



Увод в програмирането

Скаларни типове данни



Типове данни

- Вградени
- Абстрактни

Класификация на вградените ТИПОВЕ данни

■ Скаларни типове

- ☐ Булев
- ☐ Цял
- ☐ Реален
- ☐ Символен
- ☐ Изброим
- ☐ Указател
- ☐ Псевдоним

■ Съставни типове

- ☐ Масив
- ☐ Символен низ
- ☐ Вектор

Логически тип (Булев)

- Нарича се още булев тип в чест на Дж. Бул, английски логик, поставил основите на математическата логика.
- Типът е стандартен, вграден в реализацията. За означаването му се използва запазената дума **bool** (съкращение от `boolean`).
- **Множество от стойности**
Състои се от два елемента – стойностите `true` (истина) и `false` (лъжа). Тези стойности се наричат още **булеви константи**.
<булева_константа> ::= `true` | `false`.

Логически тип (Булев) ...

- Променлива величина, множеството от допустимите стойности, на която съвпада с множеството от стойности на типа булев, се нарича **булева** или **логическа променлива** или **променлива от тип булев**

- Примери

```
bool b1, b2;
```

```
bool b3 = false;
```

Логически тип (Булев) ...

Оператори

■ Логически оператори

- *Оператор за логическо умножение (конюнкция)*

Означава се с **&&**

A	B	A && B
false	false	false
false	true	false
true	false	false
true	true	true

Логически тип (Булев) ...

Оператори

- *Оператор за логическо събиране (дизюнкция)*

Означава се с ||

A	B	A B
false	false	false
false	true	true
true	false	true
true	true	true

Логически тип (Булев) ...

Оператори

- *Оператор за логическо отрицание*

Означава се с !

A	! A
false	true
true	false

Логически тип (Булев) ...

Оператори

■ Оператори за сравнение

Оператор Операция

==	сравнение за равно
!=	сравнение за различно
>	сравнение за по-голямо
>=	сравнение за по-голямо или равно
<	сравнение за по-малко
<=	сравнение за по-малко или равно

Логически тип (Булев) ...

Оператори

■ *Въвеждане*

Не е възможно въвеждане на стойност на булева променлива чрез оператора `>>`, т.е. операторът

```
cin >> b1;
```

■ *Извеждане*

Осъществява се чрез оператора

```
cout << <булева_константа>;
```

```
cout << <булев_израз>;
```

Логически тип (Булев) ...

Булеви изрази

- Булевите изрази са правила за получаване на булева стойност. Дефинират се *рекурсивно* по следния начин:
 - Булевите константи са булеви изрази.
 - Булевите променливи са булеви изрази.
 - Прилагането на булевите оператори !, &&, or || над булеви изрази е булев израз.
 - Прилагането на операторите за сравнение ==, !=, >, >=, <, <= към булеви изрази е булев израз

Числени типове

- **Целочислени типове**

- Ще разгледаме целочисления тип **int**.

Типът е стандартен, вграден в реализацията на езика. За означаването му се използва запазената дума **int** (съкращение от integer).

- **Множество от стойности**

Множеството от стойности на типа **int** зависи от хардуера и реализацията и не се дефинира от ANSI (American National Standards Institute).

Visual C++ 6.0: [-2 147 483 648, 2 147 483 647].

Целочислени типове

- Целите числа се записват като в математиката, т.е.
 $\langle \text{цяло_число} \rangle ::= [+ \mid -] \langle \text{цяло_без_знак} \rangle$
 $\langle \text{цяло_без_знак} \rangle ::= \langle \text{цифра} \rangle \mid$
 $\quad \langle \text{цифра} \rangle \langle \text{цяло_без_знак} \rangle$
 $\langle \text{цифра} \rangle ::= 0 \mid 1 \mid \dots \mid 9.$
- *Допълнение:* Целите числа могат да са в десетична, осмична и шестнадесетична позиционна система. Осмичните числа започват с 0 (нула), а шестнадесетичните - с 0x (нула, x).

Целочислени типове ...

- Елементите от множеството от стойности на типа `int` се наричат **константи от тип `int`**.
- Променлива величина, множеството от допустимите стойности, на която съвпада с множеството от стойности на типа `int`, се **нарича цяла променлива** или **променлива от тип `int`**
- Примери:

```
int i;
```

```
int j = 56;
```

Целочислени типове ...

Оператори

■ Аритметични оператори

□ Унарни оператори

Записват се пред или след единствения си аргумент.

+, **-** са префиксни оператори.

Примери:

```
int i = 12, j = -7;
```

```
-i    +j    -j    +i    -567
```

Целочислени типове ...

