

# Nesne Yönelimli Programlama

## 6.Hafta

Konu: Sinema Sistemi, Örnek  
Algoritma Tasarımı

---

Dr. Öğr. Üyesi Güncel SARIMAN  
E-posta: [guncelsariman@mu.edu.tr](mailto:guncelsariman@mu.edu.tr)



# this anahtar sözcüğü

---

this anahtar sözcüğü ilgili nesnenin referansını belirtmektedir.

```
class Ogrenci
{
    private string AdSoyad;
    private int Numara;
    private int Sinif;

    public Ogrenci(string AdSoyad, int Numara, int Sinif)
    {
        this.AdSoyad = AdSoyad;
        this.Numara = Numara;
        this.Sinif = Sinif;
    }
}
```

this anahtar sözcüğü ile nitelenen değişkenler sınıfın üye elemanlarıdır. Yukarıdaki veri modelinde yapıcı metoda parametre olarak gönderilen değerler, sınıfın üye elemanlarına aktarılacaktır.



# Bilet Satışı

Salon	<input type="text"/>	OLUŞTUR	
Koltuk Sayısı	<input type="text"/>		
SATIŞ-IPTAL		RAPOR	
<input type="checkbox"/> İndirimli			
BİLET SAT	BİLET İPTAL	BAKİYE	BOŞ KOLTUK
label3			



# Bilet Satışı

---

OLUŞTUR Butonunu kullanarak oluşturulacak Sinema sınıfından yeni bir nesne oluşturulacak. Sınıf yapısında BiletSat, BiletIptal, BosKoltukOgren ve BakiyeOgren metotlarını oluşturarak gerekli işlemlerin yapılmasını sağlayacak ve bu işlemlerle ilgili label3 kontrolünde kullanıcı bilgilendirmesi yapılacaktır.

Öncelikle **Sinema** isimli **Class**' ı oluşturulacak.

```
class Sinema
```

```
{
```

```
    int toplamKoltukSayisi;
```

```
    int bosKoltukSayisi;
```

```
    double bakiye;
```

```
    string salonNo;
```

```
    const double TAM = 15.0;
```

```
    const double INDIRIMLI =
```

```
    10.0;
```

```
}
```



# Bilet Satışı

---

Örnek için gerekli özellikler oluşturuldu. Öncelikle kurucu metot oluşturmalıdır.

```
public Sinema(string salonn,int koltuksayi) //Kurucu Metot
```

```
{
```

```
    toplamKoltukSayisi = koltuksayi;
```

```
    salonNo = salonn;
```

```
    bosKoltukSayisi = toplamKoltukSayisi;
```

```
    bakiye = 0;
```

```
}
```



# Bilet Satışı

---

BiletSat isimli metod,

```
public void BiletSat(bool indirimli) //Bilet Satış
{
    bosKoltukSayisi--;
    if (indirimli)
        bakiye += INDIRIMLI;
    else
        bakiye += TAM;
}
```

Bilet satışı gerçekleştiğinde boş koltuk sayısının azaltılması sağlanıyor. Ayrıca indirimli veya tam bilet olması durumuna göre bakiyenin arttırılması sağlanıyor.



# Bilet Satışı

---

**Biletİptal** isimli metod,

Gelen parametre ile biletin indirimli ya da tam bilet olduğunun kontrolünü yapacağız. Para iadesi olacağı için bakiye azaltılacak ve boş koltuk sayısının arttırılması sağlanacaktır.

```
public void Biletİptal(bool indirimli) //Bilet İade
{
    bosKoltukSayisi++;
    if (indirimli)
        bakiye -= INDIRIMLI;
    else
        bakiye -= TAM;
}
```



# Bilet Satışı

---

**Boş koltuk sayısını öğrenebileceğimiz BosKoltukOgren isimli metod,**

```
public int BosKoltukOgren() //Boş koltuk Sayısı
```

```
{
```

```
    return bosKoltukSayisi;
```

```
}
```



# Bilet Satışı

---

BakiyeOgren isimli metod. Bu metotda yukarıdaki BosKoltukOgren metodu gibi çalışacak ve geriye bakiye bilgisini döndürecektir.

```
public double BakiyeOgren() //Bakiye Öğren
{
    return bakiye;
}
```



