

Типы данных





Типы данных в языке Паскаль



Базовые и конструируемые типы

Базовые типы – типы, определяемые в языке программирования.

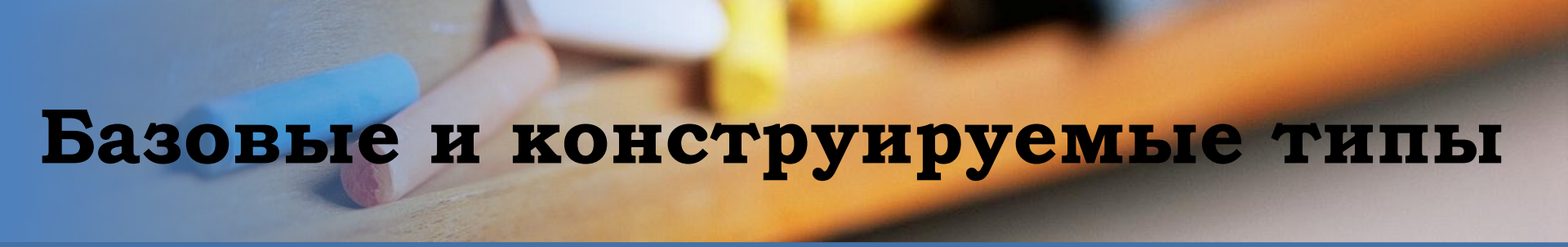
Конструируемые типы – типы, которые задаются программистом.

Базовые и конструируемые типы

Типы данных

Базовые

Конструируемые



Базовые и конструируемые типы

Например переменные базовых типов могут быть определены в разделе описания переменных

Var

```
a, b : real;  
d: integer;
```

Конструируемые типы так же могут быть описаны в разделе описания переменных

Var

```
s : string;
```

Раздел описания типов

Типы данных, конструируемые программистом, описываются в разделе Type по следующему шаблону:

Type

<ИМЯ_ТИПА> = <описание_типа>;

Например:

Type

lat_bukvy = 'a'..'z','A'..'Z';

Раздел описания типов

Базовые типы данных являются стандартными, поэтому нет нужды описывать их в разделе Type.

Однако при желании это тоже можно сделать, например, дав длинным определениям короткие имена. Скажем, введя новый тип данных

Type

```
int = integer;
```

Тогда можно описать переменные

Var

```
x, y : int;
```

Порядковые типы данных

Целые:

shortint

byte

integer

word

longint

Логические:

boolean

Символьные:

char;

Перечисляемые:

задаются перечислением значений и/или диапазонами значений.

Функции применяемые к порядковым типам

`ord(x)` возвращает порядковый номер значения переменной `x` (относительно того типа, к которому принадлежит переменная `x`).

`pred(x)` возвращает значение, предшествующее `x` (к первому элементу типа неприменима).

`succ(x)` возвращает значение, следующее за `x` (к последнему элементу типа неприменима).

Процедуры применяемые к порядковым типам

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

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

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

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

Целочисленные типы данных

Наименование типа	Формат	Область значений
byte	8 бит без знака	0..255
word	16 бит без знака	0..65535
shortint	8 бит со знаком	-128..127
integer	16 бит со знаком	-32768..32767
longint	32 бита со знаком	-2147483648.. 2147483647

Логический тип данных

Логический тип `boolean` имеет два значения:
`false` и `true`

Над операндами логического типа определены такие операции:

`or`, `and`, `not`, `xor`

Для логического типа выполняются следующие равенства:

`ord(false)=0`, `ord(true)=1`, `false<true`,
`pred(true)=false`, `succ(false)=true`,
`inc(true)=false`, `inc(false)=true`,
`dec(true)=false`, `dec(false)=true`.



Символьный тип данных

В символьный тип `char` входит 256 символов расширенной таблицы ASCII

Например,

`'a', 'b', 'я', '7', '&'`

Номер символа, возвращаемый функцией `ord()`, совпадает с номером этого символа в таблице ASCII.

