

КЛАСЕ

Дефиниција класе:

Класа као реч је настала према појму класификација.

По дефиницији: класа је кориснички, комплексни, сложени тип податка који је дефинисан од стране програмера да опише тј. симулира реално постојеће објекте. Састоји од вредносних и референтних променљивих, као и метода (функције, акције) за рад са тим променљивима.

Класе у основи представљају **шаблоне** из којих се могу стварати објекти, **шаблони за објекте**. Дакле, класе описују објекте, а објекти су конкретни **примерци** или **инстанце** класе.

151. Дате су програмске јединице у програмском језику С#. Одредити која програмска јединица представља шаблон за конструисање објекта истог типа:

1. Класа
2. Именски простор
3. Нестатичка метода
4. Променљива
5. Статичка метода

1

Компоненте класе су атрибути и методе, ове компоненте се заједничким именом називају **ЧЛАНОВИ КЛАСЕ**. Дакле, можемо да напишемо да се класе састоје од:

- атрибута или поља
- метода или акција

Атрибути или поља класе су заправо променљиве у класи.

Методе класе тј. функције објеката описују акције које могу неки објекти да обављају.

Синтакса класе:

Класу дефинишемо навођењем кључне речи **class** иза које следи идентификатор класе (*ime_klase*). Овај део се назива **ЗАГЛАВЉЕ КЛАСЕ**.

Затим у витичастим заградама { } дефинишемо **чланове класе** (атрибути и методи). Овај део називамо **ТЕЛО КЛАСЕ**.

```
class ime_klase
{
    definicija clanova klase
}
```

Пример класе **Krug**.

```
public class Krug
{
    double r;                                // atribut klase

    public double Povrsina(double r)          // metoda klase
    {
        return r * r * Math.PI;
    }

    public double Obim(double r)              // metoda klase
    {
        return 2 * r * Math.PI;
    }
}
```

Шта је објекат и како се креира објекат класе?

Као што је наведено објекат је конкретни примерак или инстанца класе.

Да би објекат могао да се користи (да се позивају његове методе, мењају вредности атрибута ...) потребно га је иницијализовати. Иницијализација објекта се врши коришћењем наредбе **"new"**. Дакле, за креирање објекта потребно је користити кључну реч **"new"**.

Када се декларише неки објекат, само се створи показивач који ће да референцира објекат. Али када се изврши иницијализација, тек онда се алоцира део меморије рачунара (део RAM меморије) у коме ће да буде објекат и повеже се са показивачем – показивач у том тренутку садржи адресу меморијске локације објекта. Тек након овога се објекат може користити.

Синтакса креирања објекта класе.

```
NazivKlase nazivobjekta = new NazivKlase();
```

Модификатори приступа

Модификатори приступа у језику C# служе да одреде које методе и атрибуте могу да виде и користе друге класе. Ако за чланове класе не наведемо модификатор приступа подразумева се да је **private**.

Генерално у C# - у разликујемо 5 модификатора приступа:

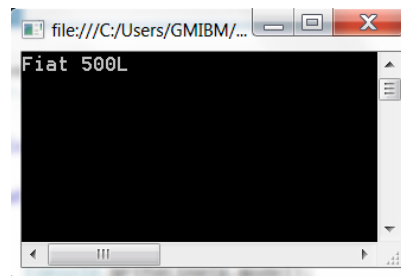
- **public** – чланови класе су јавни, тј. приступ није ограничен
- **private** – чланови класе су видљиви само унутар класе у којој су дефинисани.
- **protected** – чланови класе су видљиви унутар класе у којој су дефинисани и унутар изведених класа.
- **internal** – чланови класе су видљиви унутар класе и пројекта у коме су дефинисани.
- **protected internal** – чланови класе су видљиви унутар класе, изведених класа и пројекта у коме су дефинисани.

ЗАДАТАК:

Илустровати рад са модификатором приступа **private**.

```
using System;
class Program
{
    private string model = "Fiat 500L";

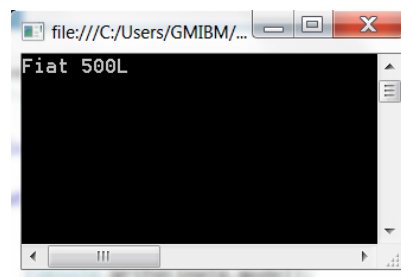
    static void Main(string[] args)
    {
        Program p = new Program();
        Console.WriteLine(p.model);
        Console.ReadKey();
    }
}
```

**ЗАДАТАК:**

Илустровати рад са модификатором приступа **public**.

```
using System;
class Program
{
    static void Main(string[] args)
    {
        Automobil p = new Automobil();
        Console.WriteLine(p.model);
        Console.ReadKey();
    }
}

class Automobil
{
    public string model = "Fiat 500L";
}
```

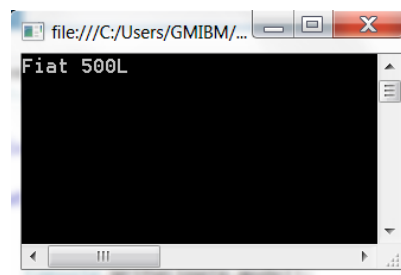
**ЗАДАТАК:**

Илустровати рад са модификатором приступа **protected**.

```
using System;
class Automobil
{
    protected string car = "Fiat 500L";
    private string car2 = "Audi A3";
}

class Model : Automobil
{
    public void ispisi()
    {
        Console.WriteLine(car);
        //Console.WriteLine(car2);
    }
}

class Program
{
    static void Main()
    {
        Model m = new Model();
        m.ispisi();
        Console.ReadKey();
    }
}
```



ЗАДАТАК:

Илустровати рад са модификатором приступа **internal**.

```
using System;
using Automobili;

namespace Projekat
{
    class Program
    {
        static void Main(string[] args)
        {
            Car c = new Car();
            Motor m = new Motor();
            Console.WriteLine(c.car1);
            //Console.WriteLine(c.car2);
            Console.WriteLine(m.kawasaki);
            Console.ReadKey();
        }
    }
    class Motor {

        internal string kawasaki = "Z H2";

    }
}
```

```
using System;

namespace Automobili
{
    public class Car
    {
        public string car1 = "Fiat 500L";
        internal string car = "Ford Kuga";
    }
}
```

