

Grananje u programu

programiranje_c++_2018

Logički podaci

To su podaci čija vrednost je rezultat nekog logičkog izraza i može imati dva stanja:

***true**-tačno*

***false**-netačno*

Za tip ovih podatka se koristi službena reč C++ jezika **bool**

Logički podaci i izrazi.

Logičke relacije tj. izrazi su izrazi koji su zapravo rezultati nekih poređenja, i koriste sledeće operatore:

<, >, <=, >=, !=, ==

“manje od”

“veće od”

“manje ili jednako”

“veće ili jednako”

“negacija”

“nije jednako”

“jednako”

Logički podaci i izrazi- primeri.

```
int a=10,b=8,c;  
bool b;  
c=a+b; // celobrojni izraz  
c=a>b; // netačno  
b=a>b; // ispravno i iznosi  
      // true  
b=! (a>b); // false
```

Logički podaci i složeniji izrazi

Da bi povezali više manjih logičkih izraza koristimo operatore:

"&&" – *Logičko "i"*

"||" – *Logičko "ili"*

Logički podaci i složeniji izrazi-primeri

Pretpostavimo da imamo sledeći kod:

```
int a,b,c;  
a=10,b=4;c=2;  
bool d,e;
```

Tada je:

```
d = (a==0) && (b>c) /* d=false && true;  
d=false; */
```

```
e=(a==0) ||(b>c) /* e=false || true;  
e=true; */
```

Logički podaci i složeniji izrazi

Složeniji logički izraz koji je dobijen povezivanjem dva prosta logička izraza sa **"&&"** će biti tačan samo ako su oba prosta izraza tačna tj:

