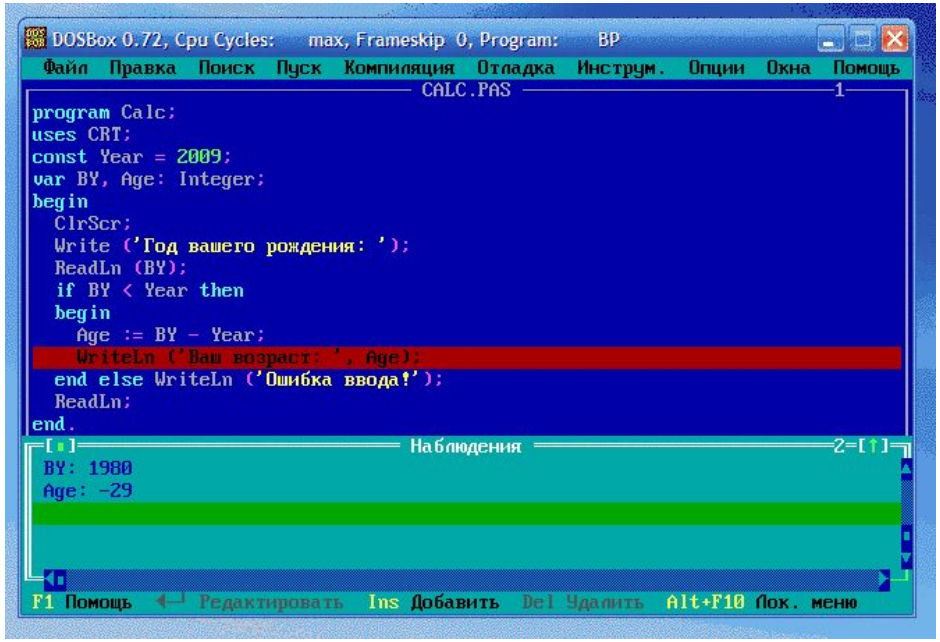
реализуется вывод значений выражений на одну строку дисплея и переход к началу следующей строки

Процедура вывода с форматом для целого типа имеет вид:

WriteLn(A : N, B : M, C : L);

N, M, L — выражения целого типа, задающие ширину поля вывода значений.

Общая структура программы на Pascal



Program имя программы; {заголовок}
Const {объявление констант}
Константа1 = значение;
Константа2 = значение;
…
КонстантаN = значение;
Type …; {объявление типов}
Var {описание переменных}
Список Переменных1 : Тип;
Список Переменных2 : Тип;
…
Список ПеременныхN : Тип;
Function {описание функции}
Procedure {описание процедуры}
Основная программа
Begin
{раздел операторов}
End.

Присваивание в Паскале

ПЕРЕМЕННАЯ := ВЫРАЖЕНИЕ;

```
X := (Y + Z) / (2 + Z * 10) - 1/3;
LogPer := (A > B) And (C <= D);
```

\- + *

Операнд 1	Операнд 2	Результат
Integer	Integer	Integer
Integer	Real	Real
Real	Integer	Real
Real	Real	Real

Деление – только Real

Порядок действий при вычислении значения выражения:

1) вычисляются значения в скобках;
2) вычисляются значения функций;
3) выполняется унарные операции (унарный минус — смена знака);
4) выполняются операции умножения и деления (в том числе целочисленного деления и нахождения остатка от деления);
5) выполняются операции сложения и вычитания.

Порядок действий при вычислении значения логического выражения:

1) вычисляются значения в скобках;
2) вычисляются значения функций;
3) выполняется унарные операции (операция NOT);
4) выполняется операция AND;
5) выполняются операции OR, XOR;

Условный оператор IF

- Может принимать одну из следующих форм: **IF *условие* THEN *оператор* ;** (неполная форма)
- Условие задаётся переменной и её соотношением с некоторой константой или значением выражения.

