



SVEUČILIŠTE U ZAGREBU  
Fakultet prometnih znanosti  
Zavod za inteligentne transportne sustave  
Vukelićeva 4, Zagreb, HRVATSKA



# Računalstvo

## Naredbe grananja

**Doc. dr. sc. Edouard Ivanjko, dipl.ing.**

# Sadržaj

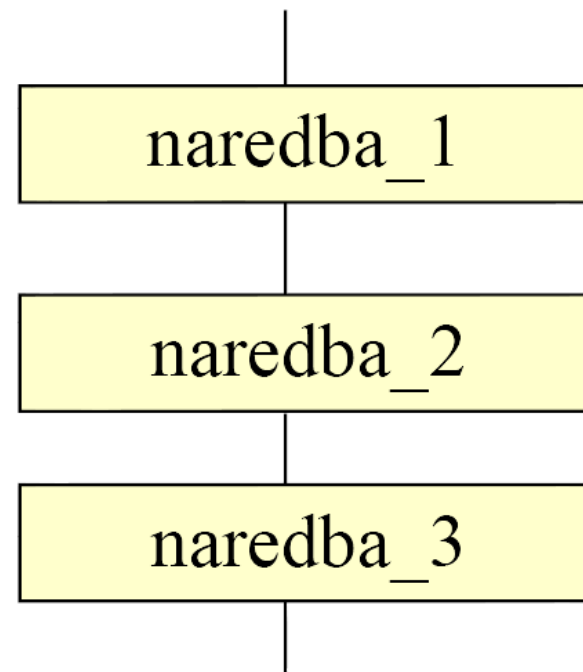
---

- Uvod
- Kreiranje uvjeta
- Grananje if
- Grananje if - else
- Grananje if - else if - else
- Grananje switch
- Primjeri



# Uvod

- Dosadašnji programi se izvode slijedno
  - Naredba po naredba, linija kôda po linija kôda
- Potreba za izvođenjem dijela naredbi programa zavisno o ispunjenju određenog uvjeta
  - Provjera unesenog podatka
  - Provjera ispravnosti rezultata
  - Konačni rezultat se računa različito za različite međurezultate
    - Npr. korijen kvadratne jednadžbe



- Kod grananja ili strukture odlučivanja postoji uvjet
  - Logički izraz
    - Mogućnost usporedbe numeričkih vrijednosti, niza znakova i logičkih vrijednosti
  - Ako je uvjet ispunjen izvodi se pripadni programski blok
  - Ako uvjet **nije** ispunjen **ne** izvodi se pripadni programski blok
    - Mogućnost uvođenja programskog bloka za ovaj slučaj
      - Moguće dodati dodatni uvjet za izvođenje programskog bloka za slučaj neispunjavanja osnovnog uvjeta

- Osnovne strukture grananja
  - Naredba **if**
    - Jednostavna selekcija -> programski blok se ili izvodi ili preskače
  - Kombinacija naredbi **if - else**
    - Selekcija s mogućnošću izvođenja dva različita programska bloka
  - Kombinacija naredbi **if – else if - else**
    - Selekcija s mogućnošću izvođenja tri ili više različita programska bloka
  - Naredba **switch**
    - Selekcija zasnovana na vrijednosti varijable ili izraza



# Kreiranje uvjeta

- Uvjet se općenito sastoji od složenog logičkog izraza
  - Kombinacija aritmetičkog (numeričkog) dijela i logičkog dijela
    - Aritmetički dio ima veći prioritet -> računa se prvi
    - Numerička vrijednost 0 je laž ("false")
    - Numerička vrijednost **različita** od 0 je istina ("true")
    - Sljedeći logički izrazi su ekvivalentni
$$c \ \&\& \ b + 3 * a$$
$$c \ \&\& \ ( b + (3 * a) )$$
- Logički uvjeti se računaju slijedno
  - Slijeva na desno po dva para prema prioritetu



# Kreiranje uvjeta – Prioritet operatora

	OPERATORI	PRIDRUŽIVANJE
↑ Viši prioritet	( )	L → D
	! ~ ++ -- sizeof & * unarni + -	D → L
	(cast)	D → L
	* / %	L → D
	+ -	L → D
	<< >>	L → D
	< <= > >=	L → D
Niži prioritet →	== !=	L → D
	&	L → D
	^	L → D
		L → D
	&&	L → D
		L → D
	? :	D → L
	= *= /= %= += -= &= ^=  = <<= >>=	D → L
	,	L → D



