

# Турбо Паскаль

Оператор присваивания.  
Арифметические выражения.  
Типы данных

# Оператор присваивания

**Оператор присваивания** - основной оператор любого языка программирования. Общая форма записи оператора:  
имя величины := выражение

Например,  $V:=A$ ; или  $V:=A+1$ ;

При помощи оператора присваивания переменной могут присваиваться константы и выражения, значения переменных любого типа.

Как только в программе встречается переменная, для неё в памяти отводится место. Оператор присваивания помещает значение выражения в место, отведённое переменной.

Если в процессе выполнения программы встречается переприсваивание (т.е. та же самая переменная принимает другое значение), то старое значение переменной стирается, на свободное место записывается новое значение. Команда присваивания позволяет лучше понять смысл слова переменная (т.е. меняющая своё значение по ходу программы).

Выражение может быть арифметическим, логическим или литерным. Важно, чтобы тип величины был согласован с видом выражения.

# Арифметические выражения

Арифметические выражения должны быть записаны в так называемой линейной записи согласно следующим правилам:

- выражение должно быть записано в виде линейной цепочки символов;

- используемые операции приведены в таблице:

## НАЗВАНИЕ ОПЕРАЦИИ    ФОРМА ЗАПИСИ

сложение	$x + y$
вычитание	$x - y$
умножение	$x * y$
деление	$x / y$

- нельзя опускать знаки операций, например, писать  $5b$ . Для записи произведения чисел 5 и  $b$  надо писать  $5*b$ ;
- аргументы функций ( $\sin$ ,  $\cos$  и др.) как и аргументы вспомогательных алгоритмов, записываются в круглых скобках, например  $\sin(x)$ ,  $\cos(4*x)$ .

- Порядок выполнения операций при вычислении арифметических выражений можно регулировать при помощи скобок по обычным правилам. Там, где скобки отсутствуют, ЭВМ выполняет операции в следующем порядке:
- вычисляет значение всех алгоритмов-функций и стандартных функций;
- выполняет справа налево все операции возведения в степень;
- выполняет слева направо все операции умножения и деления;
- выполняет слева направо все операции сложения и вычитания.
- В нашем случае сначала переменной `number1` присваивается значение равное 3 и переменной `number2` присваивается значение равное 4, затем вычисляется значение выражения  $(number1 + number2)$  и оно

Сумма чисел посчитана.

Теперь надо вывести ее значение на экран.

Для этого используют оператор Write - записать (вывести) на экран значение переменной, записанной в скобках. В нашем случае значение переменной number1, затем символ + , далее значение переменной number2, символ = и, наконец, значение результата result.

И, наконец, в конце раздела операторов стоит служебное слово End, после которого стоит точка.

- **Задание.** Наберите текст программы на компьютере и выполните ее (для запуска программы воспользуйтесь комбинацией клавиш Ctrl и F9).
- **Внимание!** Не забывайте о порядке на дискете и в файле:
- имя программы должно соответствовать ее содержанию,
- имя файла должно быть таким же, как и имя программы,
- файлы, содержащие программы, относящиеся к одной теме, должны находиться в одном каталоге,
- название этого каталога должно отражать его содержание.

**Задание.** Измените программу так, чтобы она подсчитывала сумму четырех чисел. Сохраните файл на дискете, для этого изменю F10-File выберите команду Save и в предложенной строке наберите путь а:\Vvod\Summa (каталог Vvod должен быть уже организован для файлов, содержащих программы данной темы).

# Основные определения

- Познакомимся с основными понятиями языка.
- **Алгоритм** - четкая последовательность действий, необходимая для решения задачи.
- **Программа** - алгоритм, записанный на языке программирования.
- **Алфавит языка** - набор элементарных символов, используемый для составления программ. Алфавит содержит:
  - 52 буквы латинского алфавита (строчные и заглавные);
  - арабские цифры (0-9);
  - специальные символы;
  - знаки математических действий (+ - \* / );
  - знаки пунктуации (. : , ; " ` );
  - скобки ( [ ] ( ) { } );
  - знак пробела;
  - знаки отношений (< > =).



