

*Если бы мы писали программы с детства, то с годами, возможно, научились бы их читать.*

**Тема:**

*«Понятие переменной»*

*«Типы данных»*

# Константы

- ◎ **Целочисленная константа** – целое число (определяется числом, записанным либо в десятичной, либо в шестнадцатеричной системе счисления. Эти числа должны использоваться без десятичной точки.

**Пример:**

```
const a=8;
```

# Константы

- ◎ **Вещественная константа** - дробное число (может быть определена числом, записанным в десятичной системе счисления с применением десятичной точки).

**Пример:**

```
const b=19.47;
```

# Константы

- ◎ **Символьная константа** — некоторый символ, заключенный в апострофы.

**Пример:**

```
const c='p';
```

# Константы

- ◎ **Строковая константа** — последовательность любых символов, которая заключена в апострофы.

**Пример:**

```
const d='программа';
```

- ◎ **Типизированная константа** — это инициализированные переменные (каждой такой константе ставится в соответствие имя, тип и начальное значение). Они могут быть использованы в программе наравне с обычными переменными.

**Примеры:**

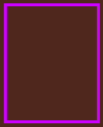
```
god: integer = 2016;
```

```
simvol: char = '?';
```

```
tip: real = 12.47;
```

# Переменная

- Под переменной в языках программирования понимают программный объект (число, слово, часть слова, несколько слов, символы), имеющий имя и значение, которое может быть получено и изменено программой.
- Если "заглянуть" в компьютер, то переменную можно определить так:
- Переменная - это имя физического участка в памяти, в котором в каждый момент времени может быть только одно значение.
- Переменная - это область в оперативной памяти компьютера для хранения какой-либо информации.
- Само название "переменная" подразумевает, что содержимое этого участка может изменяться.



У каждой переменной есть имя, тип и текущее значение.





# Имя переменной

- В качестве имен переменных могут быть латинские буквы с цифрами. Причем может быть не одна буква, а несколько.
- В Паскале не имеет значение в обозначении переменных маленькими или большими (строчными или прописными) буквами они написаны.
- Имя переменной должно удовлетворять следующим требованиям:
  - 1) *всегда начинаться с буквы, после которой могут другие буквы или цифры;*
  - 2) *в идентификаторе (имени) не должно быть пробелов, запятых или других непредусмотренных знаков;*
  - 3) *нельзя использовать в качестве имен переменных слова, которые являются служебными или операторами.*

# Типы переменных (данных)

- Понятие типа данных является ключевым в языке Паскаль. Тип данных *характеризует внутреннее представление, множество допустимых значений для этих данных, а также совокупность операций над ними.*  
Среди типов данных различают стандартные (предопределенные разработчиками языка) и пользовательские (определяемые программистом в своей программе).

Мы будем рассматривать следующие стандартные типы:

- целые числа ( $-\infty \dots -2, -1, 0, 1, 2, \dots +\infty$ )
- вещественные числа (действительные числа: все)
- ЛОГИЧЕСКИЙ ТИП (ложь, истина)
- СИМВОЛЬНЫЙ (символы, слова или текст)
- СТРОКОВЫЙ (символы, слова или текст). Пример: сравнение строк
- Программист может описать свой тип на основе этих базовых в разделе описания типов.

В языке Паскаль определено  
5 целых типов:

<u>Тип</u>	<u>Диапазон</u>	<u>Требуемая память</u>
Shortint	-128 ... 127	1 байт
Integer	-32767 ... 32768	2 байта
Longint	-2147483648 ... 2147483647	4 байта
Byte	0 ... 255	1 байт
Word	0 ... 65535	2 байта

# Для целых чисел определены операции:

<u>Операция</u>	<u>Обозначение</u>	<u>Пример</u>	<u>Результат</u>
Сложение	+	$2+3$	5
Вычитание	-	$10-8$	2
Умножение	*	$3*4$	12
Целочисленное деление	div	$17 \text{ div } 5$	3
Остаток от деления	mod	$17 \text{ mod } 5$	2
Деление	/	$17 / 5$	3.4 (результат может получиться не целым числом!)