# Kreiranje uvjeta – Primjeri

- Rezultat ispisa je “true” ili “false”

```
// deklaracija varijabli  
int a = 1, b = 2, c = 3, e = 0, f = -1;  
bool d;
```

## LogickiUvjeti

```
// primjer 1  
d = a < 10 && 2 * b < c;  
Console.WriteLine("Rezultat prvog logickog primjera je " + d);  
  
// primjer 2  
d = a > b && e > c || f < 0;  
Console.WriteLine("Rezultat drugog logickog primjera je " + d);  
  
// primjer 3  
d = a < b || e > c && f < 0;  
Console.WriteLine("Rezultat treceg logickog primjera je " + d);  
  
// primjer 4  
d = !(a < b) || e > c && f != 0;  
Console.WriteLine("Rezultat cetvrtog logickog primjera je " + d);
```





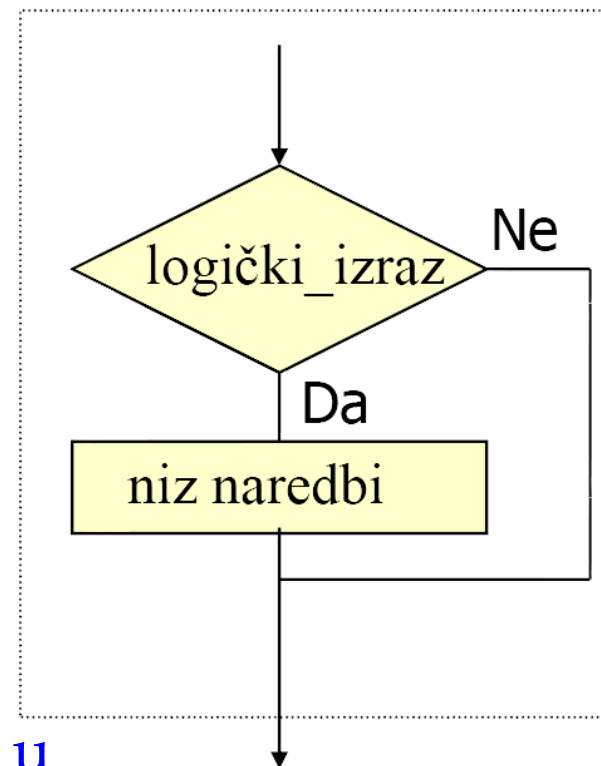
# Grananje - if

- Pseudokôd

ako je logički\_izraz onda  
niz\_naredbi

- C#

```
if ( x > 0 ) // provjera uvjeta
{
    Console.WriteLine("x je pozitivan broj");
}
```



- Ako je uvjet ispunjen izvodi se naredba u { zagradama }
- Ako uvjet nije ispunjen ne izvodi se naredba u { zagradama }
- U slučaju jedne naredbe nisu potrebne vitičaste zagrade

# Grananje – if – Primjer

- Računanje apsolutne vrijednosti učitano cijelog broja

```
// deklaracija varijable
```

```
int unos;
```

 **ApsolutnaVrijednost**

```
// unos podataka
```

```
Console.Write("Unesite broj > ");
```

```
unos = Convert.ToInt32(Console.ReadLine());
```

```
// izracun apsolutne vrijednosti
```

```
if (unos < 0)
```

```
    unos = -unos;
```

```
// ispis rezultata
```

```
Console.WriteLine("\nApsolutna vrijednost iznosi " + unos);
```

```
// da se vidi ispis konzole
```

```
Console.ReadLine();
```



# Grananje – if - else

- Pseudokôd

ako je logički\_izraz onda

niz\_naredbi\_1

inače

niz\_naredbi\_2

- C#

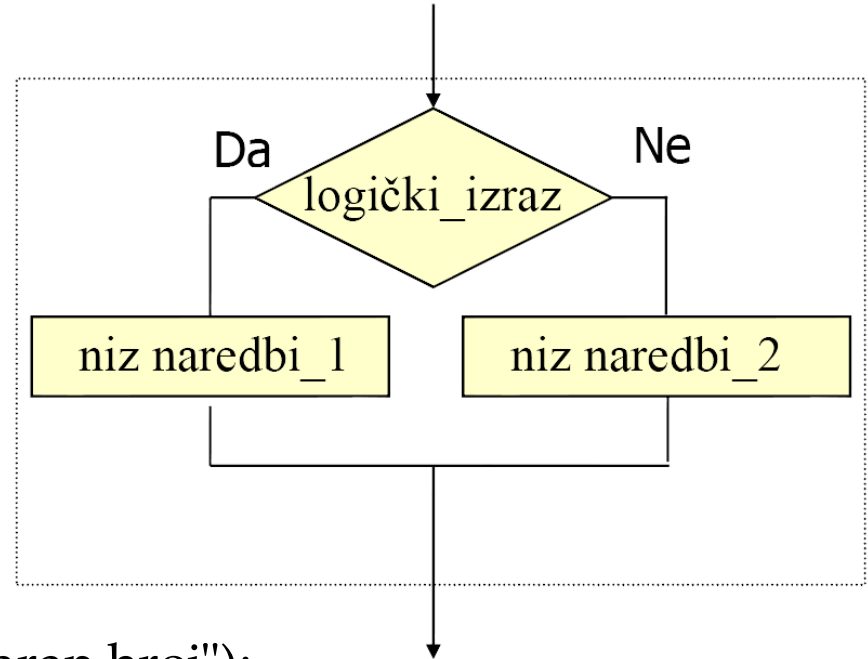
if (x % 2 == 0)