- Если условие выполняется , то будет выполнен указанный оператор. **IF *условие* THEN *оператор1* ELSE *оператор2*;** (полная форма)
- Добавление к конструкции служебного слова **ELSE** позволяет выполнить определенные действия в случае, когда условие не выполняются.

```
If a mod 2 = 1
  then WriteLn('a-чётное число');
  else WriteLn('a-нечётное число');
```

Цикл с предусловием

While <условие> Do <оператор>;

"Пока истинно условие, выполнять оператор".

```
While <условие> Do
  Begin
   <оператор #1>;
   <оператор #2>;
   <оператор #3>;
    . . .
  End;
```

```
While   I <= n Do
  Begin
     WriteLn (i,’,’);
     i:=2*I;
   End;
```

Цикл с постдусловием

Repeat <оператор> Until <условие>;

Repeat
<оператор #1>;
<оператор #2>;
<оператор #3>;
. . .
Until <условие>;

Repeat
    WriteLn (I, ‘,’);
    i:=2*I;
Until i>n;

не требуется использование составного оператора, потому, что сами слова Repeat и Until являются операторными скобками

"Выполнять оператор #1, оператор #2. : до выполнения условия".

Цикл с параметром

- **Форматы:**
  - **For** <И.П.>:=<Н.З.> **To** <К.З.> **Do** <оператор>;
    For <И.П.>:=<Н.З.> **Downto** <К.З.> **Do** <оператор>;
    И.П. - имя переменной-параметра,
    Н.З. - его начальное значение,
    К.З. - соответственно конечное значение параметра.

Выполняется по следующему алгоритму:

1) переменной-параметру присваивается начальное значение;
2) выполняется тело цикла;
3) переменная-параметр автоматически увеличивается на 1 (в первом случае формата);
4) если параметр превышает конечное значение, то происходит выход из цикла, иначе - переход к пункту 2.

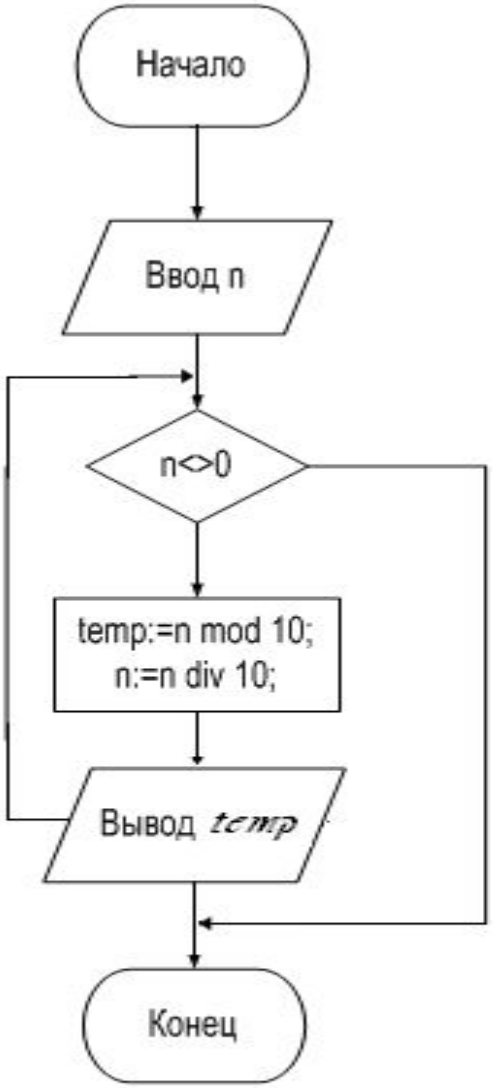
Пример. Найти сумму квадратов всех натуральных чисел от 1 до 100

```
Program Ex1;
Var
A : Integer;
S : Longint;
Begin
        A:=1; S:=0;
        While A<=100 Do
        Begin
                S:=S+A*A;
                A:=A+1
        End;
        Writeln(S)
End.
```

```
Program Ex2;
Var
A : Integer;
S : Longint;
Begin
        A:=1; S:=0;
        Repeat
                S:=S+A*A;
                A:=A+1
        Until A>100;
        Writeln(S)
End.
```

```
Program Ex3;
Var
A : Integer;
S : Longint;
Begin
        S:=0;
        For A:=1 To 100 Do S:=S+A*A;
        Writeln(S)
End.
```

Алгоритм 1. Выделение цифр в числе.



