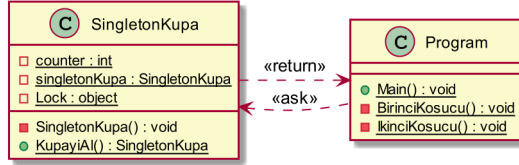


Thread Safe Singleton ALPER ÇELİK 160202023

1. Thread Safe Singleton ALPER ÇELİK 160202023



```
class Program
{
    static void Main(string[] args)
    {
        Parallel.Invoke(() => Birincikosucu(), () =>
        Ikincikosucu()); ❶
    }

    private static void Birincikosucu()
    {
        SingletonKupa birincikosucu = SingletonKupa.KupayiAl(); ❷
        Console.WriteLine("Birinci kosucu");
    }

    private static void Ikincikosucu()
    {
        SingletonKupa ikincikosucu = SingletonKupa.KupayiAl(); ❸
        Console.WriteLine("ikinci kosucu");
        Console.ReadKey();
    }
}
```

- ❶ Birincikosucu ve Ikincikosucu fonksiyonlarının eşzamanlı çalıştırılması
- ❷ SingletonKupa tipinde birincikosucu nesnesi oluşturulup KupayiAl fonksiyonu ile değer ataması
- ❸ SingletonKupa tipinde ikincikosucu nesnesi oluşturulup KupayiAl fonksiyonu ile değer ataması

```
public sealed class SingletonKupa
```

```
{
    private static int counter = 0; ❶
    private SingletonKupa() ❷
    {
        Console.WriteLine("{0} kere constructor methodu calisti",
counter);
    }
    private static SingletonKupa singletonKupa = null; ❸
    private static readonly object Lock = new object(); ❹

    public static SingletonKupa KupayiAl
    {
        get
        {
            lock (Lock) ❺
            {
                if (singletonKupa == null) ❻
                {
                    counter++;
                    singletonKupa = new SingletonKupa(); ❼
                }
            }
            return singletonKupa; ❽
        }
    }
}
```

- ❶ Constructor methodunda yazdırmak için counter tanımlandı
- ❷ Private Constructor
- ❸ Private SingletonKupa tipinde singletonKupa nesnesi
- ❹ Thread Safe işlemi için obje oluşturuldu
- ❺ Thread Safe işlemini sağlamak için lock ifadesi
- ❻ Nesnenin daha önce oluşturulup oluşturulmadığını kontrol eder
- ❼ singletonKupa adlı değişkene değer atar