

Основные понятия языка Pascal



C:\ D:\TP\BIN\TURBO.EXE



File Edit Search Run Compile Debug Tools Options Window Help

[] 36.PAS 1=[↑]

```
program primer;
uses crt;
var a,b: array [1..1000] of char;
    c: array [1..1000] of byte;
    h,m,s: array [1..3] of byte;
    d: array [1..2] of byte;
    i,n1,n2,j,day: integer;
begin
clrscr;
assign(input,'input36.txt');
reset(input);
while not eoln do begin inc(i); read(a[i]); end; n1:=i; i:=0; readln;
while not eoln do begin inc(i); read(b[i]); end; n2:=i;
close(input);
for i:=1 to n1 do c[i]:=ord(a[i])-ord('0');
for i:=1 to 2 do h[i]:=h[i]*10+c[i];
for i:=4 to 5 do m[i]:=m[i]*10+c[i];
for i:=7 to 8 do s[i]:=s[i]*10+c[i];
for i:=1 to n2 do c[i]:=ord(b[i])-ord('0');
for i:=1 to n2 do if b[i]=':' then begin inc(j); d[j]:=i; end;
if j=0 then for i:=1 to n2 do s[2]:=s[2]*10+c[i];
```

1:1

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Алфавит



Простые слова



Законы построения фраз

Программа состоит из:

- **Исполняемый оператор** задаёт законченное действие, выполняемое над данными
(Примеры: вывод на экран, занесения числа в память, выход из программы)
- **Оператор описания** – описывает данные, над которыми в программе выполняются действия
(Пример: предложение «В памяти следует отвести место для хранения целого числа, и это место мы будем обозначать А»)

```
var a,b,sum : integer;
```

{1}

```
begin
```

{2}

```
  readln(a,b);
```

{3}

```
  sum := a+b;
```

{4}

```
  writeln('Сумма чисел', a, 'и', b , 'равна',
```

{5}

```
sum);
```

{6}

```
end.
```

Алфавит языка Pascal:

- Прописные и строчные латинские буквы, знак подчёркивания `_`;
- Цифры от 0 до 9;
- Специальные символы (`+`, `*`, `{`, `,`, `@`);
- Пробельные символы: пробел, табуляция, и переход на новую строку;



Лексемы:

- Константы
- Имена (идентификаторы)
- Ключевые слова
- Знаки операции
- Разделители (скобки , точка, запятая, пробельные символы)

Константа

- это величина, не изменяющая своё значение в процессе работы программы

Константы

Целые		Вещественные		Символьные	Строковые
Десятичные	Шестнадцатеричные	С плавающей точкой	С порядком		
2, 10, 125, -215	\$0101 \$F125 \$FFA4	15.0012 -3.14 .1023	1.2e4 0.1E-5	'k' #186 ^M	'привет!' 'I'm fine'

- ***Вещественные константы***
записываются с точкой перед дробной частью. Либо целая, либо дробная часть может отсутствовать.
- Вещественная константа состоит из ***мантиссы и порядка***:
- Мантисса записывается слева от знака 'E' или 'e', порядок – справа от этого знака

Пример записи: $E=1.2e4=1,2e*10^4=12000$

Символьные константы

записываются в трёх основных формах:

- Символ, заключенный в апострофы
- Десятичный код символа со знаком #.
(Применяется для символов отсутствующих на клавиатуре)
- Буква со знаком ^ . Используется для представления управляющих символов.
Код буквы должен быть на 64 больше, чем код представляемого таким образом символа.

- ***Строковая константа*** – это последовательность любых ASCII – символов, расположенная на одной строке и заключенная в апострофы
- Представление апострофа – это его дублирование (“ ”)
- Максимальная длина строковой константы – **126 символов**

Имена (идентификаторы)

- Имена даются элементам программы: переменным, константам, процедурам, функциям и др.

Имена задаёт программист:

- Имя должно начинаться с буквы (или знака подчеркивания)
- Имя должно содержать только буквы, знак подчеркивания и цифры
- Прописные и строчные буквы не различаются
- Длина имени практически неограниченна (значимыми являются только 63 первых символа)

Vasia, A, A13, A_and_B и _____

21ate, Big gig, Сюр

- ***Ключевые (зарезервированные слова)***
– это идентификаторы, имеющие специальное значение для компилятора. Имена, задаваемые программистом, не должны совпадать с ключевыми словами
- ***Специальные символы, или сочетания***
(~20 шт.)