## В Паскале имеется 5 вещественных типов:

<u>Тип</u>	<u>Диапазон</u>	<u>Число значащих цифр</u>	<u>Требуемая память</u>
Real	2.9E-39 до 1.7E+38 ( $2.9 \cdot 10^{-39}$ до $1.7 \cdot 10^{-38}$ )	11-12	6 байт
Single	1.5E-45 до 3.4E+38	7-8	4 байта
Double	5.0E-324 до 1.7E+308	15-16	8 байт
Extended	от 3.4E-4932 до 1.1E+4932	19-20	10 байт
Comp	-9.2E18 до 9.2E18 Является 64 битным целым и хранит только целые значения от $-2E63+1$ до $2E63-1$	19-20	8 байт

Для вещественных чисел определены операции:

<u>Операция</u>	<u>Обозначение</u>	<u>Пример</u>	<u>Результат</u>
Сложение	+	$2+3$	5
Вычитание	-	$10-8$	2
Умножение	*	$3*4$	12
Деление	/	$17 / 5$ $20 / 4$	3.4 5

# Примеры описания переменных:

var

a: byte;

b: integer;

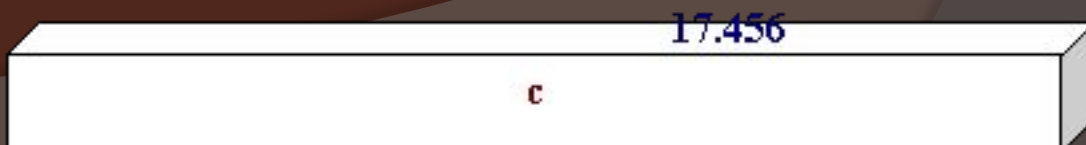
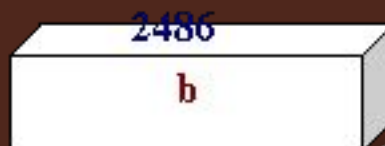
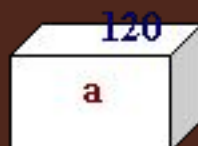
c, d, e: real;

f:double;

- ⦿ Слово var - это сокращение от слова variable (переменная) и является зарезервированным словом.
- ⦿ В разделе описания переменных может содержаться несколько строчек, каждая из которых заканчивается точкой с запятой. Для описания переменных необходимо перечислить через запятую их имена, а затем после двоеточия «:» указать их тип.

Наглядно переменную можно представить как коробочку, в которую можно положить на хранение что-либо. Имя переменной – это надпись на коробочке, значение – это то, что хранится в ней в данный момент, а тип переменной говорит о том, что допустимо класть в эту коробочку.

	120		2486				
	17.456						





# Структура программы

**Program** имя\_программы;

**Uses** Подключение модулей

**Const** Раздел описания констант

**Label** Раздел описания меток

**Type** Раздел описания типов

**Var** Раздел описания

переменных

**begin**

Тело программы (операторы, действия, которые должен выполнить компьютер)

**end.**

# Вопросы:

- Что такое переменная?
- Какие имена можно давать переменным?
- Что такое тип переменной?
- Что такое значение переменной?
- *Определить сколько времени будет падать с яблони яблоко. Высота ветки вводится с клавиатуры.*
- *На дереве висит груша. Вычислите силу тяжести действующую на грушу. Масса груши вводится с клавиатуры.*
- *Треугольник задан величинами своих сторон. Найти его площадь.*

## Д/З

1. Знать, что такое переменная, тип переменной, правила записи имен переменных.
2. Составить программу для вычисления площади треугольника по высоте и основанию.