Введем обозначения объектов алгоритма:

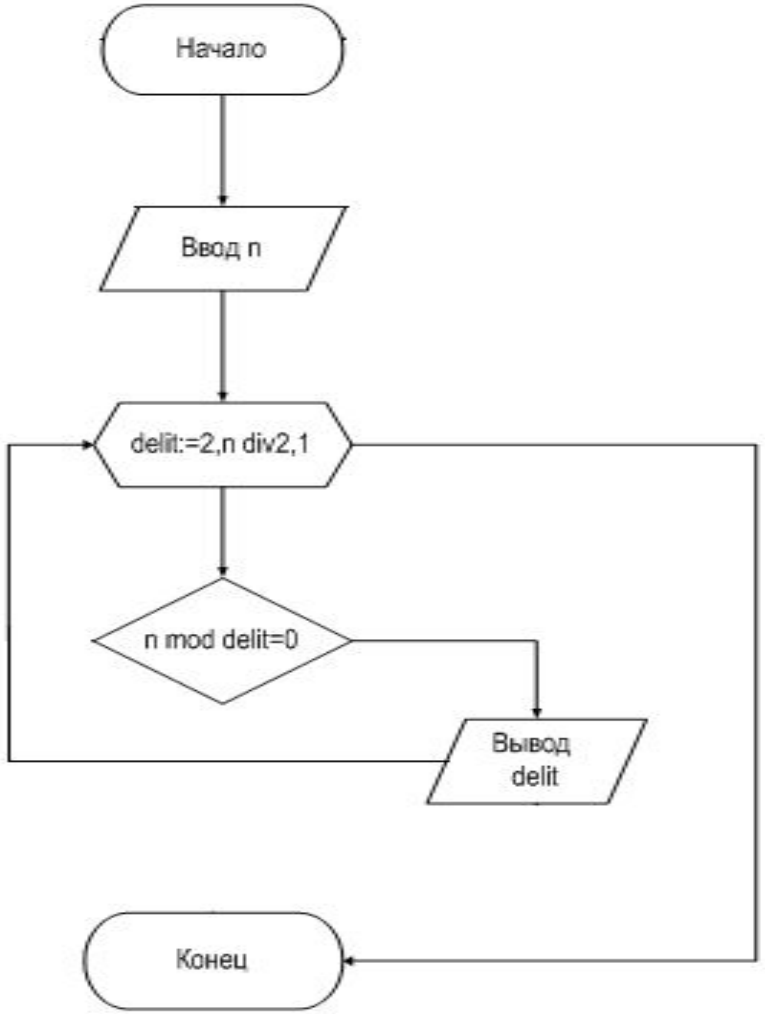
num – целое число, подлежащее обработке;

temp – переменная для хранения цифры числа.

Текст алгоритма

```
Writeln ('Введите целое число');
Readln (num);
Write ('цифры числа:');
While num<>0 do
   begin
     temp:=num mod 10;{выделение цифры
   из числа}
     num:=num div 10; {уменьшение числа на
   один разряд}
    write (temp,' ');
 end;
```

Алгоритм 2. Делители числа.