Оператори

- *Бинарни оператори*

Имат два аргумента. Следните аритметични оператори са инфиксни:

Оператор	Операция
+	събиране
-	изваждане
*	умножение
/	целочислено деление
%	остатък от целочислено деление

- *Примери:*

$$15 - 1235 = -1220$$

$$15 + 1235 = 1250$$

$$-15 * 123 = -1845$$

$$13 / 5 = 2$$

$$13 \% 5 = 3$$

$$23 \% 3 = 2$$

Целочислени типове ...

Оператори

■ Логически оператори

Логическите оператори, реализиращи конюнкция, дизюнкция и отрицание, могат да се прилагат над целочислени константи. Дефинират се по същия начин, както при булевите константи, но целите числа, които са различни от 0 се интерпретират true, а 0 – като false.

■ Примери:

123 && 0	е false
0 15	е true
! 67	е false

Целочислени типове ...

Оператори

■ Оператори за сравнение

Оператор	Операция
----------	----------

==	сравнение за равно
----	--------------------

!=	сравнение за различно
----	-----------------------

>	сравнение за по-голямо
---	------------------------

>=	сравнение за по-голямо или равно
----	----------------------------------

<	сравнение за по-малко
---	-----------------------

<=	сравнение за по-малко или равно
----	---------------------------------

■ Примери:

123 < 234	e true
-----------	--------

-123456 > 324	e false
---------------	---------

23451 >= 0	e true
------------	--------

Целочислени типове ...

Оператори

■ Поразредни оператори

Оператор	Операция
----------	----------

<<	преместване наляво
----	--------------------

>>	преместване надясно
----	---------------------

	поразрядно ИЛИ (дизюнкция)
--	----------------------------

^	поразрядно изключващо ИЛИ
---	---------------------------

&	поразрядно И (конюнкция)
---	--------------------------

~	инвертиране на битовете
---	-------------------------

■ Примери:

1 << 1	е 2
--------	-----

1 << 2	е 4
--------	-----

8 >> 3	е 1
--------	-----

127 128	е 255
-----------	-------

Целочислени типове ...

Вградени функции

- В езика C++ има набор от вградени функции. Обръщението към такива функции има следния синтаксис:
`<име_на_функция>(<израз>, <израз>, ..., <израз>)`
и връща стойност от типа на функцията
- Едноаргументна целочислена функция `abs`.
`abs(x)` – намира $|x|$, където x е цял израз
- *Примери:*
`abs(-1587) = 1587`
`abs(0) = 0`
`abs(23) = 23`

Целочислени типове ...

Въвеждане

- Реализира се по стандартния и разгледан вече начин.

- *Пример:*

```
int i, j;
```

```
cin >> i >> j;
```

Целочислени типове ...

Извеждане

- Реализира се чрез оператора

`cout << <цяла_константа>;`

или по-общо

`cout << <цял_израз>;`

- *Пример:*

```
int i = 1234, j = 9876;
```

```
cout << i << j << "\n";
```

Целочислени типове ...

Други целочислени типове

- Други цели типове се получават от **int** като се използват модификаторите **short**, **long**, **signed** и **unsigned**

Тип	Диапазон	Памет
short int	-32768 до 32767	2 байта
unsigned short int	0 до 65535	2 байта
long int	-2147483648 до 2147483647	4 байта
unsigned long int	0 до 4294967295	4 байта
unsigned int	0 до 4294967295	4 байта

Реални типове

- Ще разгледаме реалния тип **double**.

Типът е стандартен, вграден във всички реализации на езика.

- **Множество от стойности**

Множеството от стойности на типа **double** се състои от реалните числа от $-1.74 \cdot 10^{308}$ до $1.7 \cdot 10^{308}$.

Записват се във два формата – като числа с фиксирана и като числа с плаваща запетая (експоненциален формат).

Реални типове ...

$\langle \text{реално_число} \rangle ::= \langle \text{цяло_число} \rangle . \langle \text{цяло_число_без_знак} \rangle \mid$
 $\langle \text{цяло_число} \rangle \text{E} \langle \text{порядък} \rangle \mid$
 $\langle \text{цяло_число} \rangle . \langle \text{цяло_число_без_знак} \rangle \text{E} \langle \text{порядък} \rangle$
 $\langle \text{порядък} \rangle ::= \langle \text{цяло_число} \rangle$

Примери:

123.3454

-10343.034

123E13

-1.23e-4

Реални типове ...

- Елементите от множеството от стойности на типа **double** се наричат **реални константи** или по-точно **константи от реалния тип double**.
- Променлива величина, множеството от допустимите стойности, на която съвпада с множеството от стойности на типа **double**, се нарича **реална променлива** или **променлива от тип double**.

■ *Примери:*

```
double i;
```

```
double j = 5699.876;
```

Реални типове ...

