

Досег промењивих

Разлог зашто је неопходно размењивати податке са функцијама је тај што су промењиве у C# доступне само кроз локализоване делове кода.

За дату промењиву се каже да има досег (scope) унутар којег је доступна.

Пример:

```
using System;
namespace Proba
{
    class Program
    {
        static void Main(string[] args)
        {
            string mojString = "String definisan u Main()";
            Ispisivanje();
        }
        static void Ispisivanje()
        {
            Console.WriteLine($"mojString = {mojString}");
        }
    }
}
```

После компајлирања кода појављују се два типа грешке:

The name 'mojString' does not exist in the current context

и

The variable 'mojString' is assigned but its value is never used

Промењива mojString која је дефинисана у функцији Main() није доступна из функције Ispisivanje(). Разлог за недоступност је та што промењива има досег унутар друге програмске структуре.

Овај досег заокружује блок кода у којем је дефинисана промењива и све директно нестоване кодове блока.

Блокови кода у функцијама су одвојени од блокова кода из којих се те функције позивају.

Унутар функције Ispisivanje(), име mojString није дефинисано, а промењива mojString дефинисана у Main() је изван досега (out of scope) и може се користити само унутар функције Main().

Заправо, могуће је користити потпуно различиту промењиву која се зове mojString унутар функције Ispisivanje().

Пример:

```
using System;
namespace Proba
{
    class Program
    {
        static void Main(string[] args)
        {
            string mojString = "String definisan u Main()";
            Ispisivanje();
        }
    }
}
```

```

        Console.WriteLine("\nSada u Main()");
        Console.WriteLine($"mojString = {mojString}");
    }
    static void Ispisivanje()
    {
        string mojString = "String definisan u Ispisivanje()";
        Console.WriteLine("Sada u Ispisivanje()");
        Console.WriteLine($"mojString = {mojString}");
    }
}

```

Дaje:

Sada u Ispisivanje()

mojString = String definisan u Ispisivanje()

Sada u Main()

mojString = String definisan u Main()

Операције које се одигравају приликом покретања кода:

- Main() дефинише и иницијализује стринг промењиву са именом mojString
- Main() пребацује контролу на Ispisivanje()
- Ispisivanje() дефинише и иницијализује стринг промењиву под именом mojString, која је другачија промењива од промењиве mojString дефинисане у Main()
- Ispisivanje() приказује Sada u Ispisivanje() на екрану
- Ispisivanje() приказује стринг на екрану који садржи вредност промењиве mojString како је дефинисано у Ispisivanje()
- Ispisivanje() пребацује контролу назад у Main()
- Main() приказује Sada u Main() на екрану
- Main() приказује стринг на екрану који садржи вредност mojString како је дефинисано у Main()

Промењиве чији досег обухвата само једну функцију се називају локалне промењиве.

Промењиве чији досег обухвата више функција се називају глобалне промењиве.

Пример:

```

using System;
namespace Proba
{
    class Program
    {
        static string mojString;
        static void Main(string[] args)
        {
            string mojString = "String definisan u Main()";
            Program.mojString = "Globalni string";
            Ispisivanje();
        }
    }
}

```

```

        Console.WriteLine("\nSada u Main()");
        Console.WriteLine($"Lokalni mojString = {mojString}");
        Console.WriteLine($"Globalni mojString =
            {Program.mojString}");
    }
    static void Ispisivanje()
    {
        string mojString = "String definisan u Ispisivanje()";
        Console.WriteLine("Sada u Ispisivanje()");
        Console.WriteLine($"Lokalni mojString = {mojString}");
        Console.WriteLine($"Globalni mojString =
            {Program.mojString}");
    }
}

```

Дaje:

Sada u Ispisivanje()

Lokalni mojString = String definisan u Ispisivanje()

Globalni mojString = Globalni string

Sada u Main()

Lokalni mojString = String definisan u Main()

Globalni mojString = Globalni string

Да би се могла мењати вредност глобалне промењиве, мора се користити службена реч `static` у декларацији глобалне промењиве.

Друга опција је `const` али у том случају се глобална промењива не сме мењати у коду.

Унутар функција кода, приступа се глобалној промењивој преко њеног пуног квалификованог имена – `Program.mojString`.

Овиме се спречавају проблеми ако постоје локалне промењиве истог имена као и глобална промењива.