



M. C. E S C H E R.

Основные типы и операторы

О чем презентация?

- Типы данных: **целые, вещественные, булевский**
- Выражения: **арифметические и логические**
- Операторы: **ввод-вывод (чтение, запись) и присваивание**
- Программы без условных операторов и циклов

Это - фундамент программирования. Очень важно уметь этим минимумом пользоваться.

Целые типы (1)

Это группа типов обозначает множества целых чисел в различных диапазонах.

Тип **integer** (целый).

Диапазон: -32768..32767.

Размер памяти: 2 байта (16 двоичных разрядов).

Над целыми значениями допустимы следующие операции:
арифметические бинарные (два операнда)

- сложение (знак операции **+**) - результат целое число
- умножение (знак операции *****) - результат целое число
- вычитание (знак операции **-**) - результат целое число
- деление (знак операции **/**) - результат вещественное число
- целочисленное деление (знак операции **div**) - результат целое число
- взятие остатка от целочисленного деления (знак операции **mod**) - результат целое число

Целые типы (2)

арифметическая унарная (одноместная) операция

- изменение знака (знак операции **-**) - результат целое число

операции сравнения (результат операции есть величина логического (булевского) типа **true** или **false**):

- равно (**=**)
- не равно (**<>**)
- больше (**>**)
- больше или равно (**>=**)
- меньше (**<**)
- меньше или равно (**<=**)

Целые типы (3)

стандартные функции (в обычной функциональной записи)

- **абсолютная величина** (например, **abs**(-4)=4)
- **квадрат** (например, **sqr**(6)=36)
- Логическая функция **odd**(<целое>) (odd – нечетный) – результат true, если аргумент есть нечетное число (например, odd(6) = false)

Остальные целые типы в Паскале:

- **Shortint** 8-битный целый со знаком -128..127
- **Longint** 32-битный целый со знаком -2147483648..2147483647
- **Byte** 8-битный целый без знака 0..255
- **Word** 16-битный целый без знака 0..65535

Вещественные типы (1)

Обозначают множества вещественных чисел в различных диапазонах.

Тип **real** (**вещественный**).

Диапазон значений: 2.9×10^{-39} .. 1.7×10^{38} .

Размер памяти – 6 байт.

Вещественные значения могут изображаться в форме с **фиксированной точкой** и в форме с **плавающей точкой**.

В первом случае целая и дробная части вещественного числа отделяются друг от друга символом «.» (точка). Обе эти части должны обязательно присутствовать, например,

- 17.384
- 0.5

Следующие примеры демонстрируют неправильные формы записи чисел:

- .3 (правильно 0.3)
- 10. (правильно 10.0)

Вещественные типы (2)

Вещественное число в форме с плавающей точкой записывается как пара вида **<мантисса> E <порядок>** или **<мантисса> e <порядок>**.

- Число цифр мантиссы: 11–12.

Такое обозначение понимается как «мантисса, умноженная на 10 в степени, равной порядку».

Например, 7E–2 означает 7×10^{-2} ,

Обе записи

12.25E+6 или 12.25E6

обозначают 12.25×10^6 .

Мантисса представляется в виде целого числа или как вещественное с фиксированной точкой.

Порядок обозначается целым числом; допускаются как положительные, так и отрицательные значения порядка.

Вещественные типы (2)

Над вещественными значениями допустимы следующие операции: арифметические операции

- (сложение, умножение, вычитание, деление, изменение знака) - операнды могут быть или оба вещественные или один из них целый; результат всегда вещественный

- операции сравнения - результат булевский (логический)

стандартные функции

- **sqrt(x)** (квадратный корень)
- **sin(x)** (синус)
- **cos(x)** (косинус)
- **arctan(x)** (арктангенс)
- **ln(x)** (натуральный логарифм)
- **exp(x)** (экспонента, e^x)

– для этих функций аргумент может быть вещественным или целым, результат – всегда вещественный