# Идентификатор

- **Идентификатор (имя)** – имя какого-либо элемента программы, которое должно удовлетворять следующим требованиям:
- длина имени не должна превышать 63 символов;
- идентификатор может состоять из букв, цифр и символов подчеркивания (значение ASCII \$5F);
- идентификатор должен начинаться с буквы или с символа подчеркивания;
- буквы русского алфавита и пробелы не могут входить в идентификатор в Turbo Pascal;
- имя не должно совпадать с зарезервированным (служебным) словом;
- прописные и строчные буквы воспринимаются одинаково.
- Правильно выбранные идентификаторы значительно облегчают чтение и понимание программы, а также уменьшают вероятность появления ошибок при модификации программ. Например, значение даты удобнее обозначить идентификатором Date, чем просто буквой D или любым другим символом.

# Зарезервированные (служебные) слова

**Зарезервированные (служебные) слова** - это слова, использующиеся только по своему прямому назначению. Их нельзя использовать в качестве имен переменных, так как они выполняют определенную смысловую нагрузку.

Примеры зарезервированных слов: AND, GOTO, PROGRAM, ELSE, IF, RECORD, NOT, ARRAY, REPEAT, UNTIL, BEGIN, IN, SET, END, CASE, CONST, USES, INTERFACE, STRING, LABEL, THEN, OF, DIV, TO, VAR, DO, TYPE, WHILE, DOWNT0, FILE, FUNCTION, PROCEDURE и другие.

**Переменные (Var)** - вид данных, который может изменять свое значение в ходе программы, описывают переменные после зарезервированного слова Var.

**Константы (Const)** - вид данных, который является постоянным на всем протяжении выполнения программы, описывают константы после зарезервированного слова Const.

**Комментарии** - некоторая запись, служащая для пояснения программы, которая записывается в фигурных скобках.

# Типы данных

- Для временного хранения информации в операторах памяти машины в языке Паскаль используются константы и переменные. Они могут быть различных типов:
- целых чисел (см. ниже);
- действительных чисел (real);
- символьный тип (char);
- строковый (string);
- логический (boolean);
- сложные (комбинированный (record), множественный (set) и другие).

# Целые типы:

Название	Длина в байтах	Диапазон значений
Byte	1	0 ... 255
ShortInt	2	-128 ... 127
Word	2	0 ... 65535
Integer	4	-32768 ... 32767
LongInt	4	-2147483648 ... 2147483647

Над целыми типами определены такие операции:

"+" - сложение;

"\*" - умножение;

"-" - вычитание;

div - целочисленное деление;

mod - получение остатка от целочисленного деления.

# Вещественные типы:

- Вещественные типы представляются с некоторой точностью, которая зависит от компьютера. Вам необходимо знать, что вещественный тип разделяется на несколько типов, но использовать мы будем вещественные данные только типа Real, которые занимают 6 байт, имеют диапазон возможных значений модуля от  $2.9E-39$  до  $1.7E+38$  и точность представления данных - 11...12 значащих цифр.
- **Примечание.** Несмотря на то, что в Turbo Pascal имеется широкий выбор вещественных типов, доступ к некоторым из них (single, double, extended) возможен при особых режимах компиляции. Особое положение в Turbo Pascal занимает тип comp, трактуемый как вещественное число без экспоненциальной и дробной частей. Он сохраняет 19 - 20 значащих цифр и знак числа. В то же время comp полностью совместим с любыми другими вещественными типами.
- В языке Паскаль числа могут быть представлены в двух видах: с фиксированной точкой и плавающей точкой.
- Числа с фиксированной точкой изображаются десятичным числом с дробной частью, которая может быть и нулевой. Например. 27.9. 5.00

Такие большие числа, как 137.000.000, можно записать в формате с десятичным порядком:  $1.37 \cdot 10^8$ . В Turbo Pascal для подобных чисел принята форма записи **mEr**, где m - мантисса; E - признак записи числа с десятичным порядком; r - показатель степени числа 10. Такое представление чисел и называется форматом с плавающей точкой. Число  $1.37 \cdot 10^8$  в этом формате запишется так: 1.37E+8. Рассмотрим еще несколько примеров:

Математическая запись	Запись на Паскале
-----------------------	-------------------

$4 \cdot 10^{-4}$	4E -4
$0,62 \cdot 10^5$	0.62E+5
$-10,88 \cdot 10^{12}$	-10.88E12

По умолчанию, действительные числа выводятся в виде чисел с плавающей точкой. Такое представление чисел не очень нравится пользователям. Поэтому мы будем “заставлять” компьютер выдавать действительные числа в более привычном варианте следующим образом:

**R:m:n**, где R – действительное число, m – общее количество позиций, отводимых числа, n – количество позиций, отводимых для дробной части.

