

Контрола приступа

У досадашњим примерима није било велике користи од класа.

Када се напишу методе и подаци унутар класе, класа формира границе према спољном свету.

Ако су методе и подаци декларисане као `private`, ниједна друга класа не може прићи нити променити вредности у њима.

Модификатор `private` је дифолтни модификатор за класе па када се ниједан модификат не пише, подразумева се `private`.

Пример:

```

class Krug
{
    private int r;
    public double RacunanjePovrsine()
    {
        return Math.PI * r * r;
    }
}

```

Чак иако је поље `r` декларисано као `private`, метода може да приђе пољу и да га користи пошто је то поље доступно свим члановима класе који су декларисани унутар класе.

Обичај је да се поља и методе који су `private` пишу малим словом а `public` првим великим словом.

Пример:

```

using System;
namespace Proba
{
    class Program
    {
        static void Main() {}
    }
    class Krug
    {
        int r;
        double RacunanjePovrsine()
        {
            return Math.PI * r * r;
        }
    }
}

```

Тело класе садржи методу `RacunanjePovrsine()` и поље `r`.

Метода користи поље класе и константу `PI` за добијање резултата рачунањем површине круга.

Тај резултат је излаз из методе и по томе се види сличност метода и функција.

Разлика је у томе што се методе морају дефинисати унутар неке класе и директно зависе од те класе.

Методe инстанци

Алтернативни начин приказа на конзоли коришћењем методе `WriteLine()` јесте креирање методе унутар друге класе која ће дефинисати нов начин форматирања излаза.

Промена функционалности унутар друге класе је практичнији приступ и више је окренут енкапсулацији података.

Пример:

```
public class Zaposleni
{
    public string ime;
    public string prezime;
    public string plata;
    public string dobijanjeImena()
    {
        return ("{"ime} {"prezime}");
    }
}
```

Сада метода `dobijanjeImena()` приступа директно пољу објекта уместо локалним промењивима. Уз то, метод није декларисан као `static`, пошто на тај начин се не би могло приступити директно пољима инстанце унутар класе.

Сада је омогућено користити инстанцу класе преко позива било којег члана инстанце, без обзира да ли је метода или поље.

Пример:

```
#nullable enable
using System;
namespace Proba
{
    class Program
    {
        static void Main()
        {
            Zaposleni radnik1 = new Zaposleni();
            Zaposleni radnik2;
            radnik2 = new Zaposleni();
            radnik1.Ime = "Miki";
            radnik1.Prezime = "Mikic";
            radnik1.Plata = "Mala";
            Console.WriteLine($"{radnik1.dobijanjeImena()}:
                               {radnik1.Plata} plata.");
        }
    }
}

public class Zaposleni
{
    public string Ime;
    public string Prezime;
    public string Plata;
    public string dobijanjeImena()
    {
```

```

        return ($"{Ime} {Prezime}");
    }
}

```

Дaje:

Miki Mikic: Mala plata.

Декларисање методе унутар Main() функције омогућава додатну контролу над вредностима поља и метода других класа, ако су то поља инстанце и методе инстанце.

Пример:

```

#nullable enable
using System;
namespace Proba
{
    class Program
    {
        static void Main()
        {
            Zaposleni radnik1 = new Zaposleni();
            Zaposleni radnik2;
            radnik2 = new Zaposleni();
            radnik1.Ime = "Miki";
            radnik1.Prezime = "Mikic";
            radnik1.Plata = "Mala";
            radnik2.Ime = "Ana";
            radnik2.Prezime = "Anic";
            radnik2.Plata = radnik1.Plata;
            PovecanjePlate(radnik2);
            Console.WriteLine($"{radnik1.dobijanjeImena()}:
{radnik1.Plata} plata.");
            Console.WriteLine($"{radnik2.dobijanjeImena()}:
{radnik2.Plata} plata.");
        }
        static string PovecanjePlate(Zaposleni x)
        {
            return (x.Plata = "Nesto veka");
        }
    }
    public class Zaposleni
    {
        public string Ime;
        public string Prezime;
        public string Plata;
        public string dobijanjeImena()
        {
            return ($"{Ime} {Prezime}");
        }
    }
}

```

```
    }  
  }  
}
```

Даје:

Miki Mikic: Mala plata.

Ana Anic: Nesto veca plata.

Задаци за самосталан рад

1. Класа Krug дефинише јавно поље r и јавну методу `RacunanjePovrsine`, У главној методи инстанцирати класу објектом x који иницијализује поље на 2.5 па израчунати површину круга са тим полупречником.
2. Написати програм са класом која рачуна и приказује константну брзину два аутомобила ако су њихови пређени путеви 100 и 1000 метара за 5 и 20 секунди.
3. Посебна класа декларише име, презиме и плату у реалним вредностима запосленог. У главној класи реализовати методу која модификује плату запосленог множењем са фактором 1.5. Приказати податке о запосленом са модификованом платом.