

Простые типы данных и операции над ними (язык Паскаль)

Любые данные, т.е. константы, переменные, значения функций или выражения, характеризуются своими типами.

Тип определяет **множество допустимых значений**, которые может иметь тот или иной объект, а также **множество допустимых операций**, которые применимы к нему.

Кроме того, тип определяет также и **формат внутреннего представления данных в памяти ПК**.

Простые типы данных

Порядковые

Вещественные

Целый

Символьный

Логический

Перечислимый

Тип-диапазон

Стандартные

Определяемые пользователем

Порядковые типы отличаются тем, что каждый из них имеет конечное число возможных значений. Эти значения можно определенным образом упорядочить (отсюда -название типов) и, следовательно, с каждым из них можно сопоставить некоторое целое число - порядковый номер значения.

Вещественные типы тоже имеют конечное число значений, которое определяется форматом внутреннего представления вещественного числа. Однако количество возможных значений вещественных типов настолько велико, что сопоставить с каждым из них целое число (его номер) не представляется возможным

Целые типы данных

Тип	Паскаль	
	Диапазон	Размер в байтах
Shortint	-128..127	1
Integer	-32 768..32 767	2
Smallint		
Longint	-2 147 483 648 - 2 147 483 647	4
Byte	0..255	1
Word	0..65 535	2

**Над переменными целочисленного типа
определены следующие**

арифметические операции :

+, -, *, /, div, mod

**Результат выполнения операции / (деления) всегда
получается вещественного типа!!!!!!!!!!!!!!!!!!!!!!**

операции отношения:

=, <>, <, >, <=, >=

вырабатывающие результат логического типа.

Стандартные математические функции

Запись на Паскале	Обычная запись, пояснения
sin(x), cos(x), arctan(x)	sin x, cos x, arctg x
ln(x)	натуральный логарифм x
exp(x)	экспонента x
abs(x)	абсолютная величина (модуль) x
sqr(x)	квадрат x
sqrt(x)	корень квадратный из x
pred(x)	возвращает число, предшествующее x
succ(x)	возвращает число, следующее за x
odd(x)	проверяет, является ли аргумент нечетным числом и возвращает результат логического типа: для четного аргумента – false; для нечетного – true.

Функция Random(n)- возвращает случайное целое число **в диапазоне от 0 до n- 1**

Процедуры

inc(x) возвращает **значение, следующее за x** (для арифметических типов данных это эквивалентно оператору $x:=x+1$).

inc(x,k) возвращает **k-е значение, следующее за x** (для арифметических типов данных это эквивалентно оператору $x:=x+k$).

dec(x) возвращает **значение, предшествующее x** (для арифметических типов данных это эквивалентно оператору $x:=x-1$).

dec(x,k) возвращает **k-е значение, предшествующее x** (для арифметических типов данных это эквивалентно оператору $x:=x-k$).

Вещественный тип данных

Вещественные значения могут быть записаны в двух формах:

в форме с фиксированной запятой

в форме с плавающей запятой
(экспоненциальная форма).

Значения в форме с плавающей запятой представляются в виде **mEr** , где **m** -мантисса, **r** – показатель степени числа 10 (порядок), например **$8.7612E+2$** , **$1.34E-3$** . Соответственно в форме с фиксированной запятой эти числа запишутся в виде **876.12** и **0.00134**

Вещественные типы

Тип	Диапазон	Значащих цифр	Длина, байтов
Real	$2.9 * 10^{-39} - 1.7 * 10^{38}$	11-12	6
Single	$1.5 * 10^{-45} - 3.4 * 10^{38}$	7-8	4
Double	$5.0 * 10^{-324} - 1.7 * 10^{308}$	15-16	8
Extended	$3.6 * 10^{-4951} - 1.1 * 10^{4932}$	19-20	10
Comp	$2^{63} + 1 - 2^{63} - 1$	19-20	8
Currency	-922337203685477.5808 - 922337203685477.5807	19-20	8

Над данными вещественного типа определено

четыре арифметические операции (+, -, *, /),
операции отношения (=, <>, <, >, <=, >=)
восемь стандартных математических
функций (те же, что и над целыми типами)

Функции преобразования:

Int(x)- возвращает целую часть x;

Frac(x)- возвращает дробную часть x;

trunc(x) - отбрасывает дробную часть
числа x;

round(x) - округляет до ближайшего целого
числа.

Символьный тип

Значением является один символ, которому поставлен в соответствие определенный код – число:

Char - универсальный символьный тип, соответствуют числа в диапазоне от **0** до **255**

Над символьным типом определены **операции**
отношения:

Пусть S1 и S2 – переменные символьного типа, тогда
 $S1 < S2$, если код(S1) < код (S2)

Есть **две специальные функции**,
позволяющие определить символ по его
коду и наоборот:

***ord* (< символ >)** – возвращает код
символа ;

***chr* (< код >)** - возвращает символ по
указанному коду.

К аргументам символьного типа применяются **функции**, которые определяют предыдущий и последующий символы: - ***Pred*** и ***Succ***, например, $\text{Pred}('F') = 'E'$; $\text{Succ}('Y') = 'Z'$.

Для символов из интервала 'a'..'z' применима функция ***UpCase***(<символ>), которая **переводит строчные буквы в заглавные** (верхний регистр): 'A'..'Z'

Символьная константа может записываться в тексте программы двумя способами:

- 1) Как один символ, заключенный в апострофы;**
- 2) С помощью конструкции вида #K, где K - код соответствующего символа,**

Булевский (логический) тип

Величины булевского типа могут принимать два значения:

true – истина, **false** – ложь,

причем **false < true**

Над булевым типом определены следующие

логические операции:

- 1) логическое умножение (конъюнкция) – **and (и)**;
- 2) логическое сложение (дизъюнкция) – **or (или)**;
- 3) логическое отрицание – **not (не)**;
- 4) исключающее или - **xor**

Логические операции

a	b	a and b	a or b	a xor b	not a
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0