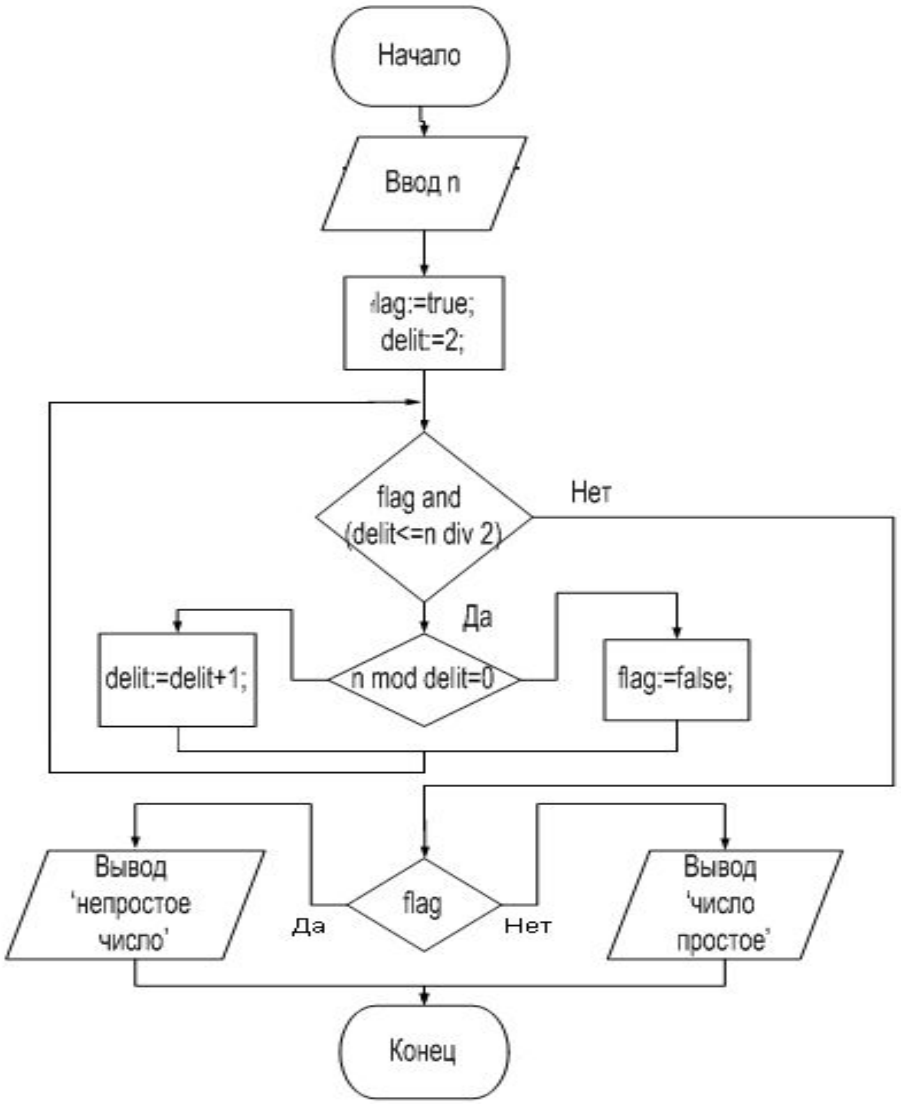
Введем обозначения объектов алгоритма:

delit – делитель числа num.

Текст алгоритма

```
Writeln (‘Введите целое число’);
Readln (num);
For delit:=2 to num div 2 do
    If num mod delit=0 then write(delit,’ ‘);
```

Алгоритм 3. Проверка, является ли число простым



Введем обозначения :

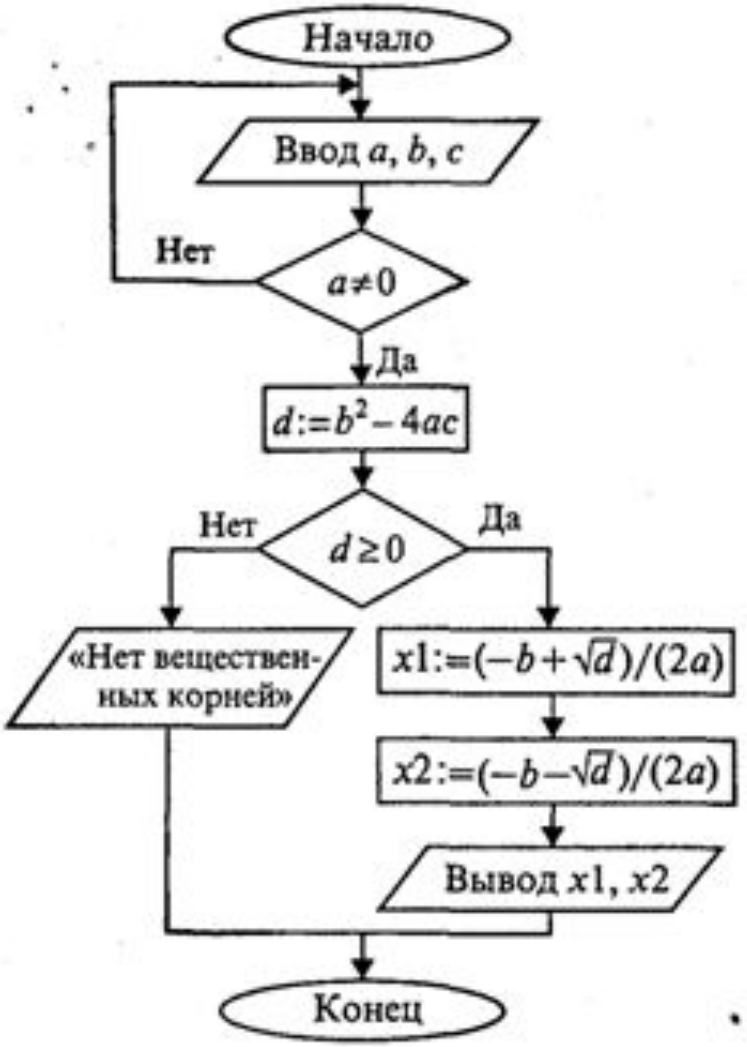
delit – делители числа num;

flag – переменная логического типа, которая используется для долгосрочного выхода из цикла.

