

Programiranje – III razred

Ponavljjanje gradiva

Ispis podataka na konzoli



Ispis podataka na konzoli se realizuje preko klase **Console**, pozivajući neku od njenih metoda:

- **Console.Write**(„OVDE JE TEKST KOJI ĆI BITI ISPISAN“);
- **Console.WriteLine**(„TEKST ZA ISPIS...“);

Razlika u načinu ispisa ovih metoda je u položaju kursora nakon ispisa.

- Kod metode **Write()**, kursor ostaje odmah iza ispisane poruke očekujući unos teksta, dok kod metode
- **WriteLine()**, kursor prelazi u novi red.

Dodatne mogućnosti:

U okviru teksta koji se ispisuje, moguće je umetnuti vrednosti pojedinih promenjivih tako što se u vitičaste zagrade unosi redni broj promenjive, pa se na kraju, iza navodnika, stavlja zarez i redom se pobroje, zarezima odvojene promenjive čiji će se sadržaj ispisati.

Ispis podataka na konzoli



Na primer:

```
int visina = 195;  
int masa = 73;
```

```
Console.WriteLine("Visina korisnika je {0}, a njegova masa je {1} kg.", visina, masa);
```

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

Klasičan način ispisa bi bio:

```
Console.WriteLine("Visina je " + visina + " a težina je "+ masa );
```

Komentar:

Ono što se nalazi pod navodnicima će biti ispisano kao tekst. Tom tekstu se često dodaju vrednosti promenjivih iz programa.

Ispis teksta sa prelaskom u novi red uključuje dodavanje: `\n` (`\t` - ubacuje razmak tabulatora,..)

Ispis podataka na konzoli



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace IspisPodataka
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Ovde je tekst iza koga cursor ostaje u istom redu..");
            Console.Write("...nastavak teksta.");
            Console.Write("\n\n\nPress any key to continue . . . ");
            Console.ReadKey(true);
            //Brisanje sadržaja na konzoli
            Console.Clear();
            // ISPIS PREKO NAREDBE Console.WriteLine();
            Console.WriteLine("Ovde je tekst iza koga cursor prelazi u novi red..");
            Console.WriteLine("...nastavak teksta.");
            Console.Write("\n\n\tPress any key to continue . . . ");
            Console.ReadKey(true);
            // Deo \n - pri ispisu prelazi u novi red.
        }
    }
}
```

Unos podataka preko konzole



Unos podataka u program preko konzole uglavnom se svodi na sledeće komande:

- **Console.Read();**

Čita sledeći znak sa unosa. Učitani podatak je tipa **int**.

- **Console.ReadKey();**

Čita sledeći znak sa unosa. Učitani podatak je tip **ConsoleKeyInfo**.

- **Console.ReadLine();**

Čita sledeći niz znaka sa unosa. Učitani podatak je tip **string**.

Console.Read();



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace UnosJednogZnaka
{
    class Program
    {
        static void Main(string[] args)
        {
            // Demonstracija očitavanja jednog znaka unetog preko tastature.
            char unetiTaster;

            Console.WriteLine("PRITISNITE NEKI TASTER...\n");

            //Očitavanje tastera:
            // Naredba koristi metodu Read() koja vraća vrednost tipa int i zato se
            //koristi operator za konverziju koji vrednost tipa int konvertuje u tip char

            unetiTaster = (char)Console.Read();

            // Ispis rezultata unosa
            Console.WriteLine("\nPritisnuli ste taster: " + unetiTaster + "\n\n\n\n\n");

            //Pauza na konzoli
            Console.WriteLine("Pritisnite bilo koji taster za završetak...");
            Console.ReadKey();
        }
    }
}
```

Console.ReadKey();



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace UnosJednogZnaka
{
    class Program
    {
        static void Main(string[] args)
        {
            ConsoleKeyInfo unetiTaster;

            Console.WriteLine("Unesite neki tekst...\n");

            unetiTaster = Console.ReadKey();

            // Ispis rezultata unosa
            Console.WriteLine("\nUneli ste sledeci taster: " +
                unetiTaster.Key.ToString() + "\n\n\n\n");