Console.WriteLine("x je paran broj");

else

Console.WriteLine("x je neparan broj");

- Izvođenje s dvije mogućnosti nastavka izvođenja naredbi zavisno o ispunjenju uvjeta



# Grananje – if - else - Primjer

- Primjer provjere mase vozila
  - Ovisno o vrsti ceste i masi vozila operateru se ispiše odgovarajuća poruka

```
// deklaracija varijabli
bool vrstaCeste;
double masaVozila;

// unos podataka
Console.WriteLine("Jeli cesta za osobna (0) ili teretna vozila (1) > ");
vrstaCeste = Convert.ToBoolean( Convert.ToInt16(Console.ReadLine()));
Console.WriteLine("Unesite masu vozila [kg] > ");
masaVozila = Convert.ToDouble(Console.ReadLine());

// provjera mase
if (!vrstaCeste && (masaVozila >= 5000.0))
    Console.WriteLine("UPOZORENJE! Masa vozila je prevelika!");
else
    Console.WriteLine("Masa vozila je unutar dopustenog podrucja!");

// da ispis ostane vidljiv
System.Console.ReadKey();
```

 **ProvjeraMase**



# Grananje – if - else - Gniježđenje

- Pojedina **if** struktura grananja unutar sebe može imati više **if** struktura grananja
  - Vrijedi i za **if – else** strukturu grananja
  - Pravilo programiranja
    - “**else**” dio pripada najbližem “**if**”-u koji nema svoj “**else**”
    - Radi lakšeg snalaženja koristi se uvlačenje

## Pogrešno "uvučeno"

```
if( uvjet1 )
    if ( uvjet2 )
        naredba2;
else
    naredba3;
```

## Ispravno "uvučeno"

```
if( uvjet1 )
    if ( uvjet2 )
        naredba2;
else
    naredba3;
```



# Grananje – if - else - Gniježđenje

- Osim uvlačenja koristi se i razdvajanje pomoću vitičastih zagrada

```
if( uvjet1 )
{
    if ( uvjet2 )
        naredba2;
}
else
    naredba3;
```

- “else” dio pripada gornjem “if”-u kako je drugi “if” odvojen vitičastim zagradama



# Grananje – if - else - Gniježđenje

- Primjer računanja najmanjeg broja

```
static void Main(string[] args)
{
```

 **Najmanji**

```
    int min;
```

```
    int br1 = Convert.ToInt32(Console.ReadLine());
```

```
    int br2 = Convert.ToInt32(Console.ReadLine());
```

```
    int br3 = Convert.ToInt32(Console.ReadLine());
```

```
    if (br1 < br2)
```

```
        if (br1 < br3) min = br1;
```

```
        else min = br3;
```

```
    else
```

```
        if (br2 < br3) min = br2;
```

```
        else min = br3;
```

```
    Console.WriteLine("Najmanji broj je broj: " + min);
```

```
    Console.ReadKey();
```

```
}
```

Ugniježdene **if** naredbe

Osnovno uvjetno grananje

Ugniježdjeno uvjetno grananje



# Grananje – if - else if - else

- Pseudokôd

ako je logički\_izraz\_1 onda

niz\_naredbi\_1

inače ako je logički\_izraz\_2 onda

niz\_naredbi\_2

inače

niz\_naredbi\_3

- C#

if (x > 0)

Console.WriteLine("x je pozitivan broj");

else if (x < 0)

Console.WriteLine("x je negativan broj");

else

Console.WriteLine("x je nula");

Omogućena višestruka  
selekcija programskog  
bloka uz korištenje više  
uvjeta provjeravanja!





