

Гомельский государственный университет
имени Франциска Скорины

Типы данных

Составил: Ассистент кафедры общей физики
Соколов С.И.

Гомель, 2015

Типы данных

Каждая инструкция состоит из идентификаторов.

Идентификатор может обозначать:

.Инструкцию языка (`:=`, `if`, `while`, `for`);

.переменную;

.константу (целое или дробное число);

.арифметическую (`+`, `-`, `*`, `/`) или логическую (`and`, `or`, `not`) операцию;

.подпрограмму (процедуру или функцию);

.отмечать начало (`procedure`, `function`) или конец (`end`) подпрограммы ИЛИ блока (`begin`, `end`).

Целые типы

Тип	Диапазон	Формат
Shortint	-128-127	8 битов
Smallint	-32 768 + 32 767	16 битов
Longint	-2 147 483 648 + 2 147 483 647	32 бита
Int64	$-2^{63} + 2^{63} - 1$	64 бита
Byte	0-255	8 битов, беззнаковый
Word	0-65 535	16 битов, беззнаковый
Longword	0 - 4 294 967 295	32 бита, беззнаковый

Вещественные (дробные) типы

Тип	Диапазон	Значащих цифр	Байтов
Real48	$2.9 \times 10^{-39} - 1.7 \times 10^{38}$	11-12	06
Single	$1.5 \times 10^{-45} - 3.4 \times 10^{38}$	7-8	04
Double	$5.0 \times 10^{-324} - 1.7 \times 10^{308}$	15-16	08
Extended	$3.6 \times 10^{-4951} - 1.1 \times 10^{4932}$	19-20	10
Comp	$2^{63} + 1 - 2^{63} - 1$	19-20	08
Currency	-922 337 203 685 477.5808 --922 337 203 685 477.5807	19-20	08

Символьный тип

- тип `Ansichar` — это символы в кодировке ANSI, которым соответствуют числа в диапазоне от 0 до 255;
- тип `widechar` — это символы в кодировке Unicode, им соответствуют числа от 0 до 65 535.

Строковый тип

- тип `shortstring` представляет собой статически размещаемые в памяти компьютера строки длиной от 0 до 255 символов;
- тип `Longstring` представляет собой динамически размещаемые в памяти строки, длина которых ограничена только объемом свободной памяти;
- тип `WideString` представляет собой динамически размещаемые в памяти строки, длина которых ограничена только объемом свободной памяти. Каждый символ строки типа `WideString` является Unicode-символом.

Логический тип

Логическая величина может принимать одно из двух значений True (истина) или False (ложь). В языке Delphi логические величины относятся к типу Boolean.

В общем виде инструкция объявления переменной
выглядит так:

Var

Имя : тип;

где:

.имя — имя переменной;

.тип — тип данных, для хранения которых
предназначена переменная.

Пример:

Var

a : Real; b : Real; i : Integer;

В общем виде инструкция объявления именованной константы выглядит следующим образом:

КОНСТАНТА = значение;

где:

.константа — имя константы;

.значение — значение константы.

```
const
```

```
Bound = 10;
```

```
Title = 'Скорость бега';
```

```
pi = 3.1415926;
```

Инструкция присваивания

В общем виде инструкция присваивания выглядит так:

Имя := Выражение;

где:

.Имя — переменная, значение которой изменяется в результате выполнения инструкции присваивания;

:= — символ инструкции присваивания.

.Выражение — выражение, значение которого присваивается переменной, имя которой указано слева от символа инструкции присваивания.



```
Summa := Cena * Kol;
```

```
Skidka := 10;
```

```
Found := False;
```

Алгебраические операторы

Оператор	Действие
+	Сложение
-	Вычитание
*	Умножение
/	Деление
DIV	Деление нацело
MOD	Вычисление остатка от деления

Примеры выражений:

$123 - 0.001 * i + 1$

$A + B / C \text{ Summa} * 0.75 + (B1 + B3 + B3) / 3 - \text{Cena MOD } 100$

Правила определения типа выражения

Оператор	Тип операндов	Тип выражения
$*$, $+$, $-$	Хотя бы один из операндов real	real
$*$, $+$, $-$	Оба операнда integer	integer
$/$	real или integer	Всегда real
DIV, MOD	Всегда integer	Всегда integer

Выполнение инструкции присваивания

Инструкция присваивания выполняется следующим образом:

1. Сначала вычисляется значение выражения, которое находится справа от символа инструкции присваивания.
2. Затем вычисленное значение записывается в переменную, имя которой стоит слева от символа инструкции присваивания.

Например, в результате выполнения инструкций:

- $i:=0$; — значение переменной i становится равным нулю;
- $a:=b+c$; — значением переменной a будет число, равное сумме значений переменных b и c ;
- $j:=j+1$; — значение переменной j увеличивается на единицу.

Инструкция присваивания считается верной, если тип выражения соответствует или может быть приведен к типу переменной, получающей значение. Например, переменной типа `real` можно присвоить значение выражения, тип которого `real` или `integer`, а переменной типа `integer` можно присвоить значение выражения только типа `integer`.

Так, например, если переменные `i` и `n` имеют тип `integer`, а переменная `d` — тип `real`, то инструкции

```
i:=n/10; i:=1.0;
```

неправильные, а инструкция

```
d:=i+1;
```

 правильная.

Во время компиляции выполняется проверка соответствия типа выражения типу переменной. Если тип выражения не соответствует типу переменной, то компилятор выводит сообщение об ошибке:

```
Incompatible types ... and ...
```

где вместо многоточий указывается тип выражения и переменной. Например, если переменная n целого типа, то инструкция $n := m/2$ неверная, поэтому во время компиляции будет выведено сообщение :

```
Incompatible types 'Integer' and 'Extended'.
```


Математические функции

Функция	Значение
Abs (n)	Абсолютное значение n
Sqrt (n)	Квадратный корень из n
Sqr (n)	Квадрат n
Sin (n)	Синус n
Cos (n)	Косинус n
Arctan (n)	Арктангенс n
Exp(n)	Экспонента n
Ln(n)	Натуральный логарифм n
Random(n)	Случайное целое число в диапазоне от 0 до n- 1

Функции преобразования

Функция	Значение функции
Chr(n)	Конвертирует целое n число в соответствующий символ
IntToStr (k)	Конвертирует целое число k в строку

Функция	Значение функции
FloatToStr (n)	Конвертирует вещественное число n в строку
FloatToStrF(n, f , k,m)	Конвертирует вещественное число n в строку с форматированием. При вызове функции указывают: f — формат (способ изображения); k — точность (нужное общее количество цифр); m — количество цифр после десятичной точки
StrToInt (s)	Преобразует строку с целым значением s в целое число
StrToFloat (s)	Преобразует строку с вещественным значением s в вещественное число
Round (n)	Целое, полученное путем округления n по известным правилам
Trunc (n)	Целое, полученное путем отбрасывания дробной части n
Frac(n)	Дробное, представляющее собой дробную часть вещественного n
Int (n)	Дробное, представляющее собой целую часть вещественного n

Использование функций

Обычно функции используют в качестве операндов выражений. Параметром функции может быть константа, переменная или выражение соответствующего типа. Ниже приведены примеры использования стандартных функций и функций преобразования.

```
n := Round((x2-x1)/dx);  
x1 := (-b + Sqrt(d)) / (2*a);  
m := Random(10);  
cena := StrToInt(Edit1.Text);  
Edit2.Text := IntToStr(100);  
mes := 'x1=' + FloatToStr(x1);
```