

Увод в програмирането

Лекция 3:
УСЛОВИЯ
Част 1

Булев (Boolean) тип

- Само две стойности: false (0) и true (1)
- **bool b = true; b = false;**
- Използвана памет: един байт (не е бит, понеже адресите (на променливите) са цели числа)

Сравнение на числа

- Булеви изрази можем да получим, ако приложим операторите за сравнение върху числови изрази
 - $<$, $>$,
 \leq (по-малко или равно), \geq ,
 $==$ (равно), $!=$ (различно)
 - Примери: $1 < 2 \rightarrow \text{true}$, $x > x + 1 \rightarrow \text{false}^*$,
 $x != x \rightarrow \text{false}$
 - Ако по погрешка използваме $=$ вместо $==$, може и да не забележим, понеже синтактично е вярно
 - * Допълнителен материал: кога $x > x + 1$ не е false ?

Логически операции

- Можем да образуваме по-сложни булеви изрази, комбинирайки ги със следните операции:
 - Конюнкция (conjunction, AND, \wedge) – има стойност "истина" тогава и само тогава, когато стойностите на двата ѝ операнда също са "истина"
 - Дизюнкция (disjunction, OR, \vee) – има стойност "истина" т.с.т.к. поне един от двата ѝ операнда има стойност "истина"
 - Отрицание (negation, NOT, \neg) – сменя стойността на операнда си

Логически операции (2)

A	B	A AND B	A OR B	NOT A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Логически операции в C++

- В C++ се използват следните оператори:
 - && - конюнкция
 - || - дизюнкция
 - ! - отрицание
- Пример: `x > 1 || !(x > 1)`
- Приоритет: първо се смята !, после &&, накрая - ||
 - Лесен начин за запомняне: ако погледнем таблицата, && е умножение, а || прилича на събиране
- Внимание – има и оператори & и |

Условен оператор if

- Досега разглеждахме само линейни програми
- Не можехме да покриваме различни случаи
 - Напр. при решаване на квадратно уравнение
- С помощта на if може дадена част от кода да се изпълни само тогава, когато дадено условие е изпълнено
- Условието се задава с булев израз

Непълнен синтаксис на if

- Тъй като текущата тема е за булеви изрази, if ще бъде разгледан подробно по-късно
- Най-общ синтаксис:

if (<условие>)

 <оператор(statement)>

- Скобите около условието са задължителни.
- Пример:

```
int n = 4;
```

```
if (n % 2 == 0) cout << "4 е четно!";
```


Демонстрация

- Да се напише програма, която пресмята цената на дадена покупка по следния начин:

Закупени са определен брой еднакви продукти с определена единична цена.

Ако общата цена е над 100 лв, да се направи отстъпка от 10%.

- Да се провери дали x принадлежи на интервала

$(-\infty, -3] \cup [-2, -1) \cup [0, 2)$

Примерни булеви изрази

- Да се напише булев израз, който има стойност true тогава и само тогава, когато:
 - Цяло число n се дели на 7:
 $n \% 7 == 0$
 - Цялото число i е два пъти по-малко от цялото число j :
ГРЕШНО: $i == j / 2$ // целочислено деление!
ВЯРНО: $i * 2 == j$

По-сложни логически изрази

- При преобразуване на логически изрази може да използваме правилата на ДеМорган (и в двете посоки):
 - $\text{NOT (A AND B)} = \text{NOT A OR NOT B}$
 - Пример: не е вярно, че е студено и вали \Leftrightarrow не е студено или не вали (или и двете)
 - $\text{NOT (A OR B)} = \text{NOT A AND NOT B}$
 - $\text{NOT NOT A} = \text{A}$

По-сложни логически изрази (2)

- Нито едно от числата a , b и c не е положително:
 - $!(a > 0) \ \&\& \ !(b > 0) \ \&\& \ !(c > 0)$
 - $a \leq 0 \ \&\& \ b \leq 0 \ \&\& \ c \leq 0$
(Отрицанието на " $>$ " е " \leq ", а не " $<$ ")
 - $!(a > 0 \ || \ b > 0 \ || \ c > 0)$

По-сложни логически изрази (3)

- Пример: искаме да проверим дали дадена година е високосна
- Условието е по-трудно, отколкото си мислим: годината трябва да се дели на 4, но ако се дели на 100, трябва да се дели и на 400 (1900 не е такава)
- Имаме 3 възможности: да се дели на 4, да се дели на 100 и да се дели на 400

```
int year; cin >> year;  
bool y4 = year % 4 == 0,  
      y100 = year % 100 == 0,  
      y400 = year % 400 == 0;
```

- Виждаме, че има само 2 случая и построяваме израз по тях:

```
if (y4 && (y100 == y400))  
    cout << "Leap" << endl;  
  
// или (y4 && !y100) || y400
```

y4	y100	y400	leap
0	0	0	false
0	0	1	impossible
0	1	0	impossible
0	1	1	impossible
1	0	0	true
1	0	1	impossible
1	1	0	false
1	1	1	true

Преобразуване между булеви и числови изрази

- От булев към числов: `false` \rightarrow 0, `true` \rightarrow 1
 - `int a = (2 < 3.1);`
- От числов към булев: 0 \rightarrow `false`, всичко останало \rightarrow `true`
 - `if (!a) cout << "a is zero";`

Преобразуване между булеви и числови изрази (2)

- `int a = 5, b = 6;`
`if (a = b) cout << "Hmm";`
- Кодът след `if` се изпълни, а 5 е различно от 6 –
защо?!

Преобразуване между булеви и числови изрази (3)

- `int a = 5, b = 6;`
`if (a = b) cout << "Hmm";`
- Горното е еквивалентно на:
- `int a = 5, b = 6;`
`a = b;`
`if (a != 0) cout << "Hmm";`
- А всъщност искахме `if (a == b)`
- Това е много неприятна грешка, която може да допусне начинаещият програмист, а и не само

Упражнение:

четене на трудно четим код

- Какво ще се отпечата на екрана при изпълнение на следния код:

```
int a, b = 0; // каква е стойността на a?  
cout << ((1 + false) && !0) +  
        ((a = b == 1) && (a = b = 1)) *  
        (48.392 * 8904 * sin(-47832) - atan(45809));
```

Сравняване на числа с плаваща запетая

- `double x = 10.0;`
`double y = sqrt(x);`
`y = y * y;`
`if (x == y) cout << "You can't see me";`
- Ако погледнем с debugger каква е стойността на `y`, ще видим, че е много близо до 10, но не точно
- В такива случаи може да се допуска известно отклонение, напр. 10^{-14}
 - Определянето на точната стойност е предмет на по-математически разсъждения
- `if (fabs(x - y) <= 1e-14) ...`

Време за въпроси

