Programiranje – III razred

Funkcije

Metode u C#

Metod je imenovani niz naredbi u C# - u

- Prema tome je sasvim sličan pojmu funkcije, procedure ili potprograma u drugim programskim jezicima
- Metod se satoji od deklaracije – prototipa (C, C++) i tela metoda
- Deklaracija metoda obuhvata naziv, povratni tip i listu parametara
- Naziv metoda treba da odgovarajući tako da ukazuje na svrhu i smisao funkcije koju obavlja metod i koja treba da bude dobro definisana
- Telo metoda se sastoji od niza naredbi koje se izvršavaju kada se pozove metod preko svog imena
- Metod može da dobije ulazne podatke pri pozivu i može da vrati podatke

```
returnType methodName ( parameterList )
{
// method body statements go here
}
```

- Povratni tip je podatak koji je rezultat rada metoda i može biti bilo koji tip. Ako nema povratne vrednosti to se označava sa “void” umesto povratnog tipa

Metode u C#

- Sintaksno naziv metoda predstavlja identifikator za koji postoje ista pravila kao i za ime promenljive – takođe se preporučuje pomenuta: Camel notation
- Lista parametra – argumenata metoda se može navesti, ali i ne mora
- Sastoji se od naziva tipova i imena promenljivih kao kod deklaracije između malih zagrada
- Dva ili više argumenata se razdvajaju zarezom
- Telo metoda se sastoji od naredbi koje se nalaze unutar zagrada {}
- C# ne podržava globalne metode kao u C, C++ i VB
- Svaki metod mora biti u nekoj klasi
- Ako metod ima povratnu vrednost, tj. nije void, onda se unutar tela metoda nalazi naredba “return” iz koje sledi vrednost (izraz) koja se vraća
- Tip povratne vrednosti iza “return” se mora slagati po tipu sa deklarisanom vrednošću
- Odmah posle nailaska na “return” i vraćanja vrednosti metoda se završava

Primer sa i bez povratne vrednosti

```
int saberi(int a, int b)
{
// ...
return a + b;
}
```

```
void prikaziRezultat(int rezultat)
{
// display the odgovor
...
return;
}
```

Metode u C#

- Ne postoji minimalna dužina tela metoda
- Telo metoda može biti i prazno {}
- Kad god postoji da se pomoću metoda spreči kopiranje koda, treba koristiti metod
- Metode treba korisiti i kada se program na taj način pojednostavljuje i kada je pregledniji, jasniji – bolje logički struktuiran
- Takođe, ne postoji ni max dužina metoda.
- Ipak, metod ne treba da bude ni neprijatno dugačak pre svega zbog preglednosti i dobro definisane logičke strukture koja treba da bude jedna celina

Poziv metoda

- result = methodName (argumentList)
- Poziv metoda se vrši navođenjem imena metoda – voditi računa da je C# case sensitive i navođenjem eventualne liste parametara u zagradi iza imena metoda
- Dodeljivanje povratne vrednosti metoda nije obavezno, tako da se promenljiva “result” u gornjem primeru može i izostaviti
- Ako je povratna vrednost void, dodela povratne vrednosti predstavlja grešku

```
int saberi(int a, int b)
{
// ...
return a + b;
}
```

- saberi; // compile-time error – nema zagrada
- saberi(); // compile-time error – nema argumenata
- saberi(39); // compile-time error – nema dovoljno argumenata
- saberi(“39”, “3”); // compile-time error – pogrešan tip argumenata

Primer pozivanja

```
int saberi(int a, int b)
{
// ...
return a + b;
}


saberi(39, 3);

int arg1 = 99;
int arg2 = 1;
saberi(arg1, arg2);
```

Opseg važenja identifikatora

- Opseg važenja identifikatora (promenljive i metodi) zavisi od mesta uvođenja – deklaracije identifikatora
- Ako se identifikator može koristiti u određenom delu programa – vidljiv je, onda se kaže da je identifikator u opsegu
- Opseg identifikatora je pre svega određen mestom definisanja u programu
- Zagrade koje definišu telo metoda {} istovremeno određuju i opseg važenja svih promenljivih deklarisanih u tom metodu
- Promenljive deklarisane u jednom metodu su lokalne promenljive i ne mogu se videti – korisiti u telu nekog drugog metoda

Primer opsega važenja

```
class Example
{
void firstMethod()
{
int myVar;
...
}
void anotherMethod()
{
myVar = 42; // error – variable not in scope
...
}
}
```

Primer opsega važenja

```
class Example
{
void firstMethod()
{
myField = 42; // ok
...
}
void anotherMethod()
{
myField++; // ok
...
}
int myField = 0;
}
```

