

Справочный материал по программированию в Delphi

Код программы в Delphi представляет собой последовательность инструкций, которые довольно часто называют операторами. Одна инструкция от другой отделяется точкой с запятой.

Каждая инструкция состоит из *идентификаторов*. Идентификатор может обозначать:

- Инструкцию языка (**:=, if, while, for**) это «ключевые слова»;
- переменную;
- константу (целое или дробное число);
- арифметическую (**+, -, *, /, div, mod**) или логическую (**and, or, not**) операцию;
- подпрограмму (процедуру или функцию);
- отмечать начало (**procedure, function**) или конец (**end**) подпрограммы или блока (**begin, end**).

Переменные объявляются в разделе объявления переменных. Значения переменных получаются и могут изменяться в ходе выполнения программы. *Тип переменных* должен быть задан при объявлении.

Константы – данные, значения которых установлены в разделе объявления констант и не изменяются. Тип констант распознается компилятором автоматически.

К константам и переменным можно обращаться по имени (*идентификатору*).

Типы данных Delphi

Типы данных определяет множество значений, которые могут принимать элементы программы, и действиями (операциями), которые можно применить к этим значениям.

Типы данных можно разделить на группы:

- Простые (не содержат других типов)
- Структурные
- Указатели
- Процедурные
- Вариантные

Совместимость типов означает, что типы могут быть равны друг другу или могут преобразовываться один в другой.

Например: Целое число может быть автоматически преобразовано в Вещественное, но не наоборот.

Таблица 1 Типы данных в Delphi

Название типа		Длина, байт	Диапазон значений	
Простые				
SMALLINT	Порядковые	Целые типы Integer	2	-32768..+32767
BYTE			1	0..255
SHORTINT			1	-128..127
WORD			2	0..65535
Longword			4	0..4 294 967 295
LONGINT			4	-2147483648.. 2147483647
Int64			8	$-2^{63} \dots 2^{63} - 1$
CHAR			Символьный тип	1
BOOLEAN	Логический тип	1	TRUE, FALSE	
REAL (есть и другие типы)	Вещественный тип	6	2.9E-39..1.7E+38	
Структурированные				
ARRAY	массивы			
RECORD	записи			
Строки				
STRING	Тип - строка	1..256	Строка из любых символов	

Delphi и Object Pascal поддерживают и наиболее универсальный целый тип – **Integer**, который Эквивалентен **Longint**.

Более подробную информацию по типам данных можно прочитать в [Каталог программиста](#)

Объявление (описание) переменных

Переменные отличаются *областью действия*:

- ❑ локальные (внутри процедуры или программного модуля)
- ❑ глобальные (открытые) – доступны всему проекту

Для явного описания переменных используется ключевое слово **Var**

Например:

```
Var  A: String ;  
      B: Integer ;  
      X: Real;  
      F: Boolean;
```

Для описания константы используется ключевое слово **Const**

Например:

```
Const C = 10;
```

Преобразование типов и действия над ними (встроенные процедуры и функции):

Название	Действие
n:=ROUND(x)	Округляет REAL x до ближайшего INTEGER n
n:=TRUNC(x)	Усекает REAL x до INTEGER n, отбрасывая дробную часть.
n:=ORD(x)	Получает INTEGER n=порядковому номеру в кодовой таблице для x – любого порядкового типа, т.е. параметр x может иметь тип CHAR, BOOLEAN, INTEGER и т.п.
x:=CHR(n)	Получает символ CHAR x по его порядковому номеру n в кодовой таблице.
xp:=PRED(x)	Получает xp = предыдущему значению x для любого порядкового типа.
xs:=SUCC(x)	Получает xs = последующему значению x для любого порядкового типа.
f:=ODD(n)	Получает BOOLEAN f= TRUE , если n-нечетное и f= FALSE , если n-четное (проверка на четность)

Логические операции над целыми или логическими операндами :

Not	Логическое НЕ
And	Логическое И
or	Логическое ИЛИ
xor	Исключительное ИЛИ

Таблица истинности для логических операций над данными типа BOOLEAN:

Переменная 1	Переменная 2	Операции			
		not	and	or	xor
false	-	true	-	-	-
true	-	false	-	-	-
false	false	-	false	false	false
false	true	-	false	true	true
true	false	-	false	true	true
true	true	-	true	true	false

Материал взят с сайта: [Самоучители по программированию](#)

Для Delphi [samouchka](#)

Более подробную информацию по типам данных можно прочитать в [Каталог программиста](#)