- **abs(x)** (абсолютная величина числа)
- **sqr(x)** (квадрат) – тип результата совпадает с типом аргумента

Вещественные типы

(3)

Операции округления

- **trunc(x)**

x – вещественное или целое, результат – целая часть числа x, дробная часть отбрасывается и не округляется

$\text{trunc}(2.718) = 2$

$\text{trunc}(-2.718) = -2$

$\text{trunc}(5) = 5$

- Функция округления **round(x)**

x – вещественное или целое, результат – округленное целое,

round(x)=trunc(x+0.5) при $x \geq 0$,

round(x)=trunc(x-0.5) при $x < 0$.

Примеры применения функции **round**:

$\text{round}(5.2) = 5 = \text{trunc}(5.2)$

$\text{round}(5.500001) = 6 \neq \text{trunc}(5.500001)$

$\text{round}(-5.2) = -5 = \text{trunc}(5.2)$

$\text{round}(-5.500001) = -6 \neq \text{trunc}(-5.500001)$

Ввод-вывод (1)

Для выполнения операций ввода–вывода служат четыре процедуры: **read**, **readln**, **write**, **writeln**.

Процедура чтения **read** обеспечивает ввод числовых данных, символов и некоторых других типов данных с клавиатуры, в результате чего эти данные становятся значениями переменных.

Вызов процедуры: **read(X1,X2,...,XN);**

где **X1,X2,...,XN** – переменные допустимых типов данных.

Значения **X1, X2, ..., XN** набираются минимум через один пробел на клавиатуре и высвечиваются на экране. После набора данных для одной процедуры **read** нажимается клавиша ввода **<Enter>**. Значения переменных должны вводиться в строгом соответствии с синтаксисом языка Паскаль. Если соответствие нарушено (например, **X1** имеет тип **integer**, а при вводе набирается значение нечислового типа **char**), то возникают ошибки ввода–вывода.

Ввод-вывод (2)

- Если в программе имеется несколько подряд идущих процедур **read**, данные для них вводятся **потоком**, т. е. после набора значений переменных для одной процедуры **read** можно не нажимая клавишу <Enter> продолжить набор данных для следующей процедуры в той же строке и только потом нажать клавишу ввода.
- Процедура чтения **readln** аналогична процедуре **read**, единственное отличие заключается в том, что после считывания последней переменной в списке значений для одной процедуры **readln** данные для следующей процедуры ввода будут считываться с начала новой строки (т. е. ввод потоком невозможен).

Схема программы

const

<описание констант>

type

<описание типов>

var

<описание переменных>

begin

<операторы>

end.

В программах могут отсутствовать некоторые разделы описания.

Если нет описания переменных, то выполнение программы может выдавать только печать одного и того же текста.

Задача 1

(1)

- **Задача.** Написать программу, которая вводит целое число и печатает это число вместе с его квадратом.

Задача 1

(2)

Программа (версия 1)

```
var a:integer;
begin
  write('input number');
  read(a);
  write(a);
  write(a*a)
end.
```

```
F:\TP\BIN>TURBO.EXE
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
input number5
525
```

Программа имеет недостатки: нет пробела после слова «number» и между значениями a и a*a

Задача 1

(3)

Программа (версия 2)

```
var a:integer;  
begin  
  writeln;  
  writeln('input number');  
  read(a);  
  writeln(a);  
  writeln(a*a)  
end.
```

```
input number  
5  
5  
25
```

Недостатки версии 1 программы **исправлены**.

Задача 1

(4)

Программа (версия 3)

```
var a:integer;  
begin  
  writeln;  
  write('input number: ');  
  read(a);  
  writeln(a);  
  writeln(a*a)  
end.
```

```
input number: 5  
5  
25
```

Небольшое **«косметическое» улучшение** версии 2.