Текст алгоритма

```
Writeln ('Введите целое число');
Readln(num);
flag:=true;{предположим, что число простое}
delit:=2;
while flag and (delit <= num div 2) do
    if num mod delit=0 then flag:=false
    {нашелся делитель, отличный от 1 и num}
                        else delit:=delit+1;
if flag then writeln ('число простое')
        else writeln ('число составное');
```

Алгоритмы решения задачи 4




```
алг квадратное
уравнение
вещ a,b,c,d,x1,x2
нач
  повторять
      ввод a,b,c
  до a≠0
  d:=b²-4ac
  если d≥0
  то x1:=(-b+√d)/(2a)
     x2:=(-b-√d)/(2a)
     вывод x1,x2
  иначе
     вывод «Нет
     вещественных
      корней»
  кв
кон
```

Оператор выбора

позволяет выбрать одно из нескольких возможных продолжений программы. Параметром, по которому осуществляется выбор, служит ключ выбора – выражение любого порядкового типа.

Case <ключ_выбора> of
<список_выбора>
[else <оператор_иначе>] end

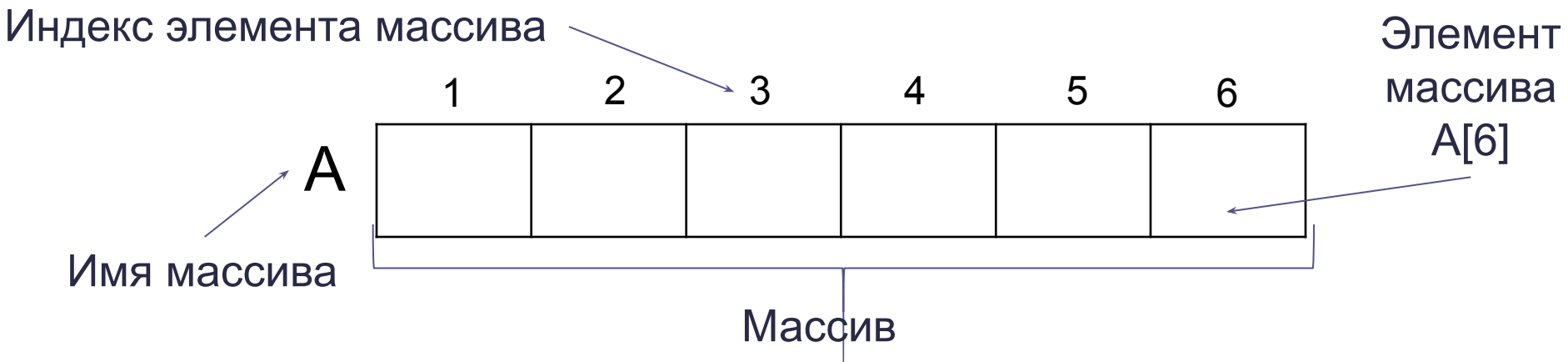
- <ключ_выбора> - выражение порядкового типа;
- <список_выбора> - одна или более конструкций вида:
  - <константа_выбора>: <оператор>;
- <константа_выбора> - константа того же типа, что и выражение
  - <ключ_выбора>;

<операторы> - произвольные операторы Паскаля.