Оператори

■ Аритметични оператори

- *Унарни оператори +, -*

Префиксни са. Потвърждават или променят знака на аргумента си.

- *Примери:*

```
double i = 1.2, j = -7.5;
```

```
-i      +j      -j      +i      -56.7
```

Реални типове ...

Оператори

■ Бинарни оператори

Имат два аргумента. Следните аритметични оператори са инфиксни:

Оператор	Операция
+	събиране
-	изваждане
*	умножение
/	деление

■ Примери:

$$15.3 - 12.2 = 3.1$$

$$15 + 12.35 = 27.35$$

$$-1.5 * 12.3 = -18.45$$

$$13.0 / 5 = 2.6$$

$$13 / 5.0 = 2.6$$

Реални типове ...

Оператори

■ Логически оператори

Логическите оператори, реализиращи операциите конюнкция, дизюнкция и отрицание, могат да се прилагат над реални константи. Дефинират се по същия начин, както при булевите константи, като реалните числа, които са различни от 0.0 се интерпретират като true, а 0.0 – като false.

■ *Примери:*

123.6 && 0.0	e false
0.0 15.67	e true
! 67.7	e false

Реални типове ...

Оператори

■ Оператори за сравнение

Оператор	Операция
----------	----------

==	сравнение за равно
----	--------------------

!=	сравнение за различно
----	-----------------------

>	сравнение за по-голямо
---	------------------------

>=	сравнение за по-голямо или равно
----	----------------------------------

<	сравнение за по-малко
---	-----------------------

<=	сравнение за по-малко или равно
----	---------------------------------

■ Примери:

123.56 < 234.09	e true
-----------------	--------

-123456.9888 > 324.0098	e false
-------------------------	---------

23451.6 >= 0	e true
--------------	--------

Реални типове ...

Вградени функции

- При цял или реален аргумент, следните функции връщат реален резултат от тип `double`

Функция

Намира

$\sin(x)$	синус, $\sin x$, x е в радиани
$\cos(x)$	косинус, $\cos x$, x е в радиани
$\tan(x)$	тангенс, $\operatorname{tg} x$, x е в радиани
$\operatorname{asin}(x)$	аркуссинус, $\arcsin x \in [-\pi/2, \pi/2]$, $x \in [-1, 1]$
$\operatorname{acos}(x)$	аркускосинус, $\arccos x \in [0, \pi]$, $x \in [-1, 1]$
$\operatorname{atan}(x)$	аркустангенс, $\operatorname{arctg} x \in (-\pi/2, \pi/2)$
$\exp(x)$	експонента, e^x
$\log(x)$	натурален логаритъм, $\ln x$, $x > 0$
$\log_{10}(x)$	десетичен логаритъм, $\lg x$, $x > 0$

Реални типове ...

Вградени функции

Функция	Намира
$\sinh(x)$	хиперболичен синус, sh x
$\cosh(x)$	хиперболичен косинус, ch x
$\tanh(x)$	хиперболичен тангенс, th x
$\text{ceil}(x)$	най-малкото цяло $\geq x$, преобразувано в тип double
$\text{floor}(x)$	най-голямото цяло $\leq x$, преобразувано в тип double
$\text{fabs}(x)$	абсолютна стойност на x, $ x $
$\text{sqrt}(x)$	корен квадратен от x, $x \geq 0$
$\text{pow}(x, n)$	степенуване, x^n (x и n са реални от тип double)

Реални типове ...

Вградени функции

■ *Примери:*

`ceil(12.345) = 13.0`

`ceil(-12.345) = -12.0`

`ceil(1234) = 1234.0`

`ceil(-1234) = -1234.0`

`floor(12.345) = 12.0`

`floor(-12.345) = -13.0`

`floor(123) = 123.0`

`floor(-123) = -123.0`

`fabs(123) = 123.0`

`fabs(-1234) = 1234.0`

`sin(PI/6)` намира `sin(30°)`, където `PI = 3.142857`

Реални типове ...

Вградени функции

```
#include <iostream.h>
#include <math.h>
int main()
{ cout << "x= ";
  double x;
  cin >> x;
  int y;
  y = floor(x);
  cout << y << "\n";
  return 0;
}
```

Преобразуване на типа

■ (тип)<израз>

където тип е име на тип.

Пресмята се стойността на <израз> и получената константа се преобразува в указания в скобите тип.

Примери:

`(int) (1.52 + 56.2) = 57`

`(double) (123+18) = 141.0`

Преобразуване на типа ...

■ `static_cast<тип>(<израз>)`

където тип е име на тип.

Пресмята се стойността на <израз> и получената константа се преобразува в указания в скобите < ... > тип.

Примери:

```
static_cast<int>(1.52 + 56.2) = 57
```

```
static_cast<double>(123+18) = 141.0
```

Реални типове ...