Klasna promenljiva – članica “myField” je vidljiva, može se koristiti iz svih metoda članova iste klase

Opseg važenja identifikatora

- Uopšte, može se reći da svaki blok koda između zagrada { i } predstavlja opseg važenja za identifikatore koji su deklarisani unutar njih

- Promenljive deklarisane unutar bloka koda se ne vide izvan i ne mogu se koristiti

```
{
        int i;
}
i = 5; // error, i se ne vidi van bloka u kome je deklarisana promenljiva int i
```

Overloading – preklapanje metoda

- Ako dva identifikatora imaju ista imena i deklarisana su u istom opsegu, kaže se da su preopterećeni
- Postoje slučajevi kada je to rezultat greške u pisanju programa i takva greška se otkriva kompajliranjem
- Međutim, postoji mogućnost da se namerno definišu metodi koji imaju ista imena ali se razlikuju prema spisku argumenata
- Kao što je već rečeno, metod **WriteLine** klase **Console** ima čak 19 različitih preklopljenih oblika
- Preklapanje metoda se koristi kada metodi vrše istu vrstu operacije nad podacima različitih tipova

Overloading – preklapanje metoda

```
static void Main()
{
Console.WriteLine(“The answer is “);
Console.WriteLine(42);
}
```

- Pošto preklopljeni metodi imaju podatke različitog tipa, i implementacije tih metoda se razlikuju
- Preklopljeni metodi se moraju razlikovati prema listi argumenata – broju i / ili tipovima
- Nije moguće da se preklopljeni metodi razlikuju samo prema tipu povratne vrednosti

Primer preklapanja metoda

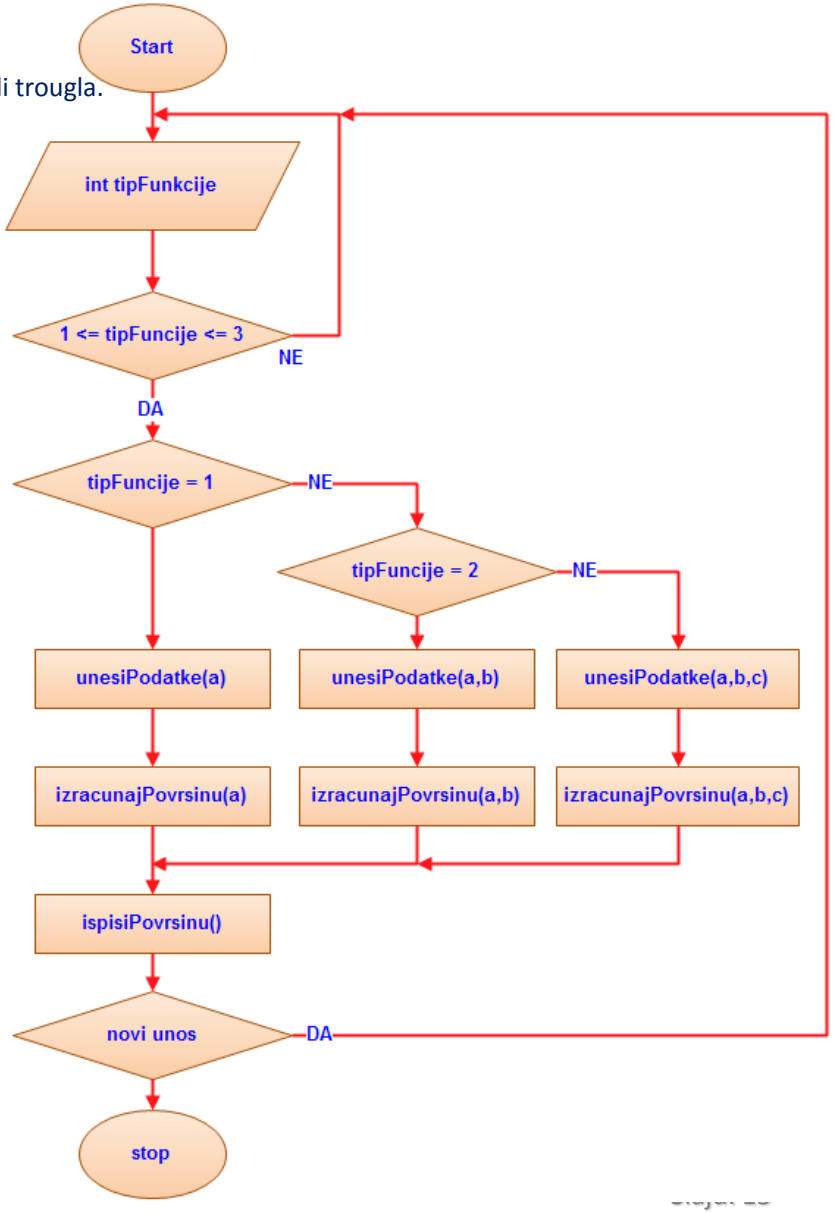
1. Realizovati konzolnu aplikaciju koja će računati površinu kvadrata, pravougaonika ili trougla.