            //Pauza na konzoli
            Console.WriteLine("Pritisnite bilo koji taster za završetak...");
            Console.ReadKey();
        }
    }
}
```

Console.ReadLine();



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace UnosJednogZnaka
{
    class Program
    {
        static void Main(string[] args)
        {
            // Demonstracija očitavanja jednog znaka unetog preko tastature.

            string unetiTaster;

            Console.WriteLine("Unesite neki tekst...\n");

            unetiTaster = Console.ReadLine();

            // Ispis rezultata unosa
            Console.WriteLine("\nUneli ste slede tekst: " + unetiTaster + "\n\n\n\n\n");

            //Pauza na konzoli
            Console.WriteLine("Pritisnite bilo koji taster za završetak...");
            Console.ReadKey();
        }
    }
}
```


Zadaci za vežbu



1. Realizovati konzolnu aplikaciju namenjenu unosu podataka za osobu. Potrebno je uneti sledeće podatke:

- Prezime,
- Ime,
- Mesto stanovanja, i
- Ulicu i broj.

Na kraju, obezbediti konzolni prikaz unetih podataka.

2. Realizovati konzolnu aplikaciju koja za unete tri stranice trougla računa njegovu površinu.

Napomena: Potrebno je koristiti Heronov obrazac za računanje površine trougla. Nije potrebna provera unetih brojeva (da li mogu činiti trougao).

Napuštanje Main metode



2. Obezbediti mehanizam za napuštanje Main metode u okviru klase. Nakon napuštanja Main metode, na konzoli ispisati tekst: „**Ovo je tekst ispisano van Main metode..**“

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IzlazIzMainMetode
{
    class Program
    {
        static void Main(string[] args)
        {
            Program p = new Program(); // Program je naziv tekuće klase
            // Ovim se program pomera u prazan konstruktor klase Program koji se kreira
            }
            Program()
            {
                // Konstruktor klase Program bez argumenata
                Console.WriteLine("Ovo je tekst ispisano van Main metode..");
                Console.ReadLine();
            }
        }
    }
}
```

Napuštanje Main metode



2. Obezbediti mehanizam za napuštanje Main metode u okviru klase. Nakon napuštanja Main metode, na konzoli ispisati tekst: „**Ovo je tekst ispisano van Main metode..**“

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IzlazIzMainMetode
{
    class Program
    {
        static void Main(string[] args)
        {
            Program p = new Program(); // Program je naziv tekuće klase
            // Ovim se program pomera u prazan konstruktor klase Program koji se kreira
        }
        Program() ←
        {
            // Konstruktor klase Program bez argumenata
            Console.WriteLine("Ovo je tekst ispisano van Main metode..");
            Console.ReadLine();
        }
    }
}
```

Kreiranje metoda



3. Napustiti main metodu, uneti tri broja i na konzoli prikazati zbir tih brojeva. Unos brojeva, izračunavanje zbira i ispis rezultata realizovati metodama.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace KreiranjeMetoda1
```

```
{
```

```
    class Program
```

```
    {
```

```
        //Nakon izlaska iz Main, program nastavlja ovde...
```

```
        Program()
```

```
        {
```

```
            // Deklarisanje promenljivih
```

```
            int a, b, c;
```

```
            int suma = 0;
```

```
            //Pozivanje metode za unos promenljivih:
```

```
            a = unesiPromenljivu(1);
```

```
            b = unesiPromenljivu(2);
```

```
            c = unesiPromenljivu(3);
```

```
            //Pozivanje metode za izracunavanje zbira
```

```
            suma = saberiBrojeve(a, b, c);
```

```
            // Pozivanje metode za ispis rezultata
```

```
            ispisiRezultat(suma);
```

```
        }
```



Pozivanje metode **unesiPromenljivu** – tri puta

Pozivanje metode **saberiBrojeve**

Pozivanje metode **ispisiRezultat**

Kreiranje metoda



3. Napustiti main metodu, uneti tri broja i na konzoli prikazati zbir tih brojeva. Unos brojeva, izračunavanje zbira i ispis rezultata realizovati metodama.

```
private void ispisiRezultat(int suma)
{
    Console.Clear();
    Console.WriteLine("Zbir unetih brojeva je: " + suma);
    Console.ReadLine();
}
```

Metoda: **ispisiRezultat**

```
private int saberiBrojeve(int a, int b, int c)
{
    return a + b + c;
}
```

Metoda: **saberiBrojeve**

```
private int unesiPromenljivu(int x)
{
    Console.Write("Unesite {0}. broj: ",x);
    return Convert.ToInt32(Console.ReadLine());
}
```

Metoda: **unesiPromenljivu**

```
static void Main(string[] args)
{
    // Izlazak iz Main metode
    Program p = new Program();
}
```

