

Увод у конструкторе

Када се употреби службена реч `new`, приликом креирања објекта, `runtime` тражи конструкцију тог објекта коришћењем дефиниције класе.

`Runtime` мора одвојити део меморије од оперативног система, попунити је са пољима дефинисаним у класи а затим позвати конструктор да би се извршила тражена иницијализација.

Конструктор је посебна метода која се аутоматски извршава када се креира инстанца класе.

Има исто име као и класа и може имати параметре али не може вратити вредност.

Свака класа мора имати конструктор, ако програмер не напише конструктор, компајлер аутоматски генерише дифолтни конструктор (иако такав конструктор не ради ништа и нема никакав параметар).

Програмер лако креира дифолтни конструктор додавањем јавне методе која не враћа вредност и даје му исто име као што је и класа.

Пример: класа `Krug` са дифолтним конструктором који иницијализује поље полупречника на 0:

```
class Krug
{
    private int r;
    public Krug()    //difoltni konstruktor
    {
        r = 0;
    }
    public double RacunanjePovrsine()
    {
        return Math.PI * r * r;
    }
}
```

У примеру, конструктор је означен као јаван (`public`).

Да није никакав модификатор коришћен, подразумевало би се да је приватан (`private`).

Ако је конструктор приватан, не може се користити изван класе, чиме се спречава креирање објекта из класе `Krug` од стране метода које нису део класе `Krug`.

Како је у примеру конструктор јаван, може се користити класа `Krug` и метода `RacunanjePovrsine()`:

```
static void Main()
{
    Krug k = new Krug();
    double povrs = k.RacunanjePovrsine();
}
```

Преклапање конструктора

У претходном примеру се појављује проблем што коришћењем дифолтног конструктора поље `r` ће имати увек вредност постављену на 0.

Пошто је поље `r` приватно, не постоји лак начин мењања његове вредности после иницијализације.

Пошто је конструктор метода и као и све методе може бити преклопљено (`overloaded`).

Као што постоји неколико верзија Console.WriteLine() методе, где свака од њих ради са различитим параметрима, исто важи и за верзије конструктора.

Могуће је додати другачији конструктор у исту класу са параметром који другачије користи поље r:

```
class Krug
{
    private int r;
    public Krug()    //difoltni konstruktor
    {
        r = 0;
    }
    public Krug(int inicijalniPoluprecnik)    //preklopljeni konstruktor
    {
        r = inicijalniPoluprecnik;
    }
    public double RacunanjePovrsine()
    {
        return Math.PI * r * r;
    }
}
```

Нови конструктор се може користити када се креира нови Krug објекат:

```
Krug k = new Krug(50);
```

Када се креира апликација, компајлер сам проналази који конструктор да користи у зависности од параметара који се специфицирају са new оператором.

У примеру, вредност у загради је целобројна, па компајлер генерише код који позива конструктор који у дефиницији користи int параметар.

Такође, ако програмер напише свој конструктор класе, у том случају, компајлер неће генерисати дифолтни конструктор.

То значи, да ако је потребно имати неколико преклапајућих конструктора и дифолтни конструктор, програмер сам мора написати дифолтни конструктор.

Конструктор без параметара

Конструктор може бити без параметара (parameterless constructor)

```
using System;
namespace Proba
{
    class Program
    {
        class Auto
        {
            Auto()
            {
                Console.WriteLine("Konstruktor za Auto");
            }
        }
    }
}
```

```

    }
    static void Main()
    {
        new Auto();
    }
}

```

Даје: Konstruktor za Auto

У примеру је креиран конструктор без параметара Auto().

Позив конструктору је изведен са кључном речи new над именом конструктора.

Задаци за самосталан рад

1. Доказати да C# компајлер креира дифолтни конструктор приликом дефинисања класе.
2. Креирати класу Taska са конструктором без параметара.
3. Креирати класу Obim која рачуна обим круга ако је иницијална вредност за полупречник 10.
4. Модификовати претходни пример и креирати дифолтни конструктор (полупречник једнак 1) и преклопљени конструктор за унети целобројни полипречник круга.