Символьный тип данных

Пример описания символьной переменной:

```
Var
```

```
    simb1, simb2 : char;
```

```
Begin
```

```
    simb1:='R'; simb2:=#65; { С помощью # производится  
                           перевод целого числа в          соответствующий  
символ                      данного ASCII-кода }
```

```
    write (simb1,simb2);
```

```
End.
```

Результат работы программы

RA

Перечисляемые типы данных

Перечисляемые типы данных задаются в разделе `Type` явным перечислением их элементов.

Например:

`Type`

```
week =(sun,mon,tue,wed,thu,fri,sat)
```

Напомним, что для этого типа данных:

```
inc(sat) = sun, dec(sun) = sat.
```

Интервальные типы данных (диапазоны)

Интервальные типы данных задаются только границами своего диапазона.

Например:

Type

```
month = 1..12;
```

Программист может создавать и собственные типы данных, являющиеся комбинацией нескольких стандартных типов.

Например:

Type

```
valid_for_identifiers = 'a'..'z','A'..'Z','_', '0'..'9';
```

Вещественные типы данных

Тип данных	Десятичные разряды	Диапазон значений	Кол-во битов
single	7-8	$1.5E-45 \dots 3.4E38$	32
real	11-12	$2.9E-39 \dots 1.7E38$	48
double	15-16	$5.0E-324 \dots 1.7E308$	64
extended	19-20	$1.9E-4951 \dots 1.1E4932$	80
comp	19-20	$-9.2E18 \dots 9.2E18$	64

Вещественные типы данных являются арифметическими, но не порядковыми.

Следовательно для этих типов данных выполняются арифметические операции (за исключением операций с целыми числами) и стандартные математические функции.

Запись вещественных чисел

Математическая запись	Запись на Паскале
$4 \cdot 10^{-4}$	4E -4
$0,62 \cdot 10^5$	0.62E+5 либо .62E+5
$-10,88 \cdot 10^{12}$	-10.88E12

Конструируемые типы данных

Конструируемые

Массивы | Множества | Записи | Файлы | **Списки** | Задания | Строки | Процедуры | Объекты

Конструируемые типы данных будут изучены на последующих лекциях.



Совместимость типов данных

В общем случае при выполнении арифметических (и любых других) операций компилятору требуется, чтобы типы операндов совпадали.

Нельзя, например, сложить массив и множество, нельзя передать вещественное число переменной, ожидающей целый аргумент, и т.п.

В то же время, любая переменная, в расчете на вещественные значения, сможет работать и с целыми числами.

Неявное преобразование типов

Тип результата арифметических операций (а следовательно, и выражений) может отличаться от типов исходных операндов.

Пример:

Var

a,b : integer;

d : real;

Begin

read (a,b);

r:=a/b;

write (r);

End.

Неявное преобразование типов

Если в некоторой операции присваивания участвуют два типа данных совместимых, но не совместимых по присваиванию, то тип присваиваемого выражения автоматически заменяется на подходящий.

Пример:

Var

a : byte;

Begin

a:=10;

a:=-a;

write (a);

End.

На экране мы увидим не -10, а 246 ($246 = 256 - 10$).



Явное преобразование типов

Тип значения можно изменить и **явным** способом:
просто указав новый тип выражения.

Пример:

```
a:= byte(b);
```

В этом случае переменной *a* будет присвоено значение, полученное новой интерпретацией значения переменной *b*.

Скажем, если *b* имеет тип *shortint* и значение -23 ,
то в *a* запишется $233 (= 256 - 23)$.

Функции изменяющие тип данных

Функции округления:

trunc real -> integer

round real -> integer

Функция преобразования строки в число

val string -> byte/integer/real

Получение символа по заданному ASCII-коду

chr byte -> char

Преобразование порядковых типов

ord <порядковый_тип> -> longint