Въвеждане/извеждане

■ Въвеждане

Реализира се по стандартния начин.

■ *Пример:*

```
double x, y;  
cin >> x >> y;
```

■ Извеждане

Реализира се чрез оператора

```
cout << <реална_константа>;
```

```
cout << <реален_израз>;
```

Реални типове ...

Други реални типове

- В езика C++ има и друг реален тип, наречен float. Различава се от типа double по множеството от стойностите си и заеманата памет.
- Множеството от стойности на типа float се състои от реалните числа от диапазона от $-3.4 \cdot 10^{38}$ до $3.4 \cdot 10^{38}$.
- За записване на константите от този диапазон са необходими 4 байта ОП.
- Броят на значещите цифри при този тип е около 7.

Аритметични изрази

- Аритметичните изрази са правила за получаване на числови константи
- Има два вида аритметични изрази: цели и реални.

$\langle \text{аритметичен_израз} \rangle ::=$
 $\langle \text{цял_израз} \rangle \mid \langle \text{реален_израз} \rangle$

Цели аритметични изрази

- Целите аритметични изрази са правила за получаване на константи от тип `int` или разновидностите му:

- Целите константи са цели аритметични изрази

123	-2345	-32767
0	22233345	-87

- Целите променливи са цели аритметични изрази

```
int i, j;
```

```
short p, q, r;
```

```
i      j
```

```
p      q      r
```

- Прилагането на унарните оператори `+` и `-` към цели аритметични изрази е цял аритметичен израз

```
-i      +j      -j
```


Цели аритметични изрази ...

- Прилагането на бинарните аритметични оператори $+$, $-$, $*$, $/$ и $\%$ към цели аритметични изрази, е цял аритметичен израз

$i \% 10 + j * i$ $-p$ $-i + j / 5$
 $r - p / 12 - q$ $r \% q - p$ $r + p - q$

- Цялата функция `abs`, приложена над цял аритметичен израз, е цял аритметичен израз от тип `int`

`abs (i+j)` `abs (p-r)`

Реални аритметични изрази

- Реалните аритметични изрази са правила за получаване на константа от тип `double` или `float`

- Реалните константи са реални аритметични изрази.

`1.23e-3 -2345e2 -3.2767 0.0`

- Реалните променливи са реални аритметични изрази.

`double i, j;`

`float p, q, r;`

`i j`

`p q r`

- Прилагането на унарните оператори `+` или `-` към реален аритметичен израз е реален аритметичен израз.

`-i +j -j`

`+p -p +r -q`

Реални аритметични изрази ...

- Прилагането на бинарните аритметични оператори +, -, * и / към аритметични изрази, поне един от които е реален, е реален аритметичен израз.

$i/10 + j*i$	$-p$	$-i + j/5$
$-p/12 - q$	$r/q - p$	$r + p - q$

- Реалните функции: sin, cos, tan, asin, acos, atan, exp, log, log10, sinh, cosh, tanh, ceil, floor, fabs, sqrt и pow, приложени над реални или цели аритметични изрази, са реални аритметични изрази от тип double

<code>fabs(i+j)</code>	<code>sin(i-p)</code>	<code>cos(p/r-q)</code>	<code>floor(p)</code>
<code>ceil(r-p+i)</code>	<code>exp(p)</code>	<code>log(r-p*q)</code>	

Семантика на аритметичните изрази

- Аритметичният израз е правило за получаване на цяла или реална константа. Приоритет на операторите и вградените функции:

1. Вградени функции
2. Действията в скобите
3. Оператори в следния приоритет
 - $+$, $-$ (унарни) най-висок
 - $*$, $/$, $\%$
 - $+$, $-$ (бинарни)
 - $<<$, $>>$ най-нисък

```
double x = 23.56, y = -123.5;  
cout << sin(x) + ceil(y)*x - cos(y);
```

Семантика на аритметичните изрази ...

- Аритметичните изрази могат да съдържат операнди от различни типове. За да се пресметне стойността на такъв израз, автоматично се извършва **преобразуване на типовете** на операндите му

Тип	Преобразува се до
bool	ВСИЧКИ ЧИСЛОВИ ТИПОВЕ
short	int
unsigned short	unsigned int
float	double

Семантика на булевите изрази

- Булевите изрази са правила за получаване на булева стойност. Приоритет на операторите и вградените функции:

1. Вградени функции
2. Действията в скобите
3. Оператори в следния приоритет

- `!, +, -` (унарни)
- `*, /, %`
- `+, -` (бинарни)
- `>> <<` (вход/изход)
- `<, <=, >, >=`
- `==, !=`
- `&&`
- `||`

най-висок

най-нисък