

Типови конструктора

Типови конструктора су:

- Конструктор без параметара
- Конструктор са параметрима
- Дифолтни конструктор
- Копирајући конструктор
- Приватни конструктор
- Статички конструктор

Конструктор са параметрима

Пример:

```
using System;
namespace Proba
{
    class Automobil
    {
        string naziv;
        int cenaDolari;
        Automobil(string naz, int cen)
        {
            naziv = naz;
            cenaDolari = cen;
        }
        static void Main()
        {
            Automobil auto1 = new Automobil("Porse", 500000);
            Console.WriteLine("Naziv: " + auto1.naziv);
            Console.WriteLine("Cena: " + auto1.cenaDolari);
        }
    }
}
```

Даје:

Naziv: Porse

Cena: 500000

У примеру је креиран конструктор Auto() са два параметра naziv и cenaDolari.

У линији, `Automobil auto1 = new Automobil("Porse", 500000);` додају се две вредности конструктору.

Ове вредности се називају аргументи а они се као параметри додају конструктору; аргумента и параметара мора бити исти број и да одговарају по типу података.

Копирајући конструктор

Можемо користити копирајући конструктор за креирање објекта копирањем података из другог објекта.

Пример:

```
using System;
namespace Proba
{
    class Automobil
    {
        string naziv;
        Automobil(string naz)
        {
            naziv = naz;
        }
        Automobil(Automobil a1)//kopirajuci konstruktor
        {
            naziv = a1.naziv;
        }
        static void Main()
        {
            Automobil auto1 = new Automobil("Porse");
            Console.WriteLine("Naziv auto1: " + auto1.naziv);
            Automobil auto2 = new Automobil(auto1);
            Console.WriteLine("Naziv auto2: " + auto2.naziv);
        }
    }
}
```

Дaje:

Naziv auto1: Porse

Naziv auto2: Porse

Копирајући конструктор добија објекат класе Automobil као параметар.

Зато, када се креира објекат auto2, придодаје се објекат auto1 као аргумент у копирајући конструктор: `Automobil auto2 = new Automobil(auto1);`

Унутар копирајућег конструктора, додељена је вредност naziv објекта auto1 у промењиву naziv објекта auto2 и зато оба објекта имају исту вредност промењиве naziv.

Приватни конструктор

Приватни конструктор се креира коришћењем специфакатора приступа (access specifier) private.

Када се конструктор декларише као приватан, не могу се креирати објекти те класе унутар других класа.

Пример:

```
using System;
namespace Proba
{
    class Automobil
```

```

{
    private Automobil()
    {
        Console.WriteLine("Privatni konstruktor");
    }
}
class VoznjaAutomobilom
{
    static void Main()
    {
        Automobil auto1 = new Automobil();
    }
}

```

Дaje: 'Automobil.Automobil()' is inaccessible due to its protection level

Пошто чланови класе који су декларисани као приватни нису доступни изван те класе, када се покуша креирање објекта класе, добија се дата грешка.

То значи да ако је потребно креирати објекте неке класе, сва поља и методе те класе треба да су статична, да би им се могло приступити коришћењем имена класе.

Статички конструктор

Да би се креирао статички конструктор, користи се службена реч static.

Пример:

```

using System;
namespace Proba
{
    class Automobil
    {
        static Automobil()
        {
            Console.WriteLine("Staticki konstruktor");
        }
        Automobil()
        {
            Console.WriteLine("Difoltni konstruktor");
        }
        static void Main()
        {
            Automobil auto1 = new Automobil();
            Automobil auto2 = new Automobil();
        }
    }
}

```

Дaje:

Staticki konstruktor

Difoltni konstruktor

Difoltni konstruktor

Статички конструктор се не може директно позвати, најчешће се реализује позивом дифолтног конструктора.

У случају да не постоји посебно креиран дифолтни конструктор, на екрану би се појавило само: Staticki konstruktor и то само једном.

Статички конструктор се позива само једном током извршења програма, па се приликом више позива дифолтног конструктора, статички конструктор позива само први пут.

Ако у класи постоји само статички конструктор, он не може имати параметре нити модификаторе приступа.

Задаци за самосталан рад

1. У класи Zivotinja креирати конструктор са параметрима boja, visina и vrsta. Проверти функционисање овог конструктора у случају објеката komaras, slon и covek.
2. Искористити конструктор из претходног примера да би се преко копирајућег конструктора креирао објекат krokodil.
3. У код првог задатака убацити статички конструктор и проверити његово функционисање.