# Bilet Satışı

---

Sinema salon;

```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
    try
```

```
    {
```

```
        salon = new Sinema(textBox1.Text, Convert.ToInt32(textBox2.Text));
```

```
        label3.Text = "Salon Oluşturuldu. Koltuk sayısı : " + salon.BosKoltukOgren();
```

```
    }
```

```
    catch
```

```
    {
```

```
        label3.Text = "Salon Oluşturulamadı. Bilgileri kontrol edin.";
```

```
    }
```

```
}
```



# Bilet Satışı

---

```
private void button2_Click(object sender, EventArgs e)
{
    salon.BiletSat(checkBox1.Checked);
    label3.Text = "Bilet satıldı. Kalan koltuk sayısı : " + salon.BosKoltukOgren();
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    salon.BiletIptal(checkBox1.Checked);
    label3.Text = "Bilet iptal edildi. Kalan koltuk sayısı : " +
salon.BosKoltukOgren();
}
```



# Bilet Satışı

---

```
private void button5_Click(object sender, EventArgs e)
{
    label3.Text = "Şu anki bakiye " + salon.BakiyeOgren() + " TL";
}

private void button4_Click(object sender, EventArgs e)
{
    label3.Text = "Boş koltuk sayısı " + salon.BosKoltukOgren();
}
```



# Örnek Çalışma: Laboratuvar Envanter Takip Sistemi

---

## İş Tanımı

- 1-Üniversite bünyesindeki tüm laboratuvarlardaki malzemelerin sisteme kaydedilmesi, güncellenmesi, silinmesi ve okul bazında rapor alınabilmesini sağlayan bir sistem geliştirilecektir.
- 2-Sistem MSKÜ'deki fakülte, yüksekokul bölümlerinin lablarını sisteme kaydetmektedir.
- 3-Kaydedilen labların içlerine türlerine göre malzemeler kaydedilmektedir.
- 4-Lab türleri bölümlere göre değişmektedir. Bilgisayar, Kimya, Biyoloji, Elektrik vb.
- 5-Her bölümün malzemeleri farkı olsa da girilmesi gereken bilgiler ortaktır.
- 6-Malzeme Adı, Markası, Ne zaman Alındığı, Kaç adet olduğu, Tipi(Demirbaş veya Sarf Malzemesi vb.)
- 7-Sistem de okul bölüm ve sorumlu kullanıcı sisteme tanıtılabilecektir.
- 8-Kullanıcılar sistemi kullanırken malzemeler hakkında tüm bilgilere erişebilmektedir.



# Örnek Çalışma: Laboratuvar Envanter Takip Sistemi

---

## Olması Gerekenler

1-Okul

2-Bölüm

3-Program

4-Laboratuvarlar

5-Lab Türleri

6-Malzemeler

7-Kullanıcılar



# Class Kullanarak Dik Üçgende Hipotenüs Hesaplama

Örneğimizi **Property** kullanarak oluşturacağız. Girilen kenarın "0" ve daha küçük olması durumunda **Class** içerisinde "1" değerine set edilmesini sağlayacağız.

```
class Ucgen
{
    double kenar1;

    public double Kenar1
    {
        get { return kenar1; }
        set
        {
            if (value <= 0) // Kenar 0 ve daha küçükse 1 yapıyoruz.
            {
                kenar1 = 1;
            }
            else
                kenar1 = value;
        }
    }
}
```



# Class Kullanarak Dik Üçgende Hipotenüs Hesaplama

---

```
double kenar2;
```

```
public double Kenar2
{
    get { return kenar2; }
    set
    {
        if(value<=0) // Kenar 0 ve daha küçükse 1 yapıyoruz.
        { kenar2 = 1; }
        else
            kenar2 = value;
    }
}
```



# Class Kullanarak Dik Üçgende Hipotenüs Hesaplama

---

```
public double Alan()  
{  
    double ucgenAlan=0;  
    ucgenAlan=(kenar1*kenar2)/2;  
    return ucgenAlan;  
}  
public double Hipotenus()  
{  
    double ucgenHipotenus = 0;  
    ucgenHipotenus = Math.Sqrt((Math.Pow(kenar1, 2) + Math.Pow(kenar2, 2)));  
    return ucgenHipotenus;  
}
```



# Class Kullanarak Dik Üçgende Hipotenüs Hesaplama

---

```
Ucgen dikUcgen = new Ucgen();  
dikUcgen.Kenar1 = Convert.ToDouble(textBox1.Text);  
dikUcgen.Kenar2 = Convert.ToDouble(textBox2.Text);  
label3.Text = "Alan = " + dikUcgen.Alan();  
label4.Text = "Hipotenüs = " + dikUcgen.Hipotenus();
```