# Задания

1. Напишите программу, которая запрашивает два числа, находит остаток от деления первого на второе и выводит результат.
2. Даны два числа. Найти их среднее арифметическое.
3. Найти площадь кольца по заданным внешнему и внутреннему радиусам.
4. \* Поменять местами значения переменных  $X$  и  $Y$ , не используя дополнительной переменной.

```
1. Program prim_3;  
Var a,b,x:integer;  
Begin  
    Write('введите два числа ');  
    Readln(a,b);  
    X:=a mod b;  
Writeln('остаток от деления-',x);  
Readln  
End.
```

### 3. Найти площадь кольца по заданным внешнему и внутреннему радиусам.

```
Program prim_4;  
Var r1,r2,s:real;  
Begin  
    Write('введите радиусы кольца ');  
    Readln(r1,r2);  
    S:=abs(2*3.14*r1-2*3.14*r2); {разность площадей берем по модулю, так как не  
    знаем  
какое из колец является внутренним, а какое внешним}  
Writeln('площадь кольца =',s:0:2);  
Readln  
End.
```

Обмен значениями.

Начнем с разбора задания предыдущего урока. 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.

0) A=A1; B=B1;

1) Занесем в переменную A результат суммирования A и B (A := A + B):

A = A1 + B1; B = B1;

2) Занесем в переменную B разность A и B (B := A - B):

т.к. A = A1 + B1; то B=(A1+B1)-B = A1;

3) Занесем в переменную A разность A и B (A := A - B):

A = B1; B = A1;

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);
```

```
Readln
```

```
End.
```

## Проверочный тест:

- 1) Какого описания величин не бывает?
  - а) Значение
  - б) Тип
  - в) Имя
  - г) Присвоение
- 2) Какого типа величин не существует?
  - а) Числовой
  - б) Буквенный
  - в) Символьный
  - г) Логический
- 3) Какое имя нельзя давать величинам?
  - а) X100
  - б) 9 "а"
  - в) OX
  - г) Любое можно
- 4) Что такое обозначение величины?
  - а) Имя
  - б) Тип
  - в) Значение
  - г) Значок присвоения
- 5) Что такое характеристика величины?
  - а) Имя
  - б) Тип
  - в) Значение
  - г) значок присвоения
- 6) Указать тип величины В, если  $V := 3,2$ 
  - а) Целое
  - б) Вещественное
  - в) Натуральное
  - г) Символьное

- 7) Указать тип величины С, если  $C := "65"$ 
  - а) Целое
  - б) Вещественное
  - в) Натуральное
  - г) Символьное

- 8) Указать значение величины М, если

$$X := 6$$

$$Y := 2$$

$$M := Y - X$$

- а) 4
- б) 6
- в) 2
- г) -4

- 9) Указать значение величины X, если

$$A := 9$$

$$B := 3$$

$$V := B + 2$$

$$X := A - V$$

- а) 6
- б) 4
- в) 9
- г) 3

# Логический тип

Переменные логического типа могут принимать значения:

**False**(ложь) или **True** (истина).

**Примеры описания:**

- `VAR b1, b2, b3, b4 : boolean;`

**Примеры присвоения значения:**

- `b1 := True; b2 := False; b3 := not b1;`

# СИМВОЛЬНЫЙ ТИП

- Переменные символьного типа могут принимать значения одного из символов с кодами от #0 до #255, т. е. допускаются буквы русского алфавита и символы псевдографики. Символы с кодами от #0 до #31 (управляющие) в тексте воспринимаются как пробел, а в операторах вывода как символ или команда.



## Примеры описания:

- VAR c, ch, e: char;

## Примеры присвоения значения:

c := 'A'; c := '!'; c := #33;

{ символ "!" задается значением кода }

ch := #13; { код клавиши Enter }

e := #27; { код клавиши Esc }

## Примеры вывода:

Write(c); { выводится символ "!" }

Write(ch); { команда перевода курсора на новую строку }

Write(e); { выводится специальный символ, соответствующий коду #27 }

# Строковый тип

Переменная строкового типа задается набором символов, заключенных в кавычки.

## Примеры описания:

