

МОДИФИКАТОР STATIC

Основне карактеристике:

- Модификатор **static** означава нешто заједничко у свим инстанцама дате класе.
- Сва статичка поља су заједничка и јединствена за све креиране објекте неке класе.
- Не статичко поље је разликује се за сваки појединачни објекат неке класе.
- Ако у класи испред типа поља или методе не наведемо модификатор **static** онда се подразумева да је поља не статичко.
- Не - статичко поље, не може да се користи у статичкој методи.

1. Задатак: Анализирати програмски код и проценити шта ће се приказати.

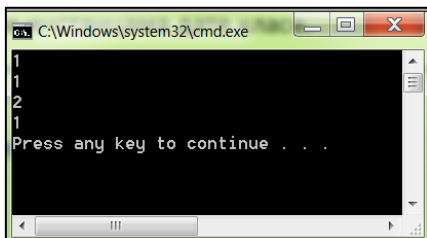
Свака инстанца класе **Broj**, дакле објекат **u1** и **u2** ће имати своју променљиву **broj2**.

Међутим променљиву **broj1** ће све инстанце делити.

Ако један објекат измени вредност променљиве **broj2** и сви остали ће имати измењену вредност.

Приступ статичкој променљивој:

imeklase.imepromenljive



```
using System;

class Broj {
    public static int broj1 = 0;
    public int broj2 = 0;

    public Broj()
    {
        broj1++;
        broj2++;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Broj u1 = new Broj();
        Console.WriteLine(Broj.broj1);
        Console.WriteLine(u1.broj2);
        Broj u2 = new Broj();
        Console.WriteLine(Broj.broj1);
        Console.WriteLine(u2.broj2);
    }
}
```

162. Која су поља заједничка и јединствена за све креиране објекте неке класе дефинисане у објектно оријентисаном програмском језику C#:

1. Јавна
2. Приватна
3. Објектна
4. Инстанцна
5. Статичка

1

149. У програмском језику C# класа може да садржи статичка и не-статичка (инстанцна) поља. Дате су изјаве које се односе на статичка поља класе и међу њих је уметнута једна изјава која се односи на не-статичка (инстанцна) поља класе.

Заокружити број испред изјаве која се односи на не-статичка поља класе:

1. Поље које се може користити без конструисања и једног објекта те класе
2. Поље које има исту вредност за све креиране објекте неке класе
3. Поље чија се вредност може разликовати за сваки појединачни објекат неке класе
4. Поље које се може користити унутар статичких метода класе, као и унутар метода инстанце

1

2. Задатак: Анализирати програмски код и објаснити да ли методе могу – не могу бити статичке.

```
using System;

public class Test{
    private int broj;
    public int kvadrant(int n) { return n * n; }
    public int getBroj() { return broj; }
}
class Program
{
    static void Main(string[] args)
    {
    }
}
```

АНАЛИЗИРАТИ ПРИМЕРЕ И ЗАОКРУЖИТИ ТАЧАН ОДГОВОР:

193. Дата је дефиниција класе у програмском језику C#. Проценити где у дефиницији класе (испред које методе) треба заменити знакове ??? службеном речју **static**.

```
1. public class Test {
2.     private int broj;
3.
4.     public ??? int kvadrant(int n) { return n * n; }
5.     public ??? int getBroj() { return broj; }
6. }
```

2

Заокружити број испред тачне изјаве:

1. Метода **kvadrant** МОРА да буде статичка, док метода **getBroj** може и не мора.
2. Обе методе морају бити статичке.
3. Ни једна од дефинисаних метода није статичка.
4. Метода **getBroj** НЕ СМЕ да буде статичка, док метода **kvadrant** може и не мора.

204. Дат је код програма у програмском језику C# који дефинише апстрактну класу **Figura**. У класи **Figura** написати статички иницијални блок у ком се иницијализује вредност атрибута **brojFigura**. Заокружити број испред наредбе коју треба додати у 7. линију кода, којом се иницијализује вредност статичког атрибута **brojFigura**.

```
1. public abstract class Figura {
2.     protected static int brojFigura;
3.     protected string naziv;
4.     public Figura(string naziv) {
5.         this.naziv = naziv;
6.     }
7.     public abstract double površinaFigure();
8.     public abstract double obimFigure();
9.
10.     public static void UvecajBrojFigura() {
11.         brojFigura++;
12.     }
13.     public override string ToString() {
14.         return "Figura ";
15.     }
16. }
```

2

Заокружити број испред одговора који даје исправно решење:

1. static Figura() { brojFigura = 0; }
2. brojFigura = 0;
3. static { brojFigura = 0; }
4. public Figura() { brojFigura = 0; }

Ако је класи потребан статички блок тј. **brojFigura**, а ово јесте статички блок (погледати 2. линију), потребно је да метода буде статичка, тј. `static Figura() { brojFigura = 0; }`. Дакле, решење је одговор под 1.

202. Дат је код програма у програмском језику C# којим су дефинисане три класе: `class Program` која садржи `Main(string[] args)` методу, `class Vozilo` и `class Auto`. Компајлер приликом превођења овог кода јавља грешку. Заокружити број испред одговора у коме је написано који редови су изазвали грешку приликом компајлирања.

```
1. class Program {
2.
3.     public static void Main(string[] args) {
4.         Vozilo v = new Vozilo(8);
5.         v.vozi();
6.         v.brojVrata = 2;
7.         Vozilo bmw = new Auto(2, 4);
8.         bmw.vozi();
9.         Auto audi = new Auto(4);
10.        Auto.vozi();
11.    }
12. }
13. class Vozilo {
14.     public int brojTockova;
15.     public Vozilo(int t) {
16.         brojTockova = t;
17.     }
18.     public void vozi() {
19.         Console.WriteLine("Vožnja vozila");
20.     }
21. }
22. class Auto : Vozilo {
23.     public int brojVrata;
24.     public Auto(int v, int t) : base(t)
25.     {
26.         brojVrata = v;
27.     }
28.     public void vozi() {
29.         Console.WriteLine("Vožnja auta");
30.     }
31. }
```

1. Грешке су у редовима 7, 10 и 16.
2. Грешке су у редовима 7, 9 и 22.
3. Грешке су у редовима 6, 7 и 10.
4. Грешке су у редовима 6, 9 и 10.
5. Грешке су у редовима 6, 9 и 24.
6. Грешке су у редовима 9, 10 и 24.

Грешке:

6. линија програмског кода: `v.brojVrata = 2;`

Приступа се атрибуту **brojVrata** који је у класи **Auto**. Класа **Auto** је изведена класа из класе **Vozilo**. Дакле, класа **Vozilo** не може да приступи жељеном атрибуту.

9. линија програмског кода: `Auto audi = new Auto(4);`

Позива се конструктор са једним параметром а класа **Auto** има само конструктор са два параметра.

10. линија програмског кода: `Auto.vozi();`

Позива се статичка метода **vozi()** а метода **vozi()** је не статичка.