Например, если мы хотим вывести на экран число Chislo с фиксированной точкой, причем знаем, что для вывода этого числа достаточно 7 мест, а вывод дробной части ограничим сотыми, то мы

# СИМВОЛЬНЫЙ ТИП (char):

Значением величины данного типа является символ из множества всех символов компьютера: русская или латинская большая или маленькая буква, цифра, знак препинания, специальный знак (например, "+", "-", "\*", "/", "", "=", и др.) или пробел " ". Каждый из символов имеет уникальный номер от 0 до 255, т. е. внутренний код, который возвращает функция ORD. Символьная константа – любой символ языка, заключённый в одинарные кавычки. Например,

Var

Simvol : char;

# Строковый тип (string):

- Значением строковой величины является строка переменной длины (может быть, пустая). Строковая константа представляет собой произвольную последовательность символов, заключенную в одинарные кавычки. Например,
- Var  
    Stroka : string;



# Логический тип (boolean):

- Логический тип данных часто называют булевым по имени английского математика Д. Буля, создателя математической логики. В языке Паскаль имеются две логические константы TRUE и FALSE. Логическая переменная принимает одно из этих значений и имеет тип Boolean. Для сравнения данных предусмотрены следующие операции отношения:  $<$ ,  $<=$ ,  $=$ ,  $<>$ ,  $>$ ,  $>=$ . Результат операции отношения имеет логический тип, т.е. результат операции может быть истинным или ложным, например,  $3>5$  – ложь.
- Над величинами этого типа можно выполнять специфические логические операции OR - или; AND - и; NOT - не.
- Более подробно этот тип данных мы рассмотрим при изучении условного оператора.

# Сложные типы:

- К сложным или структурированным типам относятся массивы, записи, множества, которые требуют специального изучения и здесь рассматриваться не будут.
- **Задание**. Откройте новый файл. Создадим программу, в которой опишем несколько переменных разного типа, введем в них значения и выведем на экран.

- Program TipDann;

Uses

Crt

Var

Chislo1 : Integer;

Chislo2 : Real;

Simvol : Char;

Stroka : String;

Logika : Boolean;

Begin

ClrScr;

Chislo1:=12;

Chislo2:=Chislo1\*2;

Chislo2:=Chislo2/5;

Simvol:='d';

Stroka:='Строчка';

Logika:= Chislo1> Chislo2;

WriteLn ('Вывод значений:');

WriteLn ('Значение переменной Chislo1 : ',Chislo1);

WriteLn ('Значение переменной Chislo2 : ',Chislo2:5:2);

WriteLn ('Значение переменной Simvol : ',Simvol);

WriteLn ('Значение переменной Stroka : ',Stroka);

WriteLn ('Значение переменной Logika : ',Logika);

End.

Внимательно рассмотрите каждую строчку программы. Обратите особое внимание на описание переменных: Chislo1 - переменная целого типа, Chislo2 - действительного, Simvol - символьного, Stroka - строкового, Logika - логического. Далее в основной программе идет присвоение переменной Chislo1 целого числа 12, переменной Chislo2 - целого числа 24. Обратим внимание, что переменной действительного типа Chislo2 присвоено целое число; никакой ошибки нет, т. к. множество целых чисел является подмножеством множества действительных чисел. Следующая строчка еще более интересная: переменной Chislo2 присваивается значение той же переменной, только деленной на 5.

Такое присваивание используют в программах, если предыдущее значение этой переменной уже не понадобится и для более рационального использования описанных переменных. Для того чтобы переменной символьного типа присвоить какой-либо символ (например, d), надо этот символ записать в одинарных кавычках (знак «'»). Аналогично поступают с переменными строкового типа (смотри следующую строку программы). А про переменные логического типа мы знаем, что им можно присваивать только два значения: True и False. В этой программе мы присвоим значение результата сравнения двух переменных, здесь оно будет равно True. А теперь выведем присвоенные значения на экран.

**Задание.** Измените значения переменных и проанализируйте результаты выполнения программы. Сохраните программу на дискете под правильным именем в соответствующем каталоге. Распечатайте программу.