```
VAR s : string[25]; { Длина переменной s  
    может изменяться от 1 до 25 символов }  
s1 : string; { длина переменной s1 не  
    указывается, наибольшее число символов  
    равно 255 }
```

## Примеры присвоения значения:

- s := 'Привет';
- s1 := 'Hello';

# Массив

- Массив - упорядоченные данные одного типа, состоящие из переменных (элементов массива). Элементы массива имеют свой порядковый номер (индекс) и значение. Значение присваивается отдельно каждому элементу массива, либо сразу всем элементам массива (копирование одного массива в другой в случае массивов одинакового типа). Массив может быть многомерным (без ограничения), но занимать не более 65520 байт.

## ○ Примеры описания:

VAR M1 : array[1..200] of integer; { M1, A1 - одномерные массивы целых и вещественных чисел }

A1 : array[100..200] of real;

ch1 : array['A'..'Z'] of char; { ch1, ch2 - одномерные массивы символов }

ch2 : array[0..255] of char;

## Примеры присвоения значения:

- M1[1] := 5;
- M1[2] := 8;
- M1[3] := M1[1];
- M1[100] := 32;
- A1[101] := 0.2;
- A1[102] := 2.31;
- ch1['B'] := 'C';
- ch2[1] := '!'

## Примеры описания:

- TYPE Date\_M = 1..31;  
Ruch\_b = 'А..Я';  
Lat\_b = 'А..Z'; {тип диапазон}  
Otmetka = (2,3,4,5);  
Ball = (1,2,3,4,5,6,7,8,9,10); {перечислимый тип}

VAR t1, t2 : Date\_M;

c1 : Ruch\_b;

s1 : Lat\_b;

a1, a2 : Otmetka;

b : Ball;

## Примеры присвоения значения:

- t1 := 10;
- c1 := 'Л';
- s1 := 'F';
- a1 := 5;
- b := 8;

# Файлы

- Файл именованная область внешней памяти ПК, содержащая различные данные.
- Длина файла ограничена доступной дисковой памятью. Файлы могут быть текстовые, типизированные и нетипизированные.
- Рассмотрим описание текстовых файлов, хранящих информацию в виде строк переменной длины с последовательным доступом к данным.

## Примеры описания:

```
VAR f1, f2 : text;
```

## Примеры присвоения значения:

- `Assign(f1, 'file1.in');`
- `Assign(f2, 'C:\PASCAL\WORK\file2.out');`
- Оператор `Assign` присваивает файловой переменной имя файла на диске. Файл `file1.in` содержится в текущем каталоге. Для файла `file2.out` указывается каталог в формате DOS.

# Записи

- Записи - вводимый программистом тип, определяемый как структура данных разного типа, содержащая названия полей с указанием их типов, расположенных между служебными словами Record и End.



## ○ Примеры описания:

Типе

Sity = Record

Name\_G : String[20]; { *название города* }

Numb\_G : LongInt; { *число жителей* }

Stol\_G : Boolean; { *Столица (Да, Нет)* }

Shir, Dol : Real; { *Координаты: широта,  
долгота* }

End;Var St1, St2, St3 : Sity;

{ *Переменные типа запись* }

○ **Примеры присвоения значения:**