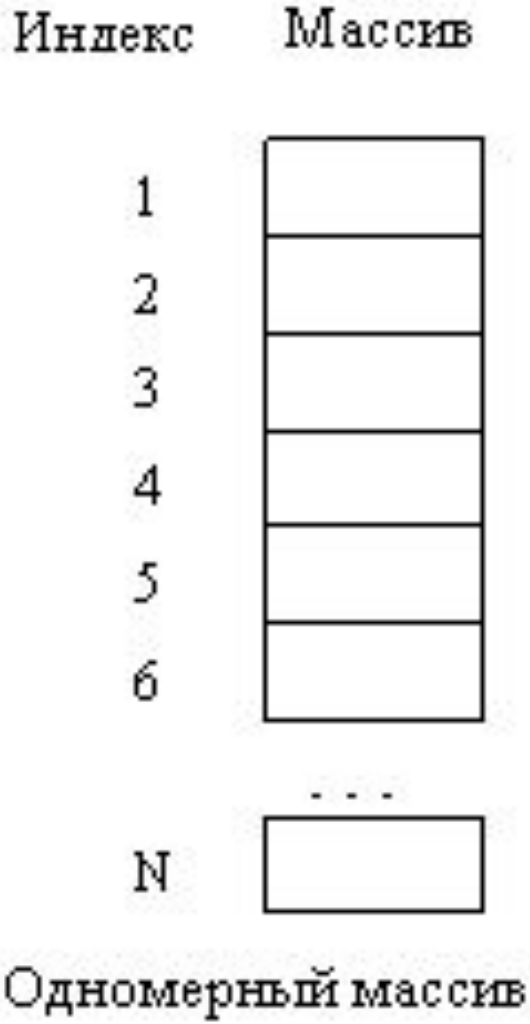
```
Case n of
  1: writeln('понедельник ');
  2: writeln('вторник ');
  3: writeln('среда ');
  4: writeln('четверг ');
  5: writeln('пятница ');
  6: writeln('суббота ');
  7: writeln('воскресенье');
  else writeln('дня недели с номером', n,'нет');
end;
```

Массив

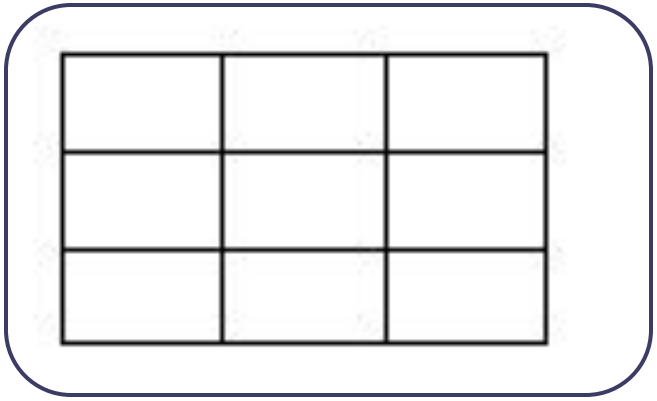
- **— это пронумерованная последовательность величин одинакового типа, обозначаемая одним именем.**
- Элементы массива располагаются в последовательных ячейках памяти, обозначаются именем массива и индексом.
- Каждое из значений, составляющих массив, называется его компонентой (или элементом массива).
- Массиву присваивается имя, посредством которого можно ссылаться как на массив данных в целом, так и на любую из его компонент



Виды массива



Одномерный массив



Двумерный массив

Многомерный массив

Описание массива

Имя_массива: array [начальное _значение _индекса..конечное _значение_индекса] Of базовый тип;

Var

B : Array [1..5] Of Real, {массив B, состоящий из 5 элементов вещественного типа**}**

R : Array [1..34] Of Char; {символьный массив R, состоящий из 34 элементов**}**

Размер массива?

Индекс в обозначении компонент массивов может быть:

- константой;
- переменной;
- выражением порядкового типа

Базовый тип элементов массива может быть любым, за исключением файлового

Заполнение массива с помощью оператора присваивания

Заполнить одномерный массив элементами, отвечающими следующему соотношению: **a1=1; a2=1; ai=ai-2+ai-1 (i = 3, 4, …, n).**

```
A[1]:= 1;
A[2]:= 1;
FOR I:= 3 TO N DO
    A[I]:= A[I - 1] + A[I - 2];
```

Заполнить одномерный массив с помощью датчика случайных чисел таким образом, чтобы все его элементы были различны

```
Program Create;
Type Mas = Array[1..100] Of Integer;
Var A : Mas; I, J, N : Byte; Log : Boolean;
Begin
  randomize;
 A[1] := -32768 + random(65535);
  For I:= 2 To N Do
  Begin
    Log:= True;
    Repeat
      A[i]:= -32768 + random(65535); J := 1;
      While Log and (j <= i - 1) Do
      begin Log := a[i] <> a[j]; j := j + 1 End
    Until Log
  End;
End.
```

Заполнение массива с клавиатуры

- используется обычно тогда, когда между элементами не наблюдается никакой зависимости

```
Program Vvod;
 Var N, I : Integer;
     A : Array [1..20] Of Integer;
Begin
  Write('Введите количество элементов массива ');
  ReadLn(N);
  FOR I:= 1 TO N DO
  Begin
   Write('Введите A[', I, '] ');
    ReadLn(A[I])
  End
End.
```

Двумерный массив(таблица) -

способ организации данных, при котором каждый элемент определяется номером строки и номером столбца, на пересечении которых он расположен.

