

Простые типы данных

Общие вопросы

Данные - это общие понятия для всего того, с чем оперирует ЭВМ. В языке данные характеризуются типом. **Тип данных** определяет множество значений, которые может принимать переменная. В языке имеют место следующие типы данных.



Логический тип (Boolean)

Величина логического типа – может принимать одно из двух значений *False* и *True*. Логические значения служат для проверки правильности условий и при сравнении величин.

False □ ложь

True □ истина

Для сравнения данных предусмотрены операции отношения:

< - меньше

<= - меньше или равно

= - равно

<> - не равно

> - больше

>= - больше или равно



*Над логич. данными допускаются след.
логические операции:*

Or(или) – дизъюнкция

And(и) – конъюнкция

Not – логическое «НЕ»

В стандарте языка логический тип определен так, что всегда выполняется: `false < true`.

Существуют предопределенные логические функции, которые дают логический результат.

Odd (I) – (где I - Integer) – функция дает результат `True` – нечетная I, и `False` – четная.

Eoln (F) (and of line) – проверка файла на конец строки.

Eof (F) (and of file) – проверка на конец файла.

Целочисленный тип (Integer)



Диапазон от -32768 до +32767

Значениями типа Integer являются элементы подмножества целых чисел.

Над данными можно выполнять:

* - умножение;

+ - сложение;

- - вычитание;

div – деление с отсечением (отбр. дробной части);

mod – остаток (получение целого остатка при делении целого на целое).

Тип `Integer` относится к целочисленным типам данных, они представлены в таблице:

Тип	Диапазон	Размер в байтах
<code>Byte</code>	0..255	1
<code>Word</code>	0..65535	2
<code>Integer</code>	-32768..32767	2
<code>Shortint</code>	-128..127	4
<code>Longint</code>	-2147483648..2147483647	1

В языке существует предопределенная константа **MaxInt**.
Значением является самое большое целое значение,
допустимое для данной ЭВМ (32767).

Целый результат дают и 4 предопределенные функции:

Abs(I) - абсолютное значение I.

Sqr(I) - целое значение I, возведенное в квадрат ($I \leq \text{MaxInt} \div I$).

Trunc(R) - (R-веществ.) результатом явл. целая часть R.

Round(R) - (R-веществ.) результат округленное целое.

$\text{Round}(R) = \begin{cases} \text{Trunc}(R + 0.5), & R \geq 0 \\ \text{Trunc}(R - 0.5), & R < 0 \end{cases}$

Если некоторое I-целое, то имеют место след. функции:

Succ(I) - результатом является следующее за I целое (I+1) значение.

Pred(I) - предыдущее перед I целое значение (I-1).

Тип Char (символьный)

Значениями типа Char явл. элементы конечного и упорядоченного множества символов. В любой ЭВС такое множество есть. С помощью этого множества происходит связь системы с внешним миром. Имеются на входных, и на выходных периферийных устройствах.

К сожалению, до сих пор не существует одного стандартного множества символов. В связи с этим для каждой системы количество символов и упорядоченность определяется при реализации. Значение типа Char обозначается одним символом, заключенным в апострофы.

В некоторых версиях в качестве символьных значений допускается использовать последовательность символов в апострофах и длина этой последовательности не должна превышать 256 символов.

Но несмотря на реализацию системы существуют минимальные допущения, которые справедливы для типа Char вне зависимости от реализации:

- десятичные цифры от '0' до '9' упорядочены с их числовым значением.
- прописные буквы от 'A' до 'Z' упорядочены в алфавитном порядке, но не обязательно следует одна за другой.
- строчные буквы от 'a' до 'z' упорядочены в алфавитном порядке, но не обязательно следуют одна за другой.

Множество символов связаны 2 предопределенными функциями:

Ord(Ch)- дает порядковый номер Ch в данном множестве.

Chr(I)- дает символ с порядковым номером=I.

Функции Ord и Chr обратны по отношению друг к другу.

Chr(Ord(Ch))=Ch

Ord(Chr(I))=I

Упорядоченность в множестве символов определяется так, что $c1 < c2$ тогда и только тогда, если $\text{Ord}(c1) < \text{Ord}(c2)$.

Для аргументов типа Char предопределенные функции **Pred** и **Succ** могут быть определены след. образом:

$$\text{Pred}(Ch) = \text{Chr}(\text{Ord}(Ch) - 1)$$
$$\text{Succ}(Ch) = \text{Chr}(\text{Ord}(Ch) + 1)$$

Со значениями типа Char можно использовать и операции отношения:

$$'A' < 'B', 'K' > 'C'$$


Вещественный тип Real

Значениями типа Real являются элементы подмножества действительных чисел.

Все операции, выполняемые над действительными величинами, явл. приближенными.

Точность этих операций зависит от конкретной реализации ЭВМ.

Вещественный тип относят к простому типу. К простому типу относят и ординальный тип.

Между вещественным и ординальным типом существует разница, поэтому вещественный тип и выделяют отдельно. Для ординального типа характерно наличие ординального номера. Понятие **ординальный номер** включает в себе то, что для данного значения существуют предшествующие и последующие значения.

Для вещественного типа не существует понятие предшествующего и последующего значения.

При условии, что хотя бы один из операндов имеет вещественный тип (другой- любой) след. операции дают вещественный результат.

* - умножение

/ - деление (даже если 2 целых- результат вещественный)

+ - сложение

- - вычитание

Существуют след. предопределенные функции, дающие вещественный результат при вещественном аргументе:

Abs(R)-абсолютное значение R

Sqr(R)- результат R^2 (результат не должен выходить за диапазон вещественных чисел 10⁻³⁸-10³⁸)

Следующие функции дают результат Real:

Sin(x), x -радианах

Cos(x), x -радианах

Arctan(x)- выражение дает значение угла в радианах

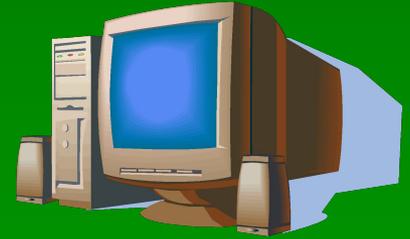
Ln(x)- значение натурального логарифма ($x > 0$)

Вещественные типы данных занимают от 4 до 10. Они могут быть как с плавающей (например, -3.2E-6, -6.42E+2), так и с фиксированной (например, 4.12, 6.05, -17.5489) точкой. Вещественные числа в формате с плавающей точкой представлены в экспоненциальной форме $mE\pm r$, где m - мантисса (целое или дробное число с десятичной точкой), r - порядок (целое число). Для того чтобы перевести число в экспоненциальной форме к обычному представлению с фиксированной точкой, необходимо мантиссу умножить на 10 в степени порядка.

Все вещественные типы данных приведены в таблице:

Тип	Диапазон	Мантисса	Размер в байтах
Real	2.9E-39..1.7E38	11-12	6
Single	1.5E-45..3.4E38	7-8	4
Double	5.0E-324..1.7E308	15-16	8
Extended	3.4E-4932..1.1E4932	19-20	10

Байтовый тип



Аналогичен целочисленному, но охватывает более узкий диапазон значений от 0 до 255. Описывается стандартным словом Byte.

Формат:

<идентификатор, ...>: Byte

Пример:

Var

Gran: byte;

Min, max: byte;

Переменная типа Byte занимает в памяти 1 байт, т.е. по сравнению с целочисленным типом память экономится в два раза. Использование этого типа целесообразно, если заранее известно значение переменной (≤ 255).