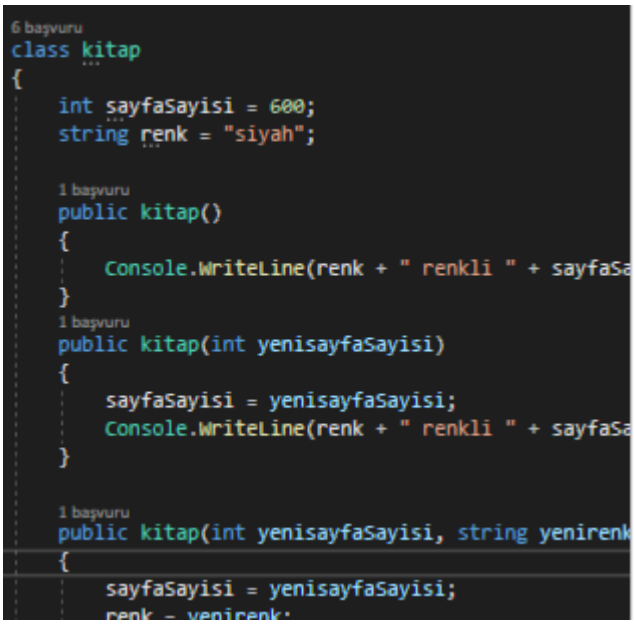
1.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Sınav
{
    class Program
    {
        static void Main(string[] args)
        {
            new kitap();
            new kitap(255);
            new kitap(123, "mavi");
            Console.ReadLine();
        }
    }

    class kitap
    {
        int sayfaSayisi = 600;
        string renk = "siyah";

        public kitap()
        {
            Console.WriteLine(renk + " renkli " + sayfaSayisi + " sayfali kitap.");
        }
        public kitap(int yenisayfaSayisi)
        {
            sayfaSayisi = yenisayfaSayisi;
            Console.WriteLine(renk + " renkli " + sayfaSayisi + "sayfali kitap.");
        }

        public kitap(int yenisayfaSayisi, string yenirenk)
        {
            sayfaSayisi = yenisayfaSayisi;
            renk = yenirenk;
            Console.WriteLine(renk + " renkli " + sayfaSayisi + "sayfali kitap.");
        }

    }
}
```

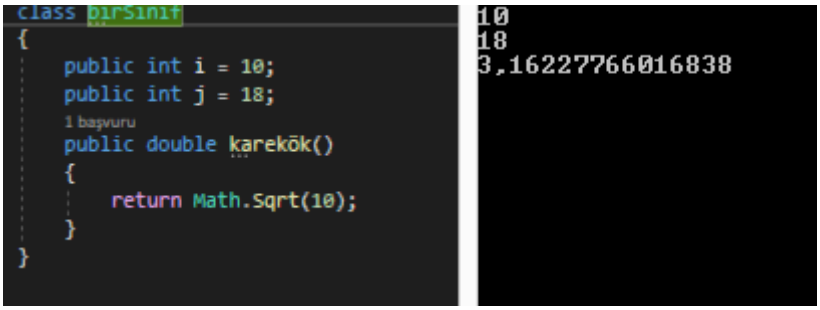


```
6 başvuru
class kitap
{
    int sayfaSayisi = 600;
    string renk = "siyah";

    1 başvuru
    public kitap()
    {
        Console.WriteLine(renk + " renkli " + sayfaSa
    }
    1 başvuru
    public kitap(int yenisayfaSayisi)
    {
        sayfaSayisi = yenisayfaSayisi;
        Console.WriteLine(renk + " renkli " + sayfaSa
    }

    1 başvuru
    public kitap(int yenisayfaSayisi, string yenirenk
    {
        sayfaSayisi = yenisayfaSayisi;
        renk = yenirenk;
```