имя массива: **Array** [начальное значение индекса..конечное значение индекса (число строк), начальное значение индекса.. конечное значение индекса (число столбцов)] **Of** базовый тип;

```
Var B : Array [1..5, 1..6] Of Real; {массив вещественных чисел B, состоящий из 5 строк и 6 столбцов }
    R : Array [1..34, 1..5] Of Char; {символьный массив R, состоящий из 34 строк и 5 столбцов}
```

n-мерный массив - это одномерный массив, элементами которого являются (n-1)-мерные массивы

Описание матрицы на языке Паскаль

1 способ

<имя матрицы>: array
<количество строк> of array
<количество столбцов>
of <тип переменной>;

Var A : Array [1..20] Of Array [1..30] Of Integer;

2 способ

<имя матрицы>: array
<количество строк
>,<количество столбцов> of
<тип переменной>]

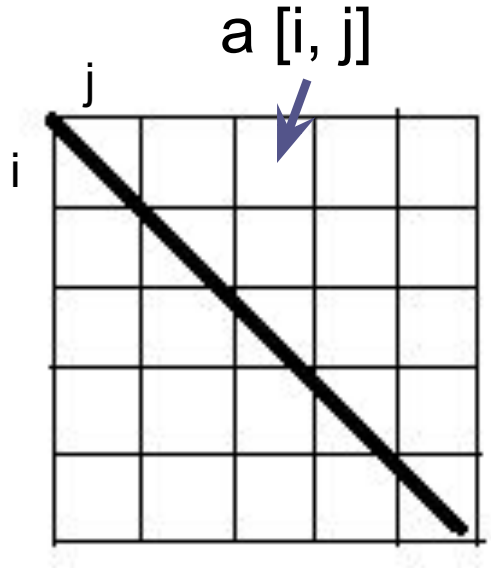
Var A : Array [1..20,1..30] Of Integer

двумерный массив, соответствующий таблице, состоящей из 20 строк и 30 столбцов

Соотношение индексов в квадратной матрице[1]

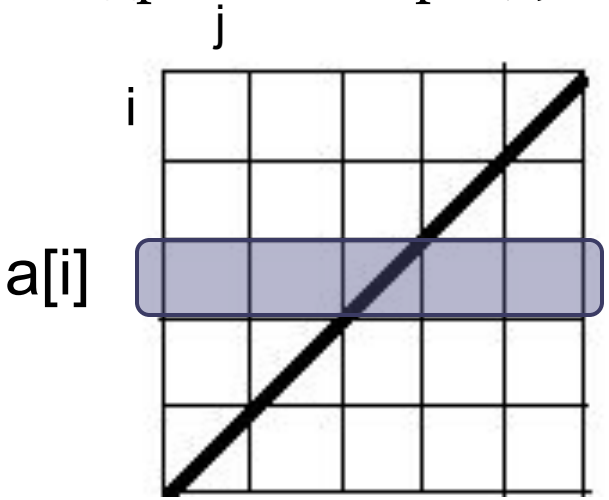
Главная диагональ

- I=J : элементы матрицы расположены на главной диагонали



Побочная диагональ

- J=N-I+1 : элементы матрицы расположены на побочной диагонали (N - количество строк или столбцов в квадратной матрице)



[1]число строк матрицы равно числу столбцов

Ввод-вывод двумерного массива

Ввод

```
for i:= 1 to N do
 begin
   for j:= 1 to M do
      read (A[i,j]:4);
   writeln;
 end;
```

Вывод

```
for i:= 1 to N do
begin
   for j:= 1 to M do
      write (A[i,j]:4);
   writeln; {переводит строку,
   для того, чтобы массив
   выводился в виде таблицы}
end;
```

В результате выполнения программы:

```
Program Vvod2;
 Var I, J : Integer;
    A : Array [1..20, 1..20] Of Integer;
 Begin
  FOR I := 1 TO 3 DO    FOR J := 1 TO 2 DO A[I, J] := 456 + I
 End.
```

элементы массива примут значения A[1, 1] = 457; A[1, 2] = 457; A[2, 1] = 458; A[2, 2] = 458; A[3, 1] = 459; A[3, 2] = 459.