```
St1.Name_G := 'Уфа';
```

```
St1.Numb_G := 1200000; {Отдельным  
полям переменной St1 }
```

```
St2.Name_G := 'Москва';
```

```
St2.Numb_G := 8200000;
```

```
St3 := St1; {Всем полям переменной  
St3}
```

Переменные типа запись могут обозначаться с помощью оператора присоединения:

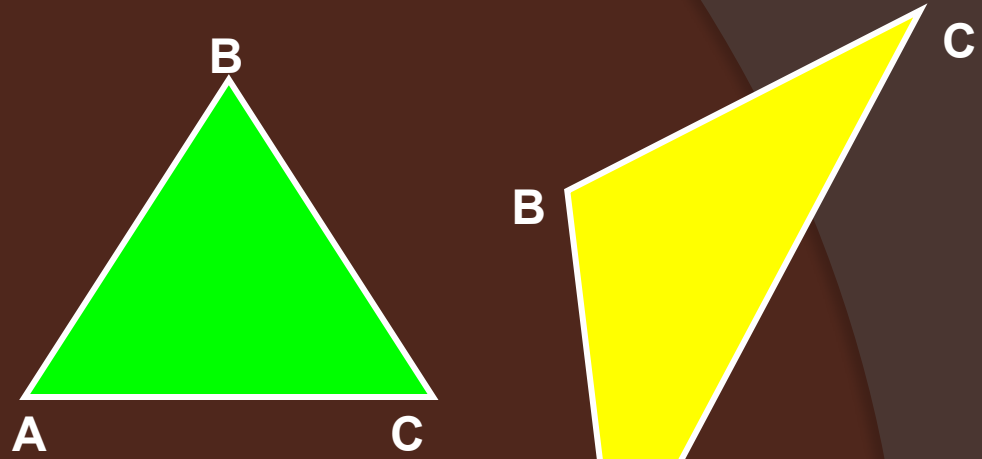
```
WITH "имя переменной" DO "оператор";
```

# пример

```
With St1 do
  begin
    Name_G := 'Уфа';
    Stol_g := True;
    Readln(Shir, Dol);
    z := Numb_G - St2. Numb_G;
  end;
  Writeln(Name_G:5, Numb_G:10);
```

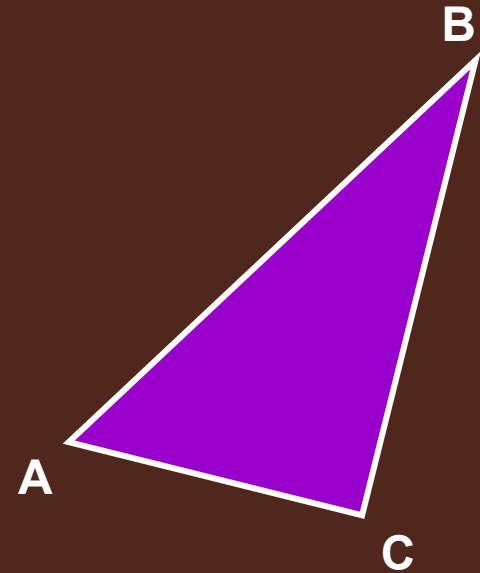
## Пример 4

Написать алгоритм для определения вида треугольника (остроугольный, прямоугольный, тупоугольный), если даны величины двух его углов  $A$  и  $B$  в градусах.



### Решение

```
Program Pr4;  
Var a, b, c: real;  
Begin  
  WriteLn ('Задайте величины углов a, b');  
  ReadLn (a, b);  
  c:=180-a-b;  
  If (a > 90) or (b > 90) or (c > 90)  
    Then WriteLn ('Треугольник тупоугольный')  
  Else  
    If (a = 90) or ( b = 90) or (c = 90)  
      Then WriteLn ('Треугольник прямоугольный')  
    Else WriteLn ('Треугольник остроугольный');  
End.
```



# Примеры программ:

Ракета запускается с точки на экваторе Земли со скоростью  $V$ (км/с) в направлении движения Земли по орбите вокруг Солнца. Каким будет результат запуска ракеты в зависимости от скорости  $V$ ?

$V < 7.8$  Ракета упадет на Землю

$7.8 \leq V < 11.2$  Ракета станет спутником Земли

$11.2 \leq V < 16.4$  Ракета станет спутником Солнца

$V > 16.4$  Ракета покинет пределы Солнечной системы

**Program PR5;**

**Var V : real;**

**Begin**

**WriteLn ('Задайте значение скорости');**

**ReadLn (V);**

**If V < 7.8 Then WriteLn('Ракета упадет на Землю');**

**If (V >=7.8) and (V <11.2) then writeLn ('Ракета станет спутником Земли');**

**If (V >=11.2) and (V <=16.4) then writeLn ('Ракета станет спутником Солнца');**

**If V >16.4 then writeLn ('Ракета покинет пределы Солнечной системы');**

**ReadLn;**

**end.**