```
siyah renkli 600 sayfali kitap.
siyah renkli 255 sayfali kitap.
mavi renkli 123 sayfali kitap.
```

2.

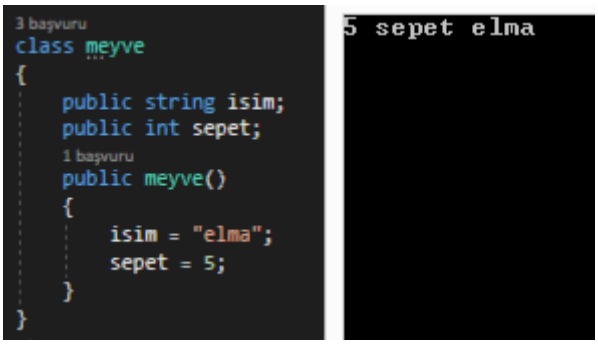
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp6
{
    class Program
    {
        static void Main(string[] args)
        {
            birSınıf nesne1 = new birSınıf();
            birSınıf nesne2 = new birSınıf();
            birSınıf nesne3 = new birSınıf();
            Console.WriteLine(nesne1.i);
            Console.WriteLine(nesne2.j);
            Console.WriteLine(nesne3.karekök());
            Console.ReadLine();
        }
    }
    class birSınıf
    {
        public int i = 10;
        public int j = 18;
        public double karekök()
        {
            return Math.Sqrt(10);
        }
    }
}
```



Verilen kod ilk verildiği haliyle çalışmıyordu ve hata veriyordu. Bunun sebebi birSınıf classının içinde olan değişkenler ve metotlar public değildi o yüzden yukarıda yarattığım nesneler bu özellikleri çağıramıyordu. Başlarına public yazdıktan sonra nesneler bu özellikleri çağırmaya başladı ve problem çözüldü.

3.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp7
{
    public class meyve
    {
        public string isim;
        public int sepet;
        public meyve()
        {
            isim = "elma";
            sepet = 5;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            meyve nesne = new meyve();
            Console.WriteLine(nesne.sepet + " sepet " + nesne.isim);
            Console.ReadLine();
        }
    }
}
```



Program yazıldığında nesne.sepet ve nesne.isim kısımları hata veriyordu çünkü nesne objesinin bu değişkenlerinin başında public yazmadığı için bu değişkenlere erişim yolu yoktu bu yüzden de hata veriyordu. Meyve classının kurucusunun da başında public yazmıyordu o yüzden nesne yaratma kısmında da hata veriyordu. Gerekli yerlere public yazarak bu hataları çözmüş oldum ve program istenildiği gibi çıktı verdi.

4. Sınıf, Nesne, Statik Öğe, Kurucu ve Yokedici kavramlarını en az bir sayfa olucak şekilde açıklayınız.

Sınıf : Sınıflar bir nesnenin olması gereken özelliklerini, davranışlarını tanımlayan bir şablondur. Sınıf içerisinde sınıf nesneleri oluşturmak için temel özellikleri, bu nesnelerin davranışlarını(metotlar), ve nesnelerin özelliklerini ve bunları işleyecek kodları içerir. Bu yüzden sınıflar nesnelerin nasıl inşa edileceğini, davranacağını tanımlayan birer kılavuzdurlar. Sınıflar soyuttur, sınıfların nesneleri oluşturulana kadar fiziksel olarak bellekte yer almazlar. Sınıflar değişkenler, metotlar, yapılandırıcılar ve yokediciler gibi özellikleri içerir.

```
class meyve
    {
        public string isim;
        public int sepet;
        public meyve()
        {
            isim = "elma";
            sepet = 5;
        }
    }
```

Nesne : Nesneler sınıfların bir örneğidir. Nesneler sınıflardaki kodu anlamlı bir şekilde modellemeye yarar. Sınıfları işe yarar hale getirmeye yarar. Bir nesne oluşturduğumuzda o sınıfta olan her şeyin kullanılabilir bir kopyasını oluşturmuş oluruz.

