

VEJITVENI STAVKI

1

PODATKOVNI TIP `bool`
LOGIČNI IZRAZI / POGOJI
`if` STAVEK
`switch` STAVEK

PODATKOVNI TIP `bool`

2

- logični izrazi so izrazi, ki imajo natanko eno od dveh možnih vrednosti
 - `true` - resnica
 - `false` - neresnica
- podatkovni tip za logične vrednosti je **`bool`**
- logične izraze bomo v določenih stavkih imenovali tudi pogoji

Enostavni logični izrazi

3

- enostavni logični izrazi vsebujejo **eno** primerjavo
- operatorji za primerjave
 - == enakost
 - != različnost
 - < manjše
 - <= manjše ali enako
 - > večje
 - >= večje ali enako

Enostavni logični izrazi - primeri

4

- `int a = 10;`
- `int b = 20;`

- `bool p1 = (a == b);`
- `bool p2 = (a != b);`
- `bool p3 = (a < b);`
- `bool p4 = (a <= b);`
- `bool p5 = (a + b < 30);`

Enostavni logični izrazi - primeri

5

- `int a = 10;`
- `int b = 20;`

- `bool p1 = (a == b);` // false
- `bool p2 = (a != b);` // true
- `bool p3 = (a < b);` // true
- `bool p4 = (a <= b);` // true
- `bool p5 = (a + b < 30);` // false

Sestavljeni logični izrazi

6

- sestavljeni logični izrazi vsebujejo
 - vsaj en enostavni logični izraz
 - vsaj en logični operator

- logični operatorji

○ negacija	...	unarni operator
○ in	...	binarni operator
○ ali	...	binarni operator

Negacija

7

- negacija obrne vrednost pogoja
- operator za negacijo je !
 - (!pogoj)
 - npr. (! (a > b))
- resničnostna tabela za negacijo

(pogoj)	(!pogoj)
true	false
false	true

In hkrati

8

- "in hkrati" združuje dva ali več pogojev
- operator za "in hkrati" je &&
 - (pogoj1 && pogoj2)
 - npr. (x > 0 && y > 0)
- resničnostna tabela za "in hkrati"

pogoj1	pogoj2	(pogoj1 && pogoj2)
true	true	true
true	false	false
false	true	false
false	false	false

Ali

9

- "ali" združuje dva ali več pogojev
- operator za "ali" je `||`
 - `(pogoj1 || pogoj2)`
 - npr. `(x > 0 || y > 0)`
- resničnostna tabela za "ali"

pogoj1	pogoj2	(pogoj1 pogoj2)
true	true	true
true	false	true
false	true	true
false	false	false

Sestavljeni logični izrazi - primeri

10

- `(a + b > c && a != 0 || c > 10)`
- `(a + b > c && (a != 0 || c > 10))`
- **Prioriteta operatorjev:**
 - oklepaji
 - negacija
 - in hkrati
 - ali

Vejitveni stavki

11

- omogočajo nadzor nad tem, ali se skupina stavkov izvede ali ne
- dve vrsti vejitvenih stavkov
 - if stavek
 - switch stavek

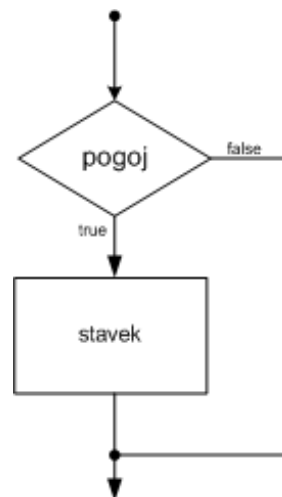
if stavek

12

- poznamo več oblik
- 1. oblika ima naslednjo sintakso


```
if (pogoj)
  stavek;
```
- beremo tako:

Če je res, kar pravi pogoj,
potem izvedi stavek.
- stavek – je en stavek v C++ ali sestavljen stavek
- sestavljen stavek – več stavkov znotraj zavitih oklepajev



if stavek - primer

13

V spremenljivki x imamo shranjeno celo število. Želimo imeti absolutno vrednost tega števila.

```
if (x < 0) {
    x = -x;
}
```

Zavite oklepaje smo pisali, kljub temu, da imamo samo en stavek. To ni narobe, je pa narobe, če oklepaje izpustimo, kadar želimo pod pogojem izvesti več stavkov. Glej naslednji primer.