# Grananje – if - else if - else - Primjer

```
// deklaracija varijabli
int kategorijaVozila;
double cijena;

// unos podataka
Console.WriteLine("Program za racunanje cestarine\n\n" +
    "Kategorija vozila IA (1)\n" +
    "Kategorija vozila I (2)\n" +
    "Kategorija vozila II (3)\n" +
    "Kategorija vozila III (4)\n" +
    "Kategorija vozila IV (5)\n\n" +
    "Odaberite kategoriju vozila > ");
kategorijaVozila = Convert.ToInt32(Console.ReadLine());

// izracun cijene cestarine
if(kategorijaVozila == 1)
    cijena = 10.0;
else if(kategorijaVozila == 2)
    cijena = 20.0;
else if(kategorijaVozila == 3)
    cijena = 30.0;
else if(kategorijaVozila == 4)
    cijena = 40.0;
else if(kategorijaVozila == 5)
    cijena = 50.0;
else
    cijena = -1.0;

// ispis cijene
if(cijena > 0.0)
    Console.WriteLine("Cijena cestarine je " + cijena + " HRK.");
else
    Console.WriteLine("GRESKA! Cijena cestarine nije mogla biti izracunata!");
```

## NaplataCestarine

Računanje cijene cestarine  
pomoću if - else if - else provjere



# Grananje switch

- Naredba switch naziva se još i skretnica
  - Koristi se umjesto višestranne “if” selekcije
  - Sintaksa 

```
switch( cjelobrojni_izraz )  
{  
    case const_izraz1: naredbe1;  
    [case const_izraz2: naredbe2;]  
    ...  
    [default : naredbeN;]  
}
```
  - Izraz uz naredbu “switch” mora biti cjelobrojan
  - Izraz uz “case” mora biti cjelobrojan i konstanta
    - Vrijednost za usporedbu mora biti unaprijed poznata
  - Zadnja naredba u pojedinom “case” programskom bloku mora biti “break”
    - Uzrokuje preskakanje ostalih slučajeva



# Grananje switch

- Mogućnost korištenja propadanja
  - Više case dijelova ima isti programski kôd

```
switch (znak) {  
    case 'a': brsamoglasnika++; break;  
    case 'e': brsamoglasnika++; break;  
    case 'i': brsamoglasnika++; break;  
    case 'o': brsamoglasnika++; break;  
    case 'u': brsamoglasnika++; break;  
    default: brostalih++; break;  
}
```



```
switch (znak) {  
    case 'a':  
    case 'e':  
    case 'i':  
    case 'o':  
    case 'u': brsamoglasnika++; break;  
    default: brostalih++; break;  
}
```

# Grananje switch - Primjer

- Klasifikacija ocjena

```
static void Main(string[] args)
{
    string prezime, ocjena;
    Console.WriteLine("Upisi prezime: ");
    prezime = Console.ReadLine();
    Console.WriteLine("Upisi ocjenu: ");
    int x = Convert.ToInt32(Console.ReadLine());
    switch (x)
    {
        case 1:
            ocjena = "nedovoljan"; break;
        case 2:
            ocjena = "dovoljan"; break;
        case 3:
            ocjena = "dobar"; break;
        case 4:
            ocjena = "vrlo dobar"; break;
        case 5:
            ocjena = "izvrstan"; break;
        default:
            ocjena = "?"; break;
    }
    Console.WriteLine("Student " + prezime + " dobio je ocjenu " + ocjena);
    Console.ReadKey();
}
```

 Odlucivanje03

Struktura odluke  
zasnovana na  
vrijednosti varijable

X



# Grananje switch - Primjer

```
// izracun cijene cestarine
```

```
switch (kategorijaVozila)
```

```
{
```

```
    case 1:
```

```
        cijena = 10.0;
```

```
        break;
```

```
    case 2:
```

```
        cijena = 20.0;
```

```
        break;
```

```
    case 3:
```

```
        cijena = 30.0;
```

```
        break;
```

```
    case 4:
```

```
        cijena = 40.0;
```

```
        break;
```

```
    case 5:
```

```
        cijena = 50.0;
```

```
        break;
```

```
    default:
```

```
        cijena = -1.0;
```

```
        break;
```

```
}
```

```
// ispis cijene
```

```
if (cijena > 0.0)
```

```
    Console.WriteLine("Cijena cestarine je " + cijena + " HRK.");
```

```
else
```

```
    Console.WriteLine("GRESKA! Cijena cestarine nije mogla biti izracunata!");
```

Računanje cijene cestarine  
pomoću switch grananja



**NaplataCestarineSwitch**

