

Objekti i klase

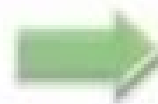
Primeri objekata:



Klase

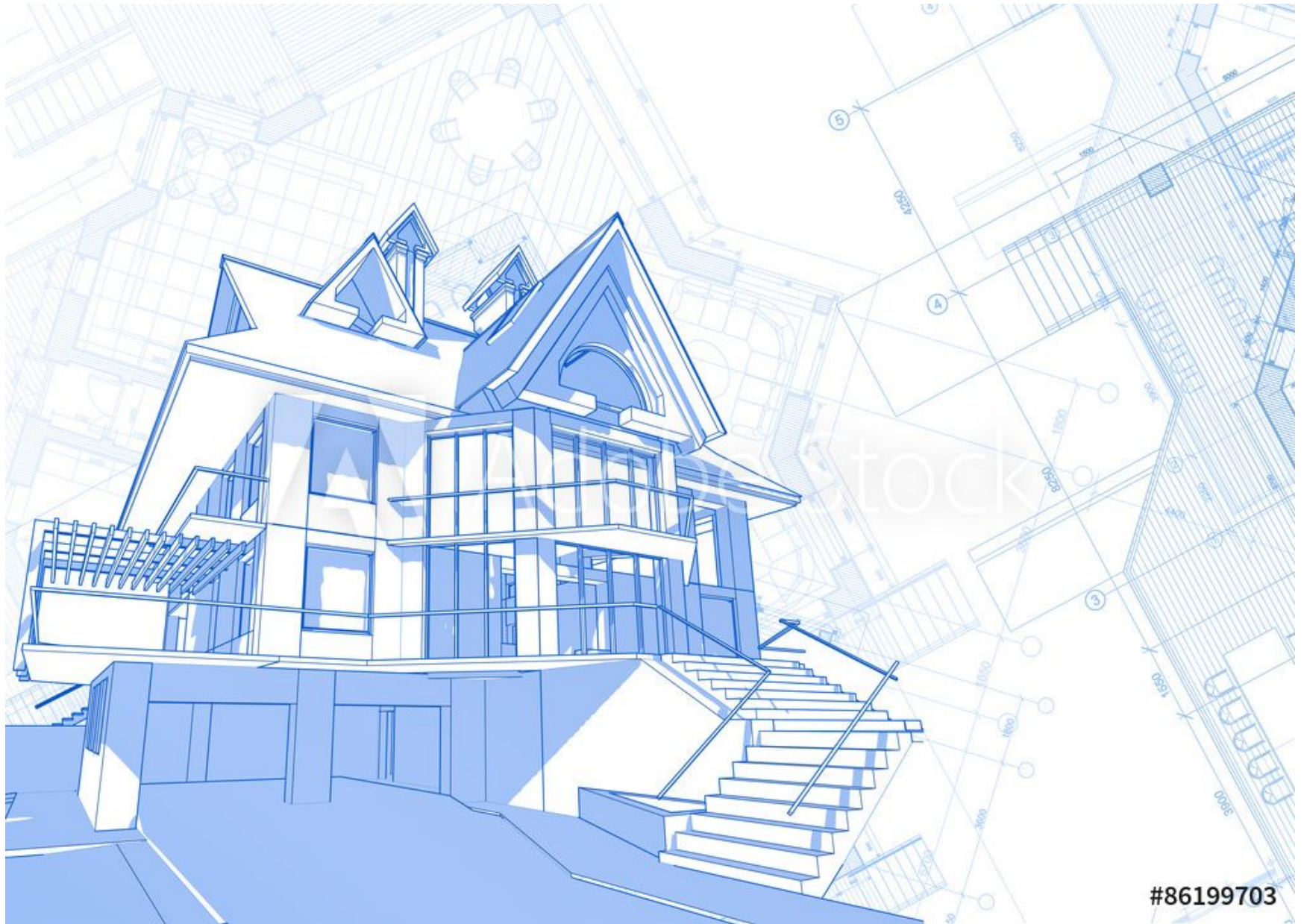


Klasa je složeni tip podataka koji predstavlja predložak prema kojem se izrađuju objekti. Klasom se definiraju osobine i ponašanje tih objekata.