Задачи для самостоятельного решения

1) Дана масса в килограммах. Найти число полных тонн в нем.
2) С некоторого момента прошло n число дней. Сколько полных недель прошло за этот период?
3) Дан прямоугольник со сторонами 543x130. Сколько квадратов со стороной 130 мм можно отрезать от него?
4) С начала 1990 года по некоторый день прошло n месяцев и 2 дня. Присвоить целочисленной величине x значение 1, 2, 3…11,12 в зависимости от того, каким месяцем является месяц этого дня. Например, при n=3 значение x равно 4.

Задачи для самостоятельного решения

1. Рассчитать значение y при заданном значении x:

   Sin^2x при $x>0$,

   $y=1+2\ Sin^2x$ в противном случае.

2. Даны два различных вещественных числа. Определить:

   а) какое из них больше;

   б) какое из них меньше.

3. Найти корни для квадратного уравнения: $ax^2+bx+c=0$.

Задачи для самостоятельного решения

1. Дано трехзначное число. Найти полученное число при прочтении его справа налево.
2. Дано трехзначное число. В нем зачеркнули последнюю цифру и приписали ее в начале. Вывести новое число.
3. Дано трехзначное число. Найти число, полученное при перестановке первой и второй цифр числа.
4. Дано четырехзначное число. Найти:
   а) сумму его цифр;
   б) произведение его цифр.

Задачи для самостоятельного решения

1. Из трехзначного числа x вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа x, то получилось число n. Найти число n. По заданному n найти число x (значение n вводится с клавиатуры, 10<=n<=999 и при этом число десятков в n не равно нулю).
2. В трехзначном числе x зачеркнули первую цифру. Когда оставшееся число умножили на 10, а произведение сложили с первой цифрой числа x, то получилось число 564. Найти число x.

Условный оператор. Задачи для самостоятельного решения

1. Определить, является ли число a делителем числа b?
2. Дано натуральное число. Определить:
   а) является ли оно четным;
   б) оканчивается ли оно цифрой 7.
3. Дано двузначное число. Определить:
   а) какая из его цифр больше;
   б) одинаковы ли его цифры.
4. Дано двузначное число. Определить:
   а) является ли сумма его цифр двузначным числом;
   б) больше ли числа a сумма его цифр.
5. Дано трехзначное число. Выяснить, является ли число полиндромом, т.е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево.
6. Дано трехзначное число. Определить, равен ли квадрат этого числа сумме кубов его цифр.

Условный оператор. Задачи для самостоятельного решения

7. Дано трехзначное число. Определить:
   а) является ли сумма его цифр двузначным числом;
   б) является ли произведение его цифр трехзначным числом;
   в) больше ли числа а произведение его цифр;
   г) кратна ли 5 сумма его цифр;
   д) кратна ли сумма его цифр числу а.
8. Дано трехзначное число.
   а) Верно ли, что все его цифры одинаковые?
   б) Определить, есть ли среди его цифр одинаковые.
9. Дано четырехзначное число. Определить:
   а) кратна ли трем сумма его цифр;
   б) кратно ли 4 произведение его цифр;
   в) кратно ли произведение его цифр числу а.
10. Дано натуральное число.
   а) Верно ли, что оно заканчивается четной цифрой?
   б) Верно ли, что оно заканчивается нечетной цифрой?

Оператор выбора. Задачи для самостоятельного решения

1. Составить программу, которая в зависимости от порядкового номера дня недели (1,2..,7) выводит на экран его название (понедельник, вторник,..., воскресенье).
2. Составить программу, которая в зависимости от порядкового номера дня месяца (1, 2,...,12) выводит на экран его название (январь, февраль,..., декабрь).
3. Составить программу, которая в зависимости от порядкового номера дня месяца (1, 2,...,12) выводит на экран время года, к которому относится этот месяц.
4. Мастям игральных карт условно присвоены следующие порядковые номера: масти «пики» - 1, масти «трефы» - 2, масти «бубны» - 3, масти «червы» - 4. По заданному номеру масти определить название соответствующей масти.