Тип данных определяет:

- Внутреннее представление данных, диапазон возможных значений
- Допустимые действия над ними (операции и функции)

Примеры отличия разных типов:

- **целые** и **вещественные** числа имеют одинаковый объём памяти, но – совершенно разные диапазоны возможных значений
- **целые** числа можно умножать друг на друга, а **символьные** нет

$$5 * 5 = 25$$

$$'?' * '!' = \underline{\hspace{2cm}}$$

Классификация типов данных:

Стандартные	Определяемые программистом	
Логические Целые Вещественные Символьный Строковый Адресный Файловые	Простые	Составные
	Перечисляемый Интервальный Адресные	Массивы Строки Записи Множества Файлы Процедурные типы Объекты

Стандартные типы данных

- **1. Логические типы**

Внутреннее представление:

- Основной тип логических данных называется **Boolean**
- Величины этого типа данных занимают – 1 байт, и могут принимать всего два значения – true (истина) и false (ложь). Внутренне представление значения **false – 0, true – 1**

- Другие логические типы данных: ByteBool, WordBool, LongBool – длиной 1,2,4 байта. Истинным в них является любое значение не равное 0

Операции:

1) Логические операции: and, or, xor и not

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

- **Таблица истинности** – в ней приведены все возможные сочетания значений аргументов и соответствующие им значения результата.

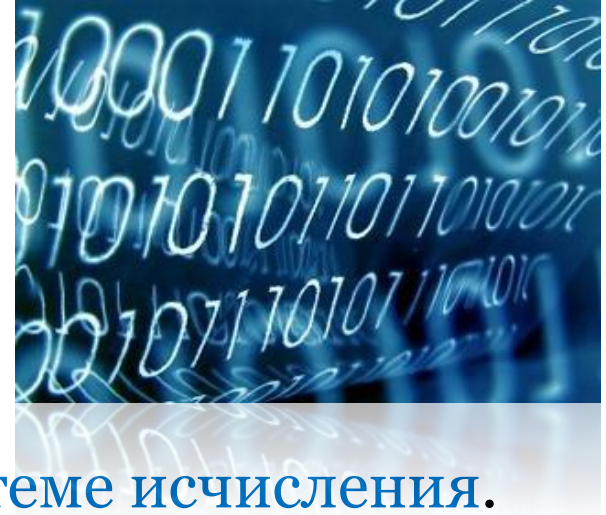
2) Операции отношения

- Результат этих операций – логический тип

Пример: `false < true` – значение `true`, `false = true` – значение `false`.

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

2. Целые типы



- Внутреннее представление:
- Представляются в **двоичной системе исчисления**.
- В Паскале определены несколько типов данных, отличающиеся **длиной и наличием знака**: старший двоичный разряд либо воспринимается как знаковый, либо является обычным разрядом числа
- Внутреннее представление определяет диапазоны допустимых значений величин (от нулей во всех двоичных разрядах до единиц).

Тип	Название	Размер	Знак	Диапазон значений
Integer	Целое	2 байта	Есть	-32 768 ... 32 767
Shortint	Короткое целое	1 байт	Есть	-128 ... 127
Byte	Байт	1 байт	Нет	0 ... 255
Word	Слово	2 байта	Нет	0 ... 65 535
Longint	Длинное целое	4 байта	Есть	-2 147 483 648 ... 2 147 483 647

Операции над целыми величинами:

- Арифметические операции – Результат их выполнения всегда целый (при делении дробная часть отбрасывается): +, -, *, div, mod
- Операции отношения – Результат этих операций имеет логический тип (true, false)
- Порядные операции: and, or, xor, not

- **Пример подразрядной операции:**

3 and 2 – значение 2,
так как в двоичной системе 3 – это 11,
а 2 – это 10

- Операции сдвига влево **shl** и вправо **shr**

Пример:

- `12 shr 2` – значение 3, поскольку двоичное представление числа 12-1100
- Выполнив операцию `12 shl 1`, то есть сдвинув это число влево на 1 разряд, получим 24
- Освободившиеся при сдвиге влево разряды заполняются нулями, а при сдвиге вправо – знаковым разрядом.

