
Singleton Pattern Kod Satırı Açıklaması

```
-----SingletonMainProgram.cs Sınıfı -----  
  
using System; ❶  
  
namespace SingletonPattern ❷  
{  
    class SingletonMainProgram ❸  
    {  
        public static void Main(string[] args)  
        {  
            Singleton singleton1 = Singleton.CreateInstance() ❹  
            Singleton singleton2 = Singleton.CreateInstance(); ❺  
  
            if(singleton1 == singleton2) ❻  
            {  
                Console.WriteLine("Objects are using the same  
instance"); ❼  
            }  
            else  
            {  
                Console.WriteLine("Objects are not using the same  
instance"); ❽  
            }  
        }  
    }  
}  
  
-----Singleton.cs Sınıfı-----  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace SingletonPattern  
{  
    public sealed class Singleton  
    {  
        private static Singleton instance = null; ❾
```

```
private Singleton() ❩
{
}

public static Singleton CreateInstance()
{
    if (instance == null) ❪
    {
        instance = new Singleton();❫
    }

    return instance; ❬
}
}
```

- ❶ Kütüphane Ekleme
- ❷ Namespace İsmi
- ❸ SingletonMainProgram adlı sınıfın kod blogunun başlangıcı
- ❹ Singleton tipindeki singleton1 nesnesi için Singleton sınıfındaki CreateInstance methoduna başvuru.
- ❺ Singleton tipindeki singleton2 nesnesi için Singleton sınıfındaki CreateInstance methoduna başvuru.
- ❻ CreateInstance methodunun gelen instance'a göre singleton örüntüsünün doğru çalışıp çalışmadığının ekrana basılması.
- ❼ Tek bir instance döndürülerek singleton örüntüsünün temel ilkesinin sağlandığını gösterir.
- ❽ Singleton örüntüsünün doğru çalışmadığını gösterir.
- ❾ Singleton tipindeki statik nesnesin tanımlanması ve null değerine eşitlenmesi
- ❿ Private Constructor Tanımlanması
- ❫ instance değerinin null olup olmadığı kontrolü
- ❬ instance oluşturma (Daha önce oluşturulmamış, ilk defa oluşturuluyor anlamı taşır.)
- ❭ Oluşturulan Instance return edilir.