Задача 2

(1)

- **Задача.** Написать программу, которая для заданного вещественного числа a печатает следующую таблицу:

a

a^3 a^3

a^6 a^3 a

Задача 2

(2)

Программа (версия 1)

```
var a, b:real;
begin
  writeln;
  write('input number: ');
  read(a);
  b:=a*a*a;
  writeln(a);
  writeln(b,' ',b);
  writeln(b*b,' ',b,' ',a)
end.
```

```
input number: 5
5.0000000000E+00
1.2500000000E+02    1.2500000000E+02
1.5625000000E+04    1.2500000000E+02    5.0000000000E+00
```

Задача 2 (3)

Программа (версия 2)

```
var a, b:real;
begin
  writeln;
  write('input number: ');
  read(a);
  b:=a*a*a;
  writeln(a:8:2);
  writeln(b:8:2,b:8:2);
  writeln(b*b:8:2,b:8:2,a:8:2)
end.
```

```
input number: 5
  5.00
 125.00 125.00
15625.00 125.00  5.00
```

Используем форматированный вывод вещественных чисел: задаем размер поля вывода числа и количество цифр в дробной части числа.

Операции, связанные с делением чисел (1)

Вещественное деление

<операнд> / <операнд>

Результат операции деления принадлежит вещественному типу:

- `writeln(5.7 / 3.5) ⇒ 1.6285714286E+00`
- `writeln(5.7 / 3.5 :6:4) ⇒ 1.6286`
- `writeln(5.7 / 2 :6:4) ⇒ 2.8500`
- `writeln(4 / 2) ⇒ 2.0000000000E+00`
- `writeln(4 / 2 :6:4) ⇒ 2.0000`
- `writeln((5+7) / (3*2)) ⇒ 2.0000`
- `writeln(4 / 0) ⇒ ошибка (деление на 0)`

Операции, связанные с делением чисел (2)

Целочисленное деление

<целый операнд> **div** <целый операнд>

$x \text{ div } y = \text{trunc}(x/y)$

Деление нацело с отбрасыванием целой части;
операнды и результат всегда целые.

- $5.7 \text{ div } 3.5$
 - $5.7 \text{ div } 2$
 - $(4/2) \text{ div } 2$
- } ошибка: типы операндов не соответствуют оператору
- $4 \text{ div } 0 \Rightarrow$ ошибка: деление на 0
 - $25 \text{ div } 3 = 8$
 - $5 \text{ div } 1 = 5$
 - $14 \text{ div } 4 = 3$
 - $-14 \text{ div } 4 = -3$
 - $14 \text{ div } -4 = -3$
 - $-14 \text{ div } -4 = 3$

Операции, связанные с делением чисел (3)

Остаток от целочисленного деления

<целый операнд> **mod** <целый операнд>

$x \text{ mod } y = x - y*(x \text{ div } y)$

операнды и результат всегда целые.

- $5.7 \text{ mod } 3.5$
 - $5.7 \text{ mod } 1$
 - $(4/2) \text{ mod } 2$
 - $4 \text{ mod } 0 \Rightarrow$ ошибка: деление на 0
- } ошибка: типы операндов не соответствуют оператору
- $25 \text{ mod } 3 = 1$
 - $5 \text{ mod } 1 = 0$
 - $14 \text{ mod } 4 = 2$
 - $-14 \text{ mod } 4 = -2$
 - $14 \text{ mod } -4 = 2$
 - $-17 \text{ mod } 4 = -1$
 - $-15 \text{ mod } 4 = -3$

Программы только с числовыми типами (1)

Задача 1. Поменять местами значения целых переменных x и y , не используя дополнительные переменные. (Оранжевым цветом выделены комментарии.)

```
var
  x,y:integer;
begin
  writeln('input x,y');
  readln(x,y);
  {x=a, y=b}
  x:=x+y;
  {x=a+b, y=b}
  y:=x-y;
  {x=a+b, y=a}
  x:=x-y;
  {x=b, y=a}
  writeln('x=',x,' y=',y)
end.
```

Программы только с числовыми типами (2)