```
}
}
```

Zadaci za vežbu



3. Realizovati konzolnu aplikaciju koja za unete tri stranice trougla računa njegovu površinu. Računanje površine realizovati pozivom metode **izracunajPovrsinu**.

Napomena: Potrebno je koristiti Heronov obrazac za računanje površine trougla. Nije potrebna provera unetih brojeva (da li mogu činiti trougao).

4. Realizovati konzolnu aplikaciju koja za unetu vrednost u dinarima računa vrednost prevedenu u € preko metode **pretvoriDinare**.

Napomena: Skinuti trenutnu vrednost € po srednjem kursu sa net – a.

Konverzija tipova – implicitna konverzija



Osnovna ideja implicitne konverzije je da se ne gube informacije i da se sačuva preciznost podataka.

Postoje pravila konverzije po kojima kompajler automatski izvršava implicitnu konverziju:

PROMENLJIVA TIPA A ČIJI SKUP MOGUĆIH VREDNOSTI JE PODSKUP SKUPA MOGUĆIH VREDNOSTI TIPA B, MOŽE SE IMPLICITNO KONVERTOVATI U TIP B.

Implicitne konverzije se najčešće primenjuju kod numeričkih tipova podataka.

Konverzija tipova – eksplicitna konverzija



Eksplicitna konverzija

Eksplicitnom konverzijom programer, dodatnim kodom, od kompajlera zahteva traženu konverziju.

Eksplicitna konverzija se ostvaruje na sledeće načine:

- Primenom operatora **cast**, ili
- Upotrebom klase **Convert**.

Operator **cast** se upotrebljava tako što se u zagradama navodi tip u koji se želi izvršiti konverzija vrednosti izraza koji sledi za operatorom **cast**.

Konverzija tipova – eksplicitna konverzija



Primer:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ImplicitnaKonverzija
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = 11;
            double b = 4.0;
            int x;
            //x = (a / b);
            //Cannot implicitly convert type 'double' to 'int'. An explicit conversion exists (are you missing a cast?)

            // Eksplicitna konverzija
            x = ((int) (a / b));
        }
    }
}
```

Konverzija tipova – eksplicitna konverzija



Primer:

Izvršiti eksplicitnu konverziju sledećeg tipa:

```
int x = 201;  
string s = (string)x;  
string p = "123";  
int y = (int)p;
```

} NEUSPELA KONVERZIJA

Prilikom pokretanja gornjeg koda dobija se izveštaj o grešci jer navedene konverzije NIJE MOGUĆE IZVESTI OPERATOROM **cast**.

Konverzija tipova – klasa Convert



Navedene konverzije, i mnoge druge, moguće je izvršiti upotrebom **statičkih metoda** klase **Convert** kojima se podaci osnovnog tipa konvertuju u drugi osnovni tip.

Svaki od metoda konverzije poziva se tako što se navodi ime metoda, a zatim u zagradama izraz čija se vrednost konvertuje.

Prethodni primer primenom klase Convert bi bio:

Prethodni pokušaj:

```
int x = 201;
string s = (string)x;
string p = "123";
int y = (int)p;
```

Konverzija sa Convert metodama:

```
int x = 201;
string s = Convert.ToString(x);
string p = "123";
int y = Convert.ToInt32(p);
```

Konverzija tipova – klasa Convert



Konverzija pomoću metoda klase **Convert** uspešno se izvršava samo kada je moguće podatak jednog tipa konvertovati u drugi.

Metode klase **Convert** mogu se primenjivati na vrednosti različitih tipova podataka. Na primer, u pozivu:

Convert.ToString(vrednost);

vrednost može biti tipa int, double , float, char,..

S obzirom na to da su sve poruke kojima Windows aplikacija komunicira sa korisnikom tekst predstavljen tipom string, neophodno je na što lakši način sve tipove podataka, ako je to moguće, prevesti u string.

Zato sve strukture, kojima su predstavljeni osnovni tipovi podataka, sadrže metod: **ToString()** kojim je omogućeno prevođenje vrednosti osnovnog tipa u string.

Konverzija tipova – klasa Convert



Primer:

```
int x = 201;  
string s = x.ToString();  
  
double a = -237.15;  
string temperatura = a.ToString();
```

Konverzija tipova – vežbanje



Zadatak 2

Napisati program koji za unetu vrednost temperature u celzijusima na izlazu ispisuje vrednost te temperature u Kelvinima i Farenhajtima.

Napomena: Izračunavanja (pretvaranja) realizovati preko posebnih metoda. Na izlazu, prilikom ispisa, izvršiti konverziju vrednosti u string.