```
meyve nesne = new meyve();
new kitap();
new kitap(255);
new kitap(123, "mavi");
```

Nesneleri bu şekillerde oluşturabiliriz. Sınıf içerisine yazdığımız yapılandırıcıların kabul ettiği parametre sayılarına göre farklı şekillerdde nesneler oluşturabiliriz. Sınıfların içerisinde yer alan özelliklerin başında public yazıyorsa bunlara nesne üzerinden ulaşabiliriz, bazen önünde private yazan değişkenler, metotlarla da karşılaşabiliriz bu durumda daha farklı bir yol izlemeliyiz. Bir nesne oluşturduğumuz zaman fiziksel olarak bir yapı oluşturmuş oluyoruz bu da ramde yer almış oluyor.

Kurucular : Bir sınıftan bir nesne oluşturmak için kullandığımız yapılardır.

```
public kitap()
 {
     Console.WriteLine(renk + " renkli " + sayfaSayisi + " sayfali kitap.");
 }
 public kitap(int yenisayfaSayisi)
 {
     sayfaSayisi = yenisayfaSayisi;
     Console.WriteLine(renk + " renkli " + sayfaSayisi + "sayfali kitap.");
 }

 public kitap(int yenisayfaSayisi, string yenirenk)
 {
     sayfaSayisi = yenisayfaSayisi;
     renk = yenirenk;
     Console.WriteLine(renk + " renkli " + sayfaSayisi + "sayfali kitap.");
 }
```

Bu soruda yaptığımız gibi birden fazla çeşitte olabilirler. Kurucular nesnelerin ilk oluşturulması sırasında ilk değerlerini vermemize yarar. Bu sayede daha güvenli ve daha düzenli bir yapıda nesnelerimizi oluşturmuş oluruz. Kurucular ait oldukları sınıf ile aynı isme sahiptir. Yapı olarak da metotlara benzemektedir. Dönüş tipleri yoktur. Farklı parametreler kabul ettikleri sürece aynı isimle birden fazla sayıda Kurucu yazılabilir.

Yokediciler : Bir programda nesneler oluşturabiliriz, onları kullanabiliriz. Fakat bazı nesneler bir süre sonra boşa çıkabilir. Bu boşa çıkan nesneleri C#da çöp toplayıcılar adını verdiğimiz mekanizma belli aralıklarla ortadan kaldırır. Bu otomatik gerçekleşen bir mekanizmadır ve biz buna müdahale edemeyiz. Fakat bazen belli nesneleri biz kaldırmaz isteriz. Çöp toplayıcısının ne zaman ne aralıkla çalışacağını bilmediğimiz için bu konuda bizler yokediciler ile müdahale ederiz. Yok ediciler kurucuların yazılmasına benzer şekilde, sınıf ismini alarak metot görünümlü yapılar olmasına rağmen ayırt edici özellikleri :

```
~sınıf-adı() {
}
```

Şeklinde olmalarıdır. Geri dönüş değerleri yoktur.Bu sayede yokedicileri kullanarak istediğimiz zaman istemediğimiz nesneleri temizleyebiliriz.

Statik Öğe: Normalde bir sınıfın özelliklerine o sınıftan oluşturulan nesneler ile erişebilmekteyiz.

```
Console.WriteLine(nesne.sepet + " sepet " + nesne.isim);
```

Ancak bazen nesnelerden bağımsız olarak bazı metotlara ya da değişkenlere erişme ihtiyacımız olursa onlara statik bir metot

meyve.sepet, meyve.isim (meyve class ismi) şeklinde çağrılabilir.

Bu yüzden static değişkenlere global değişkenler diyebiliriz. Static değişkenler programın her yerinde geçerlidir , static değişkenleri tüm üyeler kullanabilir. Static bir metot yalnızca kendi sınıfı tarafından tanımlanan static değişkenlere erişebilir. Ek olarak static kurucular ve static sınıflar oluşturulabilir. Özet olarak içinde bulunduğu sınıftan nesne oluşturulmadan ve hiçbir nesneye referans olmadan kullanabilen üyelere static denir. Örnek olarak main metotu static bir metota örnektir.