Задача 2. Целой переменной s присвоить сумму цифр трехзначного целого числа k .

```
var s,k,a,b,c:integer;
begin
  writeln('Введите трехзначное натуральное число');
  readln(k);
  a:= k mod 10;
  b:= (k mod 100) div 10;
  c:= k div 100;
  {k=100c+10b+a}
  s:=a+b+c;
  writeln('сумма цифр числа ',k,' = ',s)
end.
```

Так будет выглядеть сеанс работы программы:

Введите трехзначное натуральное число

372

сумма цифр числа 372 = 12

Программы только с числовыми типами (3)

Задача 3. Присвоить целой переменной d первую цифру из дробной части положительного вещественного числа x (так, если $x=32.597$, то $d=5$).

```
var x,y:real; k,d:integer;
begin
  writeln('Введите вещественное число');
  readln(x); {x=32.597}
  y:=10*x; {y=325.97}
  k:=trunc(y); {k=325}
  d:= k mod 10; {d=5}
  writeln('первая цифра из дробной части = ',d)
end.
```

Более краткий текст программы:

```
var x:real; d:integer;
begin
  writeln('Введите вещественное число');
  readln(x);
  d:= trunc(10*x) mod 10;
  writeln('первая цифра из дробной части = ',d)
end.
```

Программы только с числовыми типами (4)

Задача 4. Идет k -ая секунда суток. Определить, сколько полных часов (h) и полных минут (m) прошло к этому моменту. Например, если $k = 13257 = 3 \cdot 3600 + 40 \cdot 60 + 57$, то $h=3$ и $m=40$.

```
var
  k:longint;
  h,m:integer;
begin
  writeln('input seconds');
  readln(k); {k=13257}
  h:=k div 3600; {h=3}
  m:=(k mod 3600) div 60; {m=40}
  writeln('h=',h,' m=',m)
end.
```

Программы только с числовыми типами (5)

Задача 5. Определить f – угол (в градусах) между положением часовой стрелки в начале суток и ее положением в h часов, m минут и s секунд ($0 \leq h \leq 11$, $0 \leq m$, $s \leq 59$).

```
var
  f:real;
  h,m,s:longint;
  t:longint;
begin
  writeln('input h, m, s ');
  readln(h,m,s);
  t:=s+60*m+3600*h;
  {t – количество секунд от начала суток до данного момента}
  f:=t/120; {угол в один градус соответствует 120 секундам времени}
  writeln('f=',f:8:3)
end.
```

Булевский (логический) тип (1)

- Стандартное имя типа – **boolean**.
Используется для представлений значений двузначной логики:
true (истина), **false** (ложь).
Размер памяти для значений – 1 байт.
Логический тип упорядочен: **true > false**.
- Функции **ord** (порядковый номер), **pred** (предыдущее значение), **succ** (следующее значение)
ord(false)=0
ord(true)=1
pred(true)=false
succ(false)=true
- Используются операции сравнения, например,
true < false = false.

Логические операции

- Операции: **and**, **or**, **not**.
Аргументы (операнды) – булевский тип, результат – булевский тип.
- Операции **and** и **or** – *бинарные коммутативные и ассоциативные*:
- **x and y** – **логическое умножение** (логическое "и") или **конъюнкция** – значение **true** тогда и только тогда, когда оба операнда – **true**;
- **x or y** – **логическое сложение** (логическое "или") или **дизъюнкция** – значение **false** тогда и только тогда, когда оба операнда **false**.
- Операция **not x** – *унарная операция* – **логическое отрицание**: **not true = false**, **not false = true**

Вычислить значения выражений

$k \bmod 7 = k \operatorname{div} 5 - 1$ при $k=15$
false

$\operatorname{trunc}(10 * p) \bmod 2 = 1$ при $p=0.182$
true

$\operatorname{not} (x * y <> 0) \operatorname{and} (y > x)$ при $x=2, y=1$
false

$(x * y <> 0) \operatorname{or} (y > x)$ при $x=2, y=1$
true

$a \operatorname{or} (\operatorname{not} b)$ при $a=\operatorname{false}, b=\operatorname{true}$
false

