

# **Простые типы данных на языке программирования Turbo Pascal.**

# **Типы данных в языке программирования Turbo Pascal делятся на пять основных классов:**

- **-простые типы;**
- **-структурированные типы;**
- **-ссылочные типы;**
- **-процедурные типы;**
- **-объектные типы.**

## Простые типы данных

```
graph TD; A[Простые типы данных] --> B[Порядковые:]; A --> C[Вещественные]; B --> B1[-Целый]; B --> B2[-Логический]; B --> B3[-Символьный]; B --> B4[-Ограниченный]; B --> B5[-Перечисляемый];
```

### Порядковые:

-Целый


-Логический

-Символьный

-Ограниченный

-Перечисляемый

Вещественные

- 
- Функция  $PRED(x)$  определяет предыдущее значение величины  $x$ ,
  - Функция  $SUCC(x)$  определяет последующее значение величины  $x$ ,
  - Функция  $ORD(x)$  возвращает порядковый номер величины  $x$ .

## Целочисленные типы данных.

Тип	Диапазон возможных значений	Размер памяти
<u>Shortint</u>	от -128 до 127	1 байт
<u>Integer</u>	32767 до 32768	2 байта
<u>Longint</u>	-2147483648 до 2147483647	4 байта
<u>Byte</u>	0 до 255	1 байт
<u>Word</u>	0 до 65535	2 байта

# Для целых чисел определены следующие стандартные функции:

- –  $\text{Odd}(x)$  – возвращает результат логического типа: для четного аргумента – `false`, а для нечетного – `true`;
- –  $\text{Succ}(x)$  – возвращает следующее целое число  $(x+1)$ ;
- –  $\text{Pred}(x)$  – возвращает предыдущее целое число  $(x-1)$ ;
- –  $\text{Ord}(x)$  – возвращает аргумент  $x$ ;
- –  $\text{Abs}(x)$  – возвращает модуль  $x$ ;
- –  $\text{Srq}(x)$  – возвращает квадрат числа  $x$ ;
- –  $\text{Sqrt}(x)$  – возвращает значение корня числа  $x$ ;

- –  $\text{Exp}(x)$  – возвращает  $e$  в степени  $x$  (экспоненту), результат вещественного типа;
- –  $\text{Sin}(x)$  – возвращает синус  $x$ , результат вещественного типа;
- –  $\text{Cos}(x)$  – возвращает косинус  $x$ , результат вещественного типа;
- –  $\text{Ln}(x)$  – возвращает натуральный логарифм  $x$ , результат вещественного типа;
- –  $\text{Dec}(x, i)$  – уменьшает значение  $x$  на  $i$ , если  $i$  не задано, то на 1;
- –  $\text{Inc}(x, i)$  – увеличивает значение  $x$  на  $i$ , если  $i$  не задано, то на 1

# Вещественные типы данных.

Тип	Диапазон возможных значений	Размер памяти
REAL	$2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{-38}$	6 байт
SINGLE	$1,5 \cdot 10^{-45} \dots 3,4 \cdot 10^{-38}$	4 байт
DOUBLE	$5,0 \cdot 10^{-324} \dots 1,7 \cdot 10^{-308}$	8 байт
EXTENDED	$3,4 \cdot 10^{-4951} \dots 1,1 \cdot 10^{-4932}$	10 байт
COMP	$2^{63} + 1 \dots 2^{63} - 1$	8 байт



# Над данными вещественного типа определены следующие операции:

- $\sin(x)$  Возвращает синус, угол в радианах
- $\cos(x)$  Возвращает косинус, угол в радианах
- $\arctan(x)$  Возвращает арктангенс
- $\text{abs}(x)$  Возвращает абсолютную величину
- $\text{sqr}(x)$  Возвращает квадрат
- $\text{exp}(x)$  Возвращает экспоненту
- $\ln(x)$  Возвращает логарифм натуральный
-

- `trunc(x)` Отбрасывает дробную часть  
`trunc(5.6)=5`
- `round(x)` Округляет до ближайшего целого  
`round(5.6)=6`
- `frac(x)` Выделяет дробную часть  
`frac(10.1)=0.1`
- `int(x)` Выделяет целую часть `int(10.1)=10.0`
- `random` Датчик случайных чисел (ДСЧ)
- `random(x)` Датчик случайных чисел (ДСЧ)
- `randomize` Инициализация ДСЧ

# Логический тип данных и логические выражения.

Значения операндов		Результат операции		
X	Y	not X	X and Y	X or Y
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

# Перечисляемый тип данных.

- Type

```
Rainbow = (RED, ORANGE, YELLOW,  
          GREEN, LIGHT_BLUE, BLUE, VIOLET);
```

- type Traffic\_Light= (RED, YELLOW,  
GREEN);

- var Section:Traffic\_Light;

# Символьный тип данных.

- Тип данных CHAR.
- 1) ORD (C) принимает значение кода символа C;
- 2) значение функции CHR(I) является СИМВОЛ С КОДОМ
  
- Например: ORD('A ')=65
- CHR(ORD(C))=C ; CHR (65) = A ;

## ● Тип данных **STRING** .

### ● 1) Функция **LENGTH**

- $C := \text{LENGTH}(str)$ ; Переменной  $C$  будет присвоено целое значение, показывающее количество символов в строковой переменной  $str$  .
- 

### ● 2) Функция **CONCAT** - сцепление строк в порядке их перечисления.

- $str := \text{CONCAT}(st1, st2, \dots, stN)$ ;  $str$ -переменная типа **STRING**, состоящая из строк  $st1, \dots, stN$ .
- 

### ● 3) Функция **POS**

- $P := \text{POS}(st1, st2)$ ;  $P$ -целое число, показывающее номер позиции, с которой начинается строка  $st1$  в строке  $st2$ .
- 

### ● 4) Функция **COPY**

- $S1 := \text{COPY}(str, I, J)$ ;  $S1$ -символьная подстрока, выделенная из строки  $str$  с позиции  $I$ , длиной  $J$  символов.
-

- 5) Процедура DELETE(Str, I, J);
- Из строки str удаляется J символов, начиная с I позиции.
- 
- 6) Процедура INSERT(Str1, Str2, I);
- Строка Str1 вставляется с I позиции в строку Str2.
- 
- 7) Процедура STR (V, SI);
- Числовое значение переменной V преобразуется в строку символов и записывается в строку SI.
- 
- 8) Процедура VAL (SI, V, C);

# Интервальный (ограниченный) тип данных.

- Примеры отрезков:
- 1..10
- -15..25
- 'a'..'z'



# Константы.

- Иногда, бывает так, что у нас в программе встречаются величины, которые не меняются в процессе выполнения программы (т.е. фактически они противоположность переменных). Такие величины называются константами. Константы указываются в специальном разделе, который называется `Const`.

## Домашнее задание.

- Подготовить ответы на вопросы:
- 1. Чем характеризуется переменная?
- 2. Перечислите типы данных?
- 3. Как описывается переменная?
- 4. Какова структура программы?
- 5. С чего начинается программа?
- 6. Чем заканчивается программа?
-