Zajednička obilježja

- Ime
- Prezime
- Starost



#86199703



Stvaranje novih objekata

```
var polje = new int[7];
```



Operator za stvaranje
(instanciranje) novih
objekata

Osobine klasa



Osoba

- Ime
- Prezime
- Starost

Program

Main()

```
using System;

class Program
{
    static void Main()
    {
        var osoba1 = new Osoba();
        var osoba2 = new Osoba();
    }
}
```

```
using System;

class Program
{
    static void Main()
    {
        var osoba1 = new Osoba();
        var osoba2 = new Osoba();

        osoba1.Ime = "Ivo";
        osoba2.Ime
    }
}
```

Ime

- Properties
- References
- Osoba.cs
- Program.cs

Osoba.cs

Program.cs X

Program

Main()

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        var osoba1 = new Osoba();
```

```
        var osoba2 = new Osoba();
```

```
        osoba1.Ime = "Ivo";
```

```
        osoba2.Ime = "Ana";
```

```
        Console.WriteLine(osoba1.Ime);
```

```
        Console.WriteLine(osoba2.Ime);
```

```
    }
```

```
}
```

Solution Explorer

Solution 'Dump.CSh

Demo1

Properties

References

Osoba.cs

Program.cs

100 %

Solution Explorer

Anatomija metode

```
class Osoba
```

```
{  
    public string Ime;  
    public string Prezime;  
    public int Starost;  
    string VratiPunoIme()  
    {  
        return Ime + " " + Prezime;  
    }  
}
```

Konstruktor

```
class Osoba
```

```
{  
    public string Ime;  
    public string Prezime;  
    public int Starost;
```

```
    public Osoba( )
```

```
{  
    }  
}
```



Defaultni konstruktor

Konstruktor

```
class Osoba
```

```
{  
    public string Ime;  
    public string Prezime;  
    public int Starost;
```

```
    public Osoba(string ime, string prezime, int starost )  
    {  
    }  
}
```

Primer kreiranja

```
var osoba = new Osoba( );
```

Konstruktor

```
class Osoba
```

```
{  
    public string Ime;  
    public string Prezime;  
    public int Starost;
```

```
    public Osoba( string ime, string prezime, int starost )
```

```
{  
        Ime = ime;  
        Prezime = prezime;  
        Starost = starost;  
    }
```

```
}
```

Primer korišćenja

```
var osoba = new Osoba( "Ivan", "Horvat", 25 );
```

```
class Osoba
{
    public string Ime;
    public string Prezime;
    private int Starost;

    public string VratiPunoIme()
    {
        return Ime + " " + Prezime;
    }
}
```

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        var osoba1 = new Osoba();
```

```
        osoba1.Ime = "Ivo";
```

```
        osoba1.Prezime = "Ivic";
```

```
        Console.WriteLine(osoba1.Ime);
```

```
    }
```

```
}
```

Osoba.cs

Program.cs X

Program

Main()

```
using System;

class Program
{
    static void Main()
    {
        var osoba1 = new Osoba();

        osoba1.Ime = "Ivo";
        osoba1.Prezime = "Ivic";
        Console.WriteLine(osoba1.VratiPunoIme());
    }
}
```

Solution Explorer

Solution 'Dump.CShar

Demo1

Properties

References

Osoba.cs

Program.cs

100 %

Solution Explorer

Osoba.cs

Program.cs X

Program

Main()

```
using System;

class Program
{
    static void Main()
    {
        var osoba1 = new Osoba();

        osoba1.Ime = "Ivo";
        osoba1.Prezime = "Ivic";

        var punoIme = osoba1.VratiPunoIme();

        Console.WriteLine(punoIme);
    }
}
```

Solution Explorer

Solution 'Dump.CSh

Demo1

Properties

References

Osoba.cs

Program.cs

100 %

Solution Explorer

```
class Osoba
{
    public string Ime;
    public string Prezime;
    private int Starost;

    public Osoba(string ime, string prezime, int starost)
    {
        Ime = ime;
        Prezime = prezime;
        Starost = starost;
    }

    public string VратиPunoIme()
    {
        return Ime + " " + Prezime;
    }
}
```

Osoba.cs

Program.cs X

Program

Main()

```
using System;

class Program
{
    static void Main()
    {
        var osoba1 = new Osoba("Ivo", "Ivic", 25);
        var punoIme = osoba1.VratiPunoIme();

        Console.WriteLine(punoIme);
    }
}
```

100 %

Solution Explorer

Solution 'Dump.CSh

Demo1

Properties

References

Osoba.cs

Program.cs

Solution Explorer

```
        Ime = ime;
        Prezime = prezime;
        Starost = starost;
    }

    public string VратиPunoIme()
    {
        return Ime + " " + Prezime;
    }

    public void PostaviIme(string ime)
    {
        Ime = ime;
    }

    public string DohvatiIme()
    {
        return Ime;
    }
}
```

Osoba.cs*

Program.cs* X

Solution Explorer

Program

Main()

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        var osoba1 = new Osoba("Ivo", "Ivic", 25);
```

```
        osoba1.PostaviIme("Ante");
```

```
        Console.WriteLine(osoba1.);
```

```
    }
```

```
}
```

DohvatiIme

Equals

GetHashCode

GetType

PostaviIme

Prezime

ToString

VratiPunoIme

void Osoba.PostaviIme(string ime)

100 %

Solution Explorer