$a \operatorname{or} b \operatorname{and} \operatorname{not} a$ при $a=\operatorname{true}, b=\operatorname{false}$
true

$(a \operatorname{or} b) \operatorname{and} \operatorname{not} a$ при $a=\operatorname{true}, b=\operatorname{false}$
false

$\operatorname{not} a \operatorname{and} \operatorname{not} b$ при $a=\operatorname{true}, b=\operatorname{false}$
false

$\operatorname{not} (a \operatorname{and} b)$ при $a=\operatorname{true}, b=\operatorname{false}$
true

Записать на Паскале выражение, истинное при выполнении указанного условия и ложное в противном случае (1)

$0 < x < 1$

$(x > 0) \text{ and } (x < 1)$

$x = \max(x, y, z)$

$(x \geq y) \text{ and } (x \geq z)$

хотя бы одна из логических переменных a и b имеет значение *true*

$a \text{ or } b$

обе логические переменные a и b имеют значение *true*

$a \text{ and } b$

уравнение $ax^2 + bx + c = 0$ не имеет вещественных корней

$\text{sqr}(b) - 4 * a * c < 0$

Записать на Паскале выражение, истинное при выполнении указанного условия и ложное в противном случае (2)

натуральное n является полным квадратом

$\text{sqr}(\text{trunc}(\text{sqr}(n))) = n$

x лежит вне отрезка $[0, 1]$

$(x < 0) \text{ or } (x > 1)$

x принадлежит отрезку $[2, 5]$ или $[-1, 1]$

$((x \geq 2) \text{ and } (5 \geq x)) \text{ or } ((x \geq -1) \text{ and } (1 \geq x))$

каждое из чисел x , y и z положительно

$(x > 0) \text{ and } (y > 0) \text{ and } (z > 0)$

хотя бы одно из чисел x , y и z положительно

$(x > 0) \text{ or } (y > 0) \text{ or } (z > 0)$

ни одно из чисел x , y и z не является положительным

$\text{not}((x > 0) \text{ or } (y > 0) \text{ or } (z > 0))$

только одно из чисел x , y и z положительно

$((x > 0) \text{ and } (y \leq 0) \text{ and } (z \leq 0)) \text{ or } ((y > 0) \text{ and } (x \leq 0) \text{ and } (z \leq 0)) \text{ or}$

$((z > 0) \text{ and } (x \leq 0) \text{ and } (y \leq 0))$

Записать на Паскале выражение, истинное при выполнении указанного условия и ложное в противном случае (3)

логическая переменная *a* имеет значение *true*, а логическая переменная *b* имеет значение *false*;

a and not *b*

Написать оператор присваивания, в результате выполнения которого логическая переменная *t* получает значение *true*, если выполняется указанное условие, и значение *false* иначе

числа *x*, *y* и *z* равны между собой

t := (*x*=*y*) and (*x*=*z*)

из чисел *x*, *y* и *z* только два равны между собой

t := ((*x*=*y*) and not(*x*=*z*)) or ((*x*=*z*) and not(*x*=*y*)) or ((*z*=*y*) and not(*x*=*z*))

цифра 5 входит в десятичную запись трехзначного числа *k*

t := (*k* mod 10 = 5) or (*k* div 100 = 5) or ((*k* mod 100) div 10) = 5

Записать на Паскале выражение, истинное при выполнении указанного условия и ложное в противном случае (4)

- год с порядковым номером *y* является високосным

Начиная с календаря Папы римского Григориуса (1752г.) сохраняется следующее правило для високосных годов (годы с 366 днями):

год, делимый на 4, - високосный год (например, 1972);

но: если он делится на 100, это не високосный год (например, 1900);

но: если он делится на 400, это – високосный год (например, 2000).

$(y \bmod 400 = 0) \text{ or } ((y \bmod 100 \neq 0) \text{ and } (y \bmod 4 = 0))$