true && *true* = *true*

true && *false* = *false*

false && *true* = *false*

false && *false* = *false*

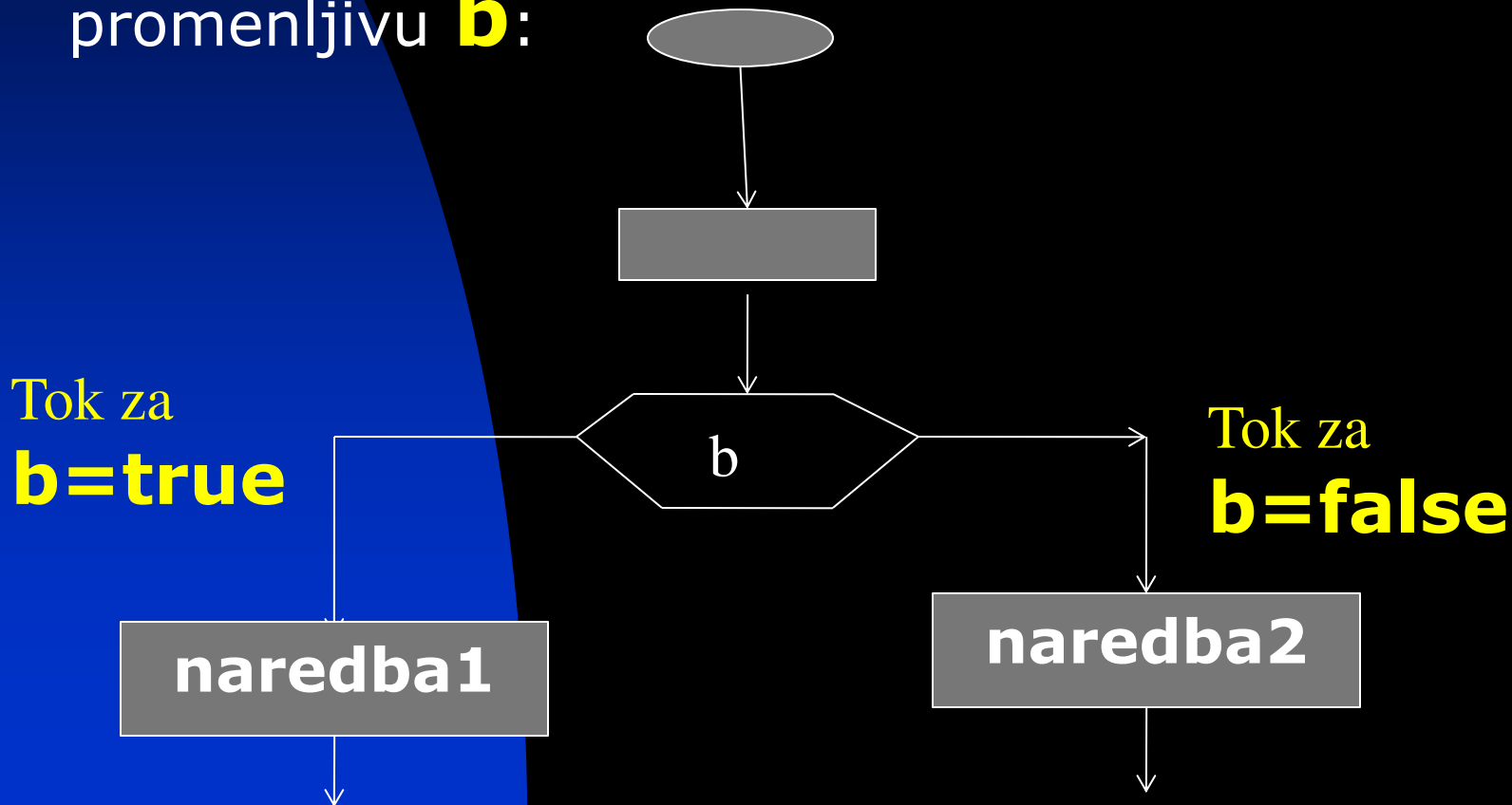
Logički podaci i složeniji izrazi

Složeniji logički izraz koji je dobijen povezivanjem dva prosta logička izraza sa "||" će biti tačan ako je bar jedan od prostih izraza tačan tj:

<i>true</i>		<i>true</i>	=	<i>true</i>
<i>true</i>		<i>false</i>	=	<i>true</i>
<i>false</i>		<i>true</i>	=	<i>true</i>
<i>false</i>		<i>false</i>	=	<i>false</i>

Grananje u programu i logički podaci.

Logički podaci se koriste kao skretnica kod grananja u programu. Ako imamo logičku promenljivu **b**:



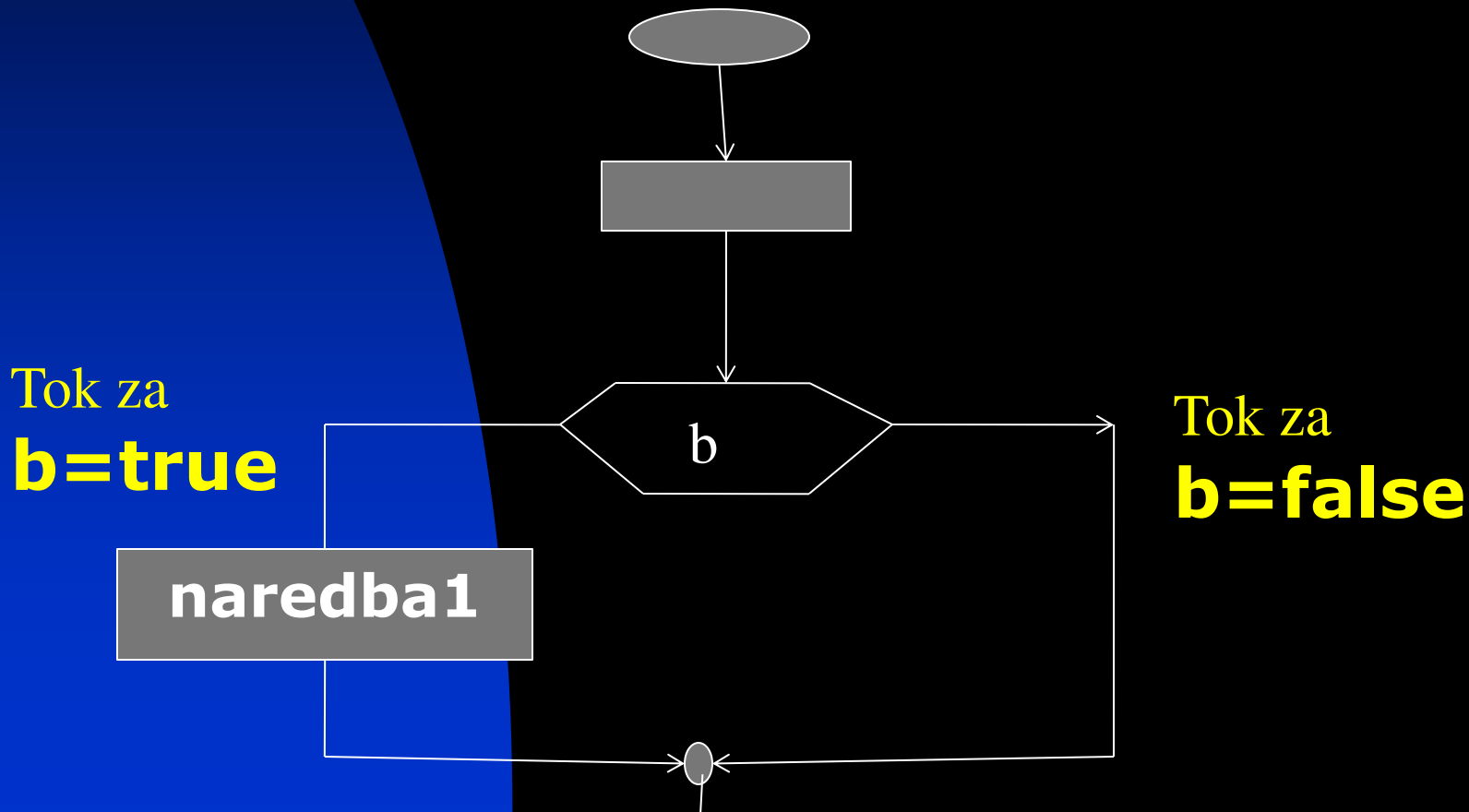
Grananje u programu i logički podaci.