P = a*a

P = a*b

Primer preklapanja metoda

1. Realizovati konzolnu aplikaciju koja će računati površinu kvadrata, pravougaonika ili trougla.



Rešenje primera – prvi deo

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

namespace IzracunavanjePovrsine
{
    class Program
    {
        int tipFunkcije;
        bool ispravanTip = false;
        int a = 0; int b = 0; int c = 0;
        double povrsina = 0;
        Program()
        {
            tipFunkcije = unesiTip();
            proveriOpseg(tipFunkcije); // Provera da li je uneti broj u opsegu od 1 do 3
            if (ispravanTip)
            {
                if (tipFunkcije == 1) // KVADRAT
                {
                    unesiPodatke(a);
                }
                else if (tipFunkcije == 2) // PRAVOUGAONIK
                {
                    unesiPodatke(a, b);
                }
                else
                {
                    unesiPodatke(a, b, c); //TROUGAO
                }
            }
```

Rešenje primera – drugi deo

```
Console.WriteLine("Novo izračunavanje Da / Ne");
            String s = Console.ReadLine();
            if(s == "D" | s == "d")
            {
                Console.Clear();
                new Program();
            }
        }
        private void unesiPodatke(int a, int b, int c)
        {
            Console.Clear();
            Console.WriteLine("*****************************");
            Console.WriteLine("IZRAČUNAVANJE POVRŠINE TROUGLA");
            Console.WriteLine("*****************************\n");
            Console.Write("Unesite prvu stranicu trougla ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("Unesite drugu stranicu trougla ");
            b = Convert.ToInt32(Console.ReadLine());
            Console.Write("Unesite trecu stranicu trougla ");
            c = Convert.ToInt32(Console.ReadLine());

            povrsina = izracunajPovrsinu(a, b, c);
            ispisiPovrsinu(povrsina);
        }

        private void ispisiPovrsinu(double povrsina)
        {
            Console.Clear();
            Console.WriteLine("Površina je: " + povrsina);
            Console.ReadLine();
        }
```

Rešenje primera – treći deo

```
private double izracunajPovrsinu(int a, int b, int c)
{
    double s = 0.5 * (a + b + c);
    double p = Math.Sqrt(s * (s - a) * (s - b) * (s - c));
    return p;
}

private void unesiPodatke(int a, int b)
{
    Console.Clear();
    Console.WriteLine("************************************");
    Console.WriteLine("IZRAČUNAVANJE POVRŠINE PRAVOUGAONIKA");
    Console.WriteLine("************************************\n");
    Console.Write("Unesite prvu stranicu pravougaonika ");
    a = Convert.ToInt32(Console.ReadLine());
    Console.Write("Unesite drugu stranicu pravougaonika ");
    b = Convert.ToInt32(Console.ReadLine());

    povrsina = izracunajPovrsinu(a, b);
    ispisiPovrsinu(povrsina);
}
private double izracunajPovrsinu(int a, int b)
{
    return a * b;
}
```

Rešenje primera – četvrti deo

```
private void unesiPodatke(int a)
        {
            Console.Clear();
            Console.WriteLine("*******************************");
            Console.WriteLine("IZRAČUNAVANJE POVRŠINE KVADRATA");
            Console.WriteLine("*******************************\n");
            Console.Write("Unesite stranicu kvadrata ");
            a = Convert.ToInt32(Console.ReadLine());

            povrsina = izracunajPovrsinu(a);
            ispisiPovrsinu(povrsina);
        }

        private double izracunajPovrsinu(int a)
        {
            return a * a;
        }

        private void proveriOpseg(int tipFunkcije)
        {
            if (tipFunkcije == 1 | tipFunkcije == 2 | tipFunkcije == 3)
            {
                ispravanTip = true;
            }
            else
            {
                Console.Clear();
                new Program();
            }
        }
```

Rešenje primera – peti deo

```
        private int unesiTip()
        {
            Console.WriteLine("Odaberite površinu unosom broja ispred:\n");
            Console.WriteLine("1 - Kvadrat");
            Console.WriteLine("2 - Pravougaonik");
            Console.WriteLine("1 - Trougao");
            string s = Console.ReadLine();
            return Convert.ToInt32(s);
        }


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

Zadatak za vežbu

1. Realizovati konzolnu aplikaciju koja će računati obim kruga, pravougaonika ili trougla.

Programiranje – III razred

Funkcije