

Преоптерећене функције

Пример: Написати програм који помоћу преоптерећења функције Prikaz (са различитим бројем параметара) приказује вредности аргумената које се користе при позиву функције.

```
using System;
namespace Proba
{
    class Program
    {
        static void Prikaz(int a)
        {
            Console.WriteLine("Argumenti: " + a);
        }
        static void Prikaz(int a, int b)
        {
            Console.WriteLine("Argumenti: " + a + " i " + b);
        }
        static void Main(string[] args)
        {
            Prikaz(100);
            Prikaz(100, 200);
        }
    }
}
```

Дaje:

Argumenti: 100

Argumenti: 100 i 200

Пример: Написати програм који помоћу преоптерећења функције Prikaz (са различитим типом параметара и различитим распоредом параметара) приказује вредности аргумената које се користе при позиву функције.

```
using System;
namespace Proba
{
    class Program
    {
        static void Prikaz(int a, string s)
        {
            Console.WriteLine($"Argumenti int {a} i string {s}.");
        }
        static void Prikaz(string s, int a)
        {
            Console.WriteLine($"Argumenti string {s} i int {a}.");
        }
        static void Prikaz(string s1, string s2)
        {

```

```

        Console.WriteLine($"Argumenti string {s1} i string
                           {s2}.");
    }
    static void Main(string[] args)
    {
        Prikaz(100, "Skola");
        Prikaz("Skola", 200);
        Prikaz("Moja", "Skola");
    }
}

```

Дaje:

Argumenti int 100 i string Skola.

Argumenti string Skola i int 200.

Argumenti string Moja i string Skola.

Пример: Написати програм који коришћењем делегата функција исписује поруке на екрану. Употребити и циљну функцију и ламбда израз.

```

using System;
namespace Proba
{
    class Program
    {
        delegate void MojDelegat(string poruka);
        static void Main(string[] args)
        {
            MojDelegat x = Funkcija1;
            x("Zdravo Svete!");
            x = Funkcija2;
            x("Da!");
            x = (string poruka) => Console.WriteLine("Pozvan lambda
                                                       izraz: " + poruka);
            x("Lepo!");
        }
        static void Funkcija1(string por)
        {
            Console.WriteLine("Pozvana Funkcija1() sa parametrima: " +
                               por);
        }
        static void Funkcija2(string por)
        {
            Console.WriteLine("Pozvana Funkcija2() sa parametrima: " +
                               por);
        }
    }
}

```

Даје:

Pozvana Funkcija1() sa parametrima: Zdravo Svete!

Pozvana Funkcija2() sa parametrima: Da!

Pozvan lambda izraz: Lepo!

Да би се функција предала као параметар, најчешће се користе делегати (тип података по референци) са којима се дефинишу потписи функција.

Потребно је декларисати делегата, поставити циљна функција (или ламбда израз) и покренути делегата.

У примеру, са линијом `delegate void MojDelegat(string poruka);` се декларише делегат `MojDelegat`, а потпис делегата функције је цео израз без службене речи `delegate`.

Унутар главне функције се поставља циљна функција: `MojDelegat x = Funkcija1;` али и друга циљна функција: `x = Funkcija2;`

Да би циљна функција била делегат мора да у потпуности одговара потпису делегата функције.

У главној функцији је постављен и ламбда израз: `x = (string poruka) =>`

`Console.WriteLine("Pozvan lambda izraz: " + poruka);` којим се на други начин постављају и реализују параметри за делегата `x`.

Задаци за самосталан рад:

1. Написати програм који помоћу преоптерећења функције барата са две целобројне вредности или са две реалне вредности, позивајући функције за сабирање и одузимање. Програм одлучује о унетом знаку операције преко switch-case структуре.
2. Написати програм који коришћењем делегата функција исписује резултате сабирања два цела броја. Употребити циљну функцију.