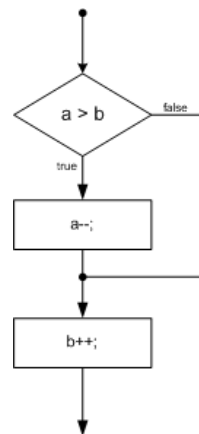
if stavek - primer

14

Imamo dve celi števili a in b . Če je a večji od b , želimo zmanjšati a in povečati b .

```
if (a < b)
    a--;
    b++;
```

Ker ni zavutih oklepajev, samo prvi stavek spada k if stavku, drugi se izvede vedno.



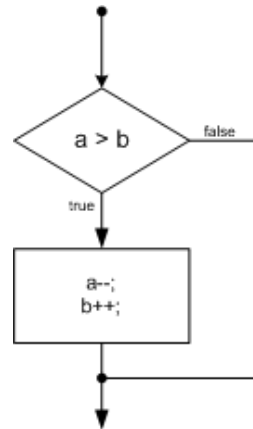
if stavek - primer

15

Imamo dve celi števili a in b . Če je a večji od b , želimo zmanjšati a in povečati b .

```
if (a < b) {
    a--;
    b++;
}
```

Z zavitimi oklepaji povemo, da oba stavka spadata k if stavku in se izvedeta, ko je določen pogoj.



if stavek - 2. oblika

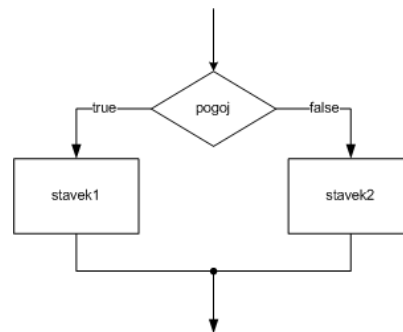
16

- 2. oblika ima naslednjo sintakso

```
if (pogoj)
    stavek1;
else
    stavek2;
```

- beremo tako:

Če je res, kar pravi pogoj, potem izvedi stavek1, v nasprotnem primeru izvedi stavek2.



if stavek - primer

17

- Preveri sodost oz. lihost nekega prebranega celega števila.

```
if (stevilo % 2 == 0) {
    cout << "Stevilo je sodo." << endl;
}
else {
    cout << "Stevilo je liho." << endl;
}
```

if stavek - primeri

18

1. Napiši program, ki prebere dve celi števili. Na ekran izpišite manjše od obeh števil.
2. Napiši program, ki prebere dve celi števili a in b . Preverite, ali je število a večkratnik števila b .
3. Napiši program, ki preko tipkovnice prebere celo število med 1 in 5. Na ekran z besedo izpišite vrednost števila.

REŠITVE: na tabli ☺

izbirni stavek

19

- včasih želimo glede na vrednost neke spremenljivke izvesti določene ukaze
- torej imamo spremenljivko in več vrednosti
- vsaka vrednost določa neko dejanje

- lotimo na dva načina
 - gnezdeni if stavki
 - switch stavek

gnezdeni if stavki

20

```
if (stevilo == 1)
    cout << "ENA";
else
    if (stevilo == 2)
        cout << "DVA";
    else
        if (stevilo == 3)
            cout << "TRI";
        else
            cout << "Ni stevilo med 1 in 3";
```

switch stavek - primer

21

```

switch (stevilo) {
  case 1:
    cout << "ENA";
    break;
  case 2:
    cout << "DVA";
    break;
  case 3:
    cout << "TRI";
    break;
  default:
    cout << "NI STEVILO MED 1 IN 3.";
}

```

switch stavek

22

V splošnem je sintaksa switch stavka naslednja:

```

switch ( test ) {
  case vrednost1 :
    // stavki za primer test == vrednost1
    ...
    break;
  case vrednost2 :
    // stavki za primer test == vrednost2
    ...
    break;
  default:
    // stavki za primer, ko test ni enak
    // nobeni od zgornjih vrednosti
    ...
}

```

switch stavek – kako deluje?

23

```

switch ( test ) {
  case vrednost1 :
    // test == vrednost1
    ...
    break;
  case vrednost2 :
    // test == vrednost2
    ...
    break;
  default:
    // vse druge vrednosti
    ...
}

```

izraz, ki se ovrednoti na števeno vrednost

nato se ena za drugo preverjajo naštete vrednosti – torej ali je test enak kateri od teh vrednosti

če je enak, izvede stavke pri tisti vrednosti

stavek break, skoči nemudoma ven iz switch stavka (ne preverja nadaljnjih vrednosti)

možnost default – če test ni enak nobeni od prej naštetih vrednosti, izvede te stavke. Po potrebi lahko možnost default izpustimo.