
Tema 10

KLASA

- fabrika po cijim pravilima se stancaju objekti, primjerci
- sablon za opis zajednickih karakteristika grupe objekata
- generalizacija grupe tj zanemarivane specificnosti pojedinaca

OBJEKAT

- jedan proizvod fabrike
- ima sva svojstva definisana fabrickim podesavanjima

SUMA ČASA

INDIKTORI PRISTUPA

- PRIVATE** - Niko van klase ne moze da pristupi tom podatku
- PUBLIC** - Bilo ko van klase moze da pristupi tom podatku
- PROTECTED** - Dostupan samo unutar svoje klase i izvedenih klasa
- PROTECTED INTERNAL** - Dostupan samo unutar klase, unutar izvedenih klasa i unutar klasa u istom programu
- INTERNAL** - Dostupan samo unutar klase i unutar klasa u istom programu

DEFINICIJA KLASA

```
class NAZIV KLASE {...}
```

ATIBUTI

METODE

- konstruktor
- destruktor
- ostale funkcije

this - pristup kroz objekat klase

static - pristup bez da postoji instanca klase

new Objekat

```
imeObjekta.ImeAtributa  
imeObjekta.NazivMetode(stvarniArgumenti)
```

ZADACI

Zadatak 1

Napisati program kojim se simulira klasa nastavnik i klasa ucenik.

Napraviti nekoliko objekata ovih klasa

Napraviti primer sa static atributom i metodom.

Napraviti getter i setter

Napraviti drudge funkcije sa i bez argumenata.

PRIMER 1 - Osnovni nacin deklarisanja klase.

```
using System;

namespace primerKlase1
{
    class Program
    {
        static void Main(string[] args)
        {
            Ucenik ana = new Ucenik("Ana", "Golubovic", "VI", 2008);
            Console.WriteLine(Ucenik.UkupnoUcenika());
            ana.PrikaziStudenta();
            Console.WriteLine(ana.ImeIPrezime());
            Console.WriteLine(ana.BrojPredmeta());

            Ucenik tara = new Ucenik();
            Console.WriteLine(Ucenik.UkupnoUcenika());
            tara.Ime = "Tara";
            tara.Prezime = "Milosevic";
            tara.Razred = "VIII";
            tara.Godiste = 2006;

            tara.PrikaziStudenta();
            Console.WriteLine(tara.Godine());
            Console.WriteLine(tara.BrojPredmeta());

            GC.Collect();
            // samo primer da se pokaze da se tako eksplicitno pozove garbage
            collector, ali nije potreban jer se automatski aktivira
        }
    }

    class Ucenik
    {
        private string ime;
        private string prezime;
        private string razred;
        private int razredBroj;
    }
}
```

```

private int godiste;

private static int brojacUcenika = 0;
private static string skola = "OS Petar Petrovic Njegos";
private static int[] brojPredmeta = new int[8]{ 7, 8, 8, 9, 13, 13,
14, 15 };

// sta jos moze biti static?
// 1) brojac
// 2) univarzalni podatak za sve ove studente, npr ime skole kojoj
pripadaju, time se stedi memorija

public Ucenik(string _ime, string _prezime, string _razred, int
_godiste)
{
    this.ime = _ime;
    this.prezime = _prezime;
    this.razred = _razred;
    this.godiste = _godiste;
    brojacUcenika++;
    this.razredBroj = Ucenik.PodesiRazred(_razred);
    // moze i bez this
    // this govori da se ova dodjela odnosi na ovaj primjerak klase
}

// podrazumevani konstruktor
public Ucenik()
{
    brojacUcenika++;
}

~Ucenik()
{
    Console.WriteLine("Aktiviran Destruktor");
}

public string Ime
{
    get
    {
        return ime;
    }
    set
    {
        ime = value;
    }
}

public string Prezime
{
    get
    {
        return prezime;
    }
    set
    {
        prezime = value;
    }
}

public string Razred

```

```

    {
        get
        {
            return razred;
        }
        set
        {
            razred = value;

            razredBroj = PodesiRazred(razred);
        }
    }

public int Godiste
{
    get
    {
        return godiste;
    }
    set
    {
        godiste = value;
    }
}

public void PrikaziStudenta()
{
    Console.WriteLine("{0} {1}, razred {2}, godiste {3}", prezime,
ime, razred, godiste);
}

public string ImeIPrezime()
{
    return this.Ime + " " + this.Prezime;
}

public int Godine() // nema onog self kao u pythonu
{
    return 2021 - this.Godiste;
}

public int BrojPredmeta()
{
    return brojPredmeta[this.razredBroj - 1];
}

public static int UkupnoUcenika()
{
    return brojUcenika;
}

private static int PodesiRazred(string razred)
{
    int razredBroj;
    switch(razred)
    {
        case "I":
            razredBroj = 1;
            break;
    }
}

```

```

        case "II":
            razredBroj = 2;
            break;
        case "III":
            razredBroj = 3;
            break;
        case "IV":
            razredBroj = 4;
            break;
        case "V":
            razredBroj = 5;
            break;
        case "VI":
            razredBroj = 6;
            break;
        case "VII":
            razredBroj = 7;
            break;
        case "VIII":
            razredBroj = 8;
            break;
        default:
            razredBroj = 0;
            break;
    }
    return razredBroj;
}
}
}

```

NAPOMENA 1: Istraziti static fore zasebno

NAPOMENA 2: Istraziti getere zasebno

NAPOMENA 3: Istraziti kreiranje obicne funkcije zasebno

NAPOMENA 4: Prisetiti se switch naredbe

NAPOMENA 5: Povuci paralelu sa self iz pythona, constructor umjesto init.