Ovo grananje se vrši pomoću naredbe if-else

```
if(b)
{
    naredba1;
}
else
{
    naredba2;
}
```

Grananje u programu i logički podaci.

Ako u drugoj grani nemamo naredbu onda ova grana služi da preskoči naredbu1



Grananje u programu i logički podaci.

Ovo preskakanje se vrši pomoću naredbe **if**

```
if(b)
{
    naredba1;
}
```

-
-
-

Ovo je uslovno izvršavanje naredbe 1

Bez obzira da li će se izvršiti naredba1 ili preskočiti, program nastavlja sa izvršavanjem posle zatvorene zagrade

Grananje u programu i uslov.

Logički izraz može biti napisan direktno u maloj zagradi bez uvođenja **boolean** promenljive.

Pretpostavimo da imamo:

int a=3,x=5;

```
if(a > x)
{
    naredba1;
}
.
```

a > x je logički izraz i u ovom primeru daje **false**, tako da se naredba 1 neće izvršiti

Grananje u programu sa 3 grane

```
if(b1)
{
    naredba1;
}
else if(b2)
{
    naredba2;
}
else{
    naredba3;
}
```

Ovo grananje se vrši pomoću naredbe **if-else if-else** i dva uslova(logičke promenljive)

b1 i b2 su logičke promenljive

Grananje u programu sa 3 grane-može i ovako

```
if(a>0)
{
    naredba1;
}
else if(a==0)
{
    naredba2;
}
else{
    naredba3;
}
```

Ovo grananje se vrši pomoću naredbe if-else if-else i dva uslova(logički izrazi)

a>0 i a==0
su logički izrazi

Primer T3.1: Razlomak

Uneti dva realna broja x i y , i
izračunaj vrednost razlomka.

Primer T3.1: Razlomak

Uneti dva realna broja x i y , i
izračunaj vrednost razlomka.

```
double x,y,z;  
x=scanf("%lf",&x);  
y=scanf("%lf",&y);
```

```
z=x/y;
```

Ovo rešenje nije
potpuno tačno jer
razlomak nije
definisano za $y=0$

Primer T3.1 : Razlomek

Ispravno rešenje

```
double x,y,z;  
scanf("%lf", &x);  
scanf("%lf", &y);  
if(y!=0)  
{  
    z=x/y;  
}  
else{  
    printf("deljenje sa 0");  
}
```

Primer T3.2

- Tri automobila krecu sa startne pozicije sa kašnjenjem od $T_1 > T_2 > T_3$ sekundi, brzinama v_1, v_2, v_3 odrediti koji automobil je na prvom mestu posle vremena $t > T_1$



Hvala na pažnji