Стандартные функции и процедуры:

Имя	Описание	Результат	Пояснения
Abs	Модуль	Целый	$ x $ записывается abs(x)
Arctan	Арктангенс угла	Вещественный	Arctg x записывается arctan(x)
Cos	Косинус угла	Вещественный	Cos x записывается cos(x)
Exp	Экспонента	Вещественный	e^x записывается exp(x)
In	Натуральный логарифм	Вещественный	$\text{Log}_e x$ записывается ln(x)
Odd	Проверка на четность	Логический	Odd(3) даст в результате true
Pred	Предыдущее число	Целый	Pred(3) даст в результате 2

Имя	Описание	Результат	Пояснения
Sin	Синус угла	Вещественный	Sin x записывается sin(x)
Sqr	Квадрат	Целый	X ² записывается sqrt(x)
Sqrt	Квадратный корень	Вещественный	Записывается sqrt(x)
Succ	Следующее значение	Целый	Succ(3) даёт в результате 4

Процедуры

Inc	Инкремент		Inc(x) – увеличить x на 1 Inc(x,3) – увеличить x на 3
Dec	Декремент		Dec(x) – уменьшить x на 1 Dec(x,3) – уменьшить x на 3

- **3. Вещественные типы:**

Внутреннее представление:

состоит из двух частей – мантиссы и порядка, и каждая часть имеет знак. Точность числа определяется длиной мантиссы, а диапазон – длиной порядка.

Тип	Название	Размер, байт	Значащих цифр	Диапазон значений
Real	Вещественный	6	11-12	$2,9e-39 .. 1,7e+38$
Single	Одинарной точности	4	7-8	$1,5e-45 .. 3.4e+38$
Double	Двойной точности	8	15-16	$5.0e-324 .. 1,7e+308$
Extended	Расширенный	10	19-20	$3,4e-4932 .. 1.1e+4923$
Comp	Большое целое	8	19-20	$-9.22e18 .. 9.22e18$

Операции над вещественными типами:

- Арифметические – результат их выполнения вещественный тип: +, -, *, /
(Целые и вещественные операнды разрешается смешивать)
- Операции отношения – результат их выполнения логический тип: =, <>, >, <, <=, >=
- Стандартные функции: abs, arctan, cos, exp, ln, sin, sqr, sqrt

Имя	Описание	Результат	Пояснение
Frac	Дробная часть	Вещественный	$\text{Frac}(2,5)=0,5$
Int	Целая часть	Вещественный	$\text{Int}(2,5)=2,0$
Pi	π	Вещественный	3,1415926536
Round	Округление до целого	Целое	$\text{Round}(3,8)=4$ $\text{Round}(3,1)=3$
Trunk	Целая часть	Целое	$\text{Trunk}(3,8)=3$

• 4. Символьный тип

Внутреннее представление:

- Этот тип данных, обозначаемый ключевым словом `char`, служит для представления любого символа из набора допустимых символов.
- Под каждым символом отводится 1 байт.

Операции:

- Операции отношения ($<$, $>$, $<=$, $>=$, $<>$), при этом сравнивают коды символов.
- Стандартные функции

Имя	Описание	Результат	Пояснение
Ord	Порядковый номер числа	Целое	Ord('b') = 98
Chr	Преобразование в символ	Символьный	Chr(98)='b'
Pred	Предыдущий по таблице	Символьный	Pred('b')='a'
Succ	Последующий символ	Символьный	Succ('b')='c'
Uppcase	Перевод в верхний регистр	Символьный	Uppcase('b')='B'

- **5. Порядковые типы:**

К ним относятся: Целый тип, Символьный, Логический, Перечисляемый и Интервальный.

Они обладают общими чертами

- Все возможные значения порядкового типа представляют собой ограниченное упорядоченное множество
- К любому порядковому типу может быть применена стандартная функция **Ord**, которая в качестве результата возвращает порядковый номер конкретного значения в данном типе.
- К любому порядковому типу могут быть применены стандартные функции **Pred** и **Succ**, которые возвращают предыдущее и последующие значения соответственно;
- К любому порядковому типу могут быть применены стандартные функции **Low** и **High**, которые возвращают наименьшее и наибольшее значения величин данного типа

Приведение типов

Например:

- Integer ('A')
- Byte (500)
- Размер преобразуемой величины должен быть равен числу байтов, отводимых под тип, в который она преобразуется

Домашнее задание

Законспектировать:

- Ключевые зарезервированные слова
- Специальные символы языка Pascal