Naredba if



Naredba **if** omogućava uslovno grananje (uslovno izvršavanje) programa. Naredba **if** može biti realizovana na više različitih načina, ali je generalna forma naredbe:

```
if (logički uslov)
{
// Kod koji se izvršava ako je ispunjen uslov u maloj zagradi
}
else
{
// Kod koji se izvršava ako nije ispunjen uslov u maloj zagradi
}
```

Naredba **else** nije obavezna!

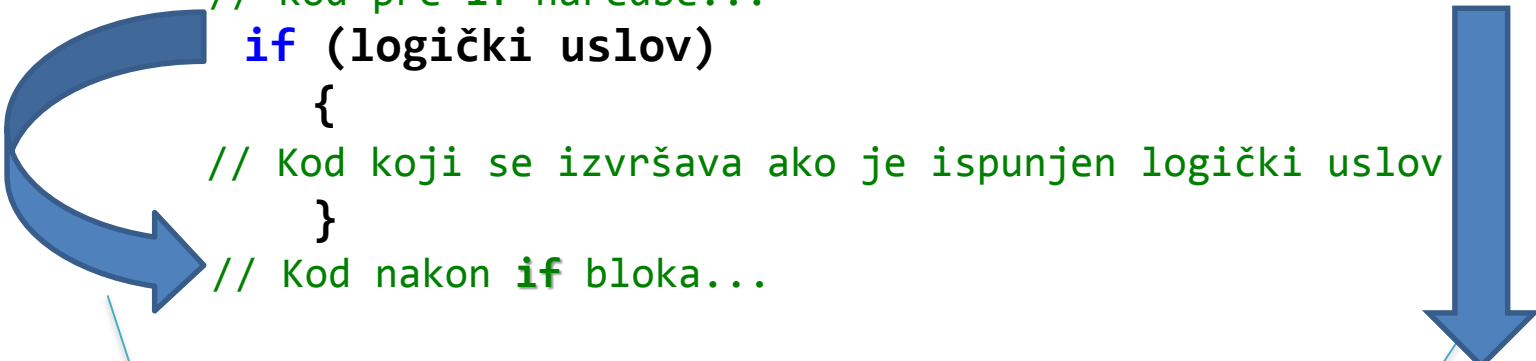
Ukoliko se naredba **else** izostavi, deo koda u okviru prvih vitičastih zagrada će se izvršiti samo ako je logički uslov=true. U protivnom, taj deo koda se preskače.

Naredba if



Naredba **if** bez upotrebe **else** dela:

```
// Kod pre if naredbe...  
if (logički uslov)  
  {  
    // Kod koji se izvršava ako je ispunjen logički uslov  
  }  
// Kod nakon if bloka...
```



Tok koda ako logički uslov nije ispunjen

Tok koda ako je logički uslov ispunjen

Ukoliko je u okviru **if** naredbe samo jedna linija koda, vitičaste zagrade se mogu izostaviti:

```
if (logički uslov)  
  // Jedna linija koda...;
```


Naredba if



Naredba **if** omogućava i višestruko uslovno grananje programa u formi sa **else If ()**:

```
if (logički uslov 1)
{
// Kod koji se izvršava ako je ispunjen uslov 1 u maloj zagradi
}
else If(logički uslov 2)
{
// Kod koji se izvršava ako je ispunjen uslov 2 u maloj zagradi
}
else
{
// Kod koji se izvršava ako nije ispunjen ni jedan gornji uslov
}
```

Else If delova programer može uvrstiti i više ukoliko je to potrebno.

Naredba if - primeri



1. Uneti ceo broj preko konzole. Ukoliko je broj pozitivan, izračunati koren tog broja i proslediti na konzolni izlaz, a ukoliko je broj negativan, na izlaz proslediti koren apsolutne vrednosti tog broja.
2. Uneti ceo broj preko konzole. Ukoliko je broj negativan, na izlaz proslediti poruku „Uneti broj je negativan“. Ukoliko broj nije negativan, na izlaz proslediti poruku „Uneti broj nije negativan“.
3. Uneti ceo broj preko konzole. Ukoliko je broj pozitivan, na izlaz proslediti poruku „Broj je pozitivan“, ukoliko je broj nula na izlaz proslediti poruku „Broj je nula“. Na kraju, ukoliko je broj negativan, na izlaz proslediti poruku „Broj je negativan.“

Višestruka if naredba

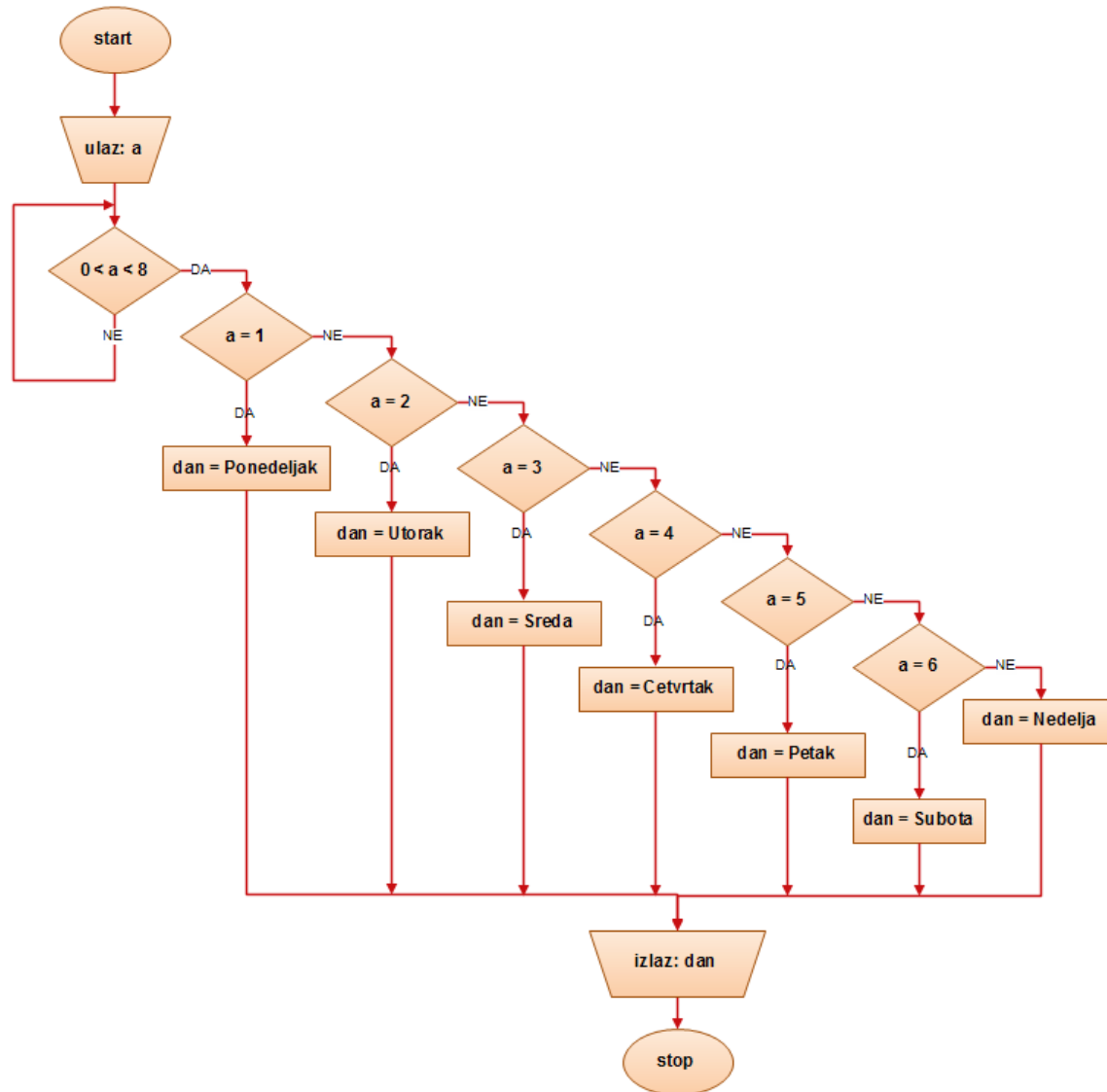


1. Ulaz konzolne aplikacije je redni broj dana u nedelji (1 – 7). Preko **if – else if – else** strukture obezbediti prikaz dana koji odgovara broju (1 = Ponedeljak).
2. Ulaz je proizvoljan prirodan broj. Proveriti da li je uneti broj deljiv sa 6 (broj je deljiv sa 6 ako je istovremeno deljiv sa 2 i sa 3).
3. Realizovati konzolnu aplikaciju za izračunavanje obima i površine trougla. Ulazni parametri su stranice trougla a , b i c , a izlazni parametri su površina P i obim O .

Uputstvo:

Prvo proveriti da li vrednosti a , b i c mogu biti stranice, a tek nakon toga primeniti Heronov obrazac za izračunavanje površine trougla.

Višestruka if naredba



Višestruka if naredba

1. Ulaz konzolne aplikacije je redni broj dana u nedelji (1 – 7). Preko **if – else if – else** strukture obezbediti prikaz dana koji odgovara broju (1 = Ponedeljak).

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace DanUNedeljiSaIf
{
    class Program
    {
        static void Main(string[] args)
        {
            string unos;
            string dan;
            int brojDana;

            Console.WriteLine("Unesite redni broj dana u nedelji (1-7):");
            unos = Console.ReadLine();
            brojDana = Convert.ToInt32(unos);

            // IF STRUKTURA
            if( brojDana == 1 )
            {
                dan = "Ponedeljak";
            }
            else if( brojDana == 2 )
            {
                dan = "Utorak";
            }

```

```
            else if ( brojDana == 3 )
            {
                dan = "Sreda";
            }
            else if ( brojDana == 4 )
            {
                dan = "Četvrtak";
            }
            else if ( brojDana == 5 )
            {
                dan = "Petak";
            }
            else if ( brojDana == 6 )
            {
                dan = "Subota";
            }
            else
            {
                dan = "Nedelja";
            }
            // Štampanje rezultata:
            Console.Clear();
            Console.WriteLine("Dan pod renim brojem {0} je: {1}", brojDana, dan);
            Console.ReadKey();
        }
    }
}
```

Višestruka if naredba



2. Ulaz je proizvoljan prirodan broj. Proveriti da li je uneti broj deljiv sa 6 (broj je deljiv sa 6 ako je istovremeno deljiv sa 2 i sa 3).

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DeljivostSaSest
{
    class Program
    {
        static void Main(string[] args)
        {
            string unos;
            string rezultat="";
            int broj;

            Console.WriteLine("Unesite
prirodan broj...\n\n");
            unos = Console.ReadLine();
            broj = Convert.ToInt32(unos);
```

```
        //Provera deljivosti
        if (broj % 2 == 0)
        {
            if (broj % 3 == 0)
            {
                rezultat = "Broj je
deljiv sa 6.";
            }
            else
            {
                rezultat = "Broj nije deljiv
sa 6.";
            }
        }

        Console.Clear();
        Console.WriteLine(rezultat);
        Console.ReadLine();
    }
}
```

Višestruka if naredba



```
namespace PovrsinaIObimTrougla
{
    class Program
    {
        double a; // Stranica a
        double b; // Stranica b
        double c; // Stranica c
        double povrsina, obim;
        string strA, strB, strC;
        Boolean flag = false;
        Program()
        {
            unesiPodatke();
            proveraUslovaTrougla();
            if(flag){
                povrsina = izracunajPovrsinu(a, b, c);
                obim = izracunajObim(a, b, c);
            }
            Console.Clear();

            if ( flag == false ){
                Console.WriteLine("UNETE VREDNOSTI NE MOGU BITI STRANICE TROUGLA.");
            }
            else
            {
                Console.WriteLine("Površina trougla je P = "+povrsina);
                Console.WriteLine("Obim trougla je O
= " + obim);
            }
            Console.ReadKey();
        }
    }
}
```

```
private double izracunajObim(double a,
double b, double c)
{
    obim = a + b + c;
    return obim;
}

private double izracunajPovrsinu(double a,
double b, double c)
{
    double poluobim = (a+b+c)/2;
    povrsina = Math.Sqrt( poluobim * (
poluobim - a ) * ( poluobim - b ) * ( poluobim - c
));
    return povrsina;
}

private void proveraUslovaTrougla()
{
    //Provera uslova trougla. Ako stranice ispunjavaju uslov, flag
se postavlja na true.
    if ((a + b > c) & (b + c > a) & (a + c > b))
    {
        flag = true;
    }
}

private void unesiPodatke()
{
    Console.Write("Unesite prvu stranicu
trougla, a = ");
    strA = Console.ReadLine();
}
```

Višestruka if naredba



```
Console.WriteLine("\n\nUnesite drugu stranicu trougla, b = ");
    strB = Console.ReadLine();
    Console.WriteLine("\n\nUnesite treću stranicu trougla, c = ");
    strC = Console.ReadLine();
    a = Convert.ToDouble(strA);
    b = Convert.ToDouble(strB);
    c = Convert.ToDouble(strC);
}

static void Main(string[] args)
{
    Program p = new Program();
}
}
```