BİL 201
NESNE YÖNELİMLİ PROGRAMLAMA (NYP)

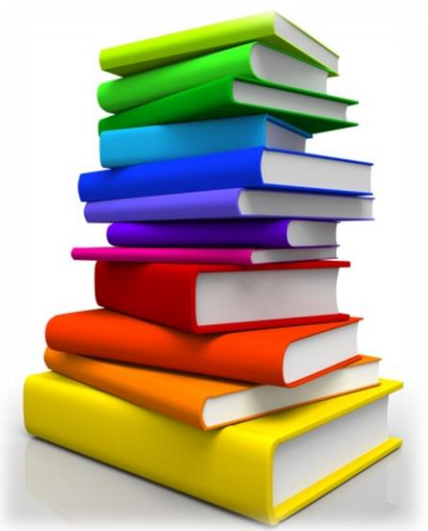
DERS #6

Öğretim Üyesi: Doç. Dr. Deniz Kılınç

BAKIRÇAY ÜNİVERSİTESİ

BÖLÜM 6 – Kalıtım, Miras (Inheritance)

- Bu bölümde aşağıdaki konular anlatılacaktır
  - Kalıtım Kavramı,
  - Kalıtım Terimleri,
  - Sınıfların Genişletilmesi,
  - protected Erişim Belirleyicisi,
  - Temel Sınıfların Metotlarını Ezme



Kalıtım, Miras (Inheritance) Kavramı

- Sınıfları anlamak gerçek hayatta nesneleri **düzenlemenize** yardımcı olur.
- Kalıtımı anlamak onları daha net bir şekilde **organize** etmenizi sağlar.
- Eğer Braford'u hiç duymadıysanız zihninizde canlandırmanız **mümkün değildir**.

Kalıtım, Miras (Inheritance) Kavramı (devam…)

? Braford ?

Hayvan

Memeli

Kalıtım, Miras (Inheritance) Kavramı (devam…)

? Braford ?

İnek

Kalıtım, Miras (Inheritance) Kavramı (devam…)

- Bu fikir onun memeli olduğunu öğrenince daha da büyür ve onun bir inek olduğunu öğrenince bu fikir zihninizde net bir hal alır.
- Braford'un bir inek olduğunu öğrendiğinizde, onun birçok inekte ortak olan özelliklere sahip olduğunu anlarsınız.
- Bir Braford'u ayırt edebilmek için sadece ona ilişkin rengi, büyüklüğü, işaretleri gibi ufak detayları öğrenmeniz gerekir.

Kalıtım, Miras (Inheritance) Kavramı (devam…)

- Halbuki Braford'un özelliklerinin çoğu, şu sınıfların **hiyerarşik yapısından** gelir:

  Hayvan → Memeli → İnek

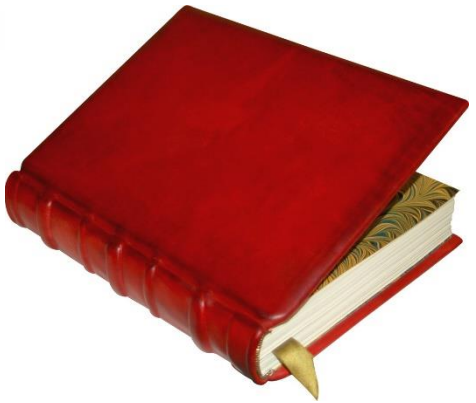
- Tüm **"object-oriented"** programlama dilleri kalıtım özelliğinden *aynı sebepler* için *faydalanmaktadır*:
  - Programlarda kullanılan nesneleri düzenlemek,
  - Kalıtımla bildiklerinizi kullanarak yeni nesneleri yaratmak
  - Kod parçalarının **reusability'sini** (tekrar kullanılabilirlik) arttırmak.

Kalıtım, Miras (Inheritance) Kavramı (devam…)

- Kalıtım, sizin genel bir kategori hakkındaki bildiklerinizi daha **spesifik bir kategoriye** uygulamanıza olanak sağlayan prensiptir.
- Kalıtım terimi kullanıldığında, genetik kalıtımı düşünebilirsiniz.
  - **Kan grubu** veya **göz rengi** kalıtılmış genlerin ürünüdür.
  - **Yürüyüşünüzün** babaannenizle aynı olması, ki bu yürüyüş size **babanızdan kalıtılmış** denebilir.

Kalıtım, Miras (Inheritance) Kavramı (devam…)

- Farklı tipte *ürünler* satan Ürün Satış uygulaması geliştirmek istediğimizi varsayalım.



Bu ürünlerin **sınıflarını** **oluşturabilir miyiz**?

Kalıtım, Miras (Inheritance) Kavramı (devam…)

- Telefon ve Kitap ürünlerinin **özellikleri** nelerdir?

Telefon
+No: int +Adi: string +Marka: string +Model: string +Aciklama: string +Fiyat: decimal

Kitap
+No: int +ISBN: int +Adi: string +Yazar: string +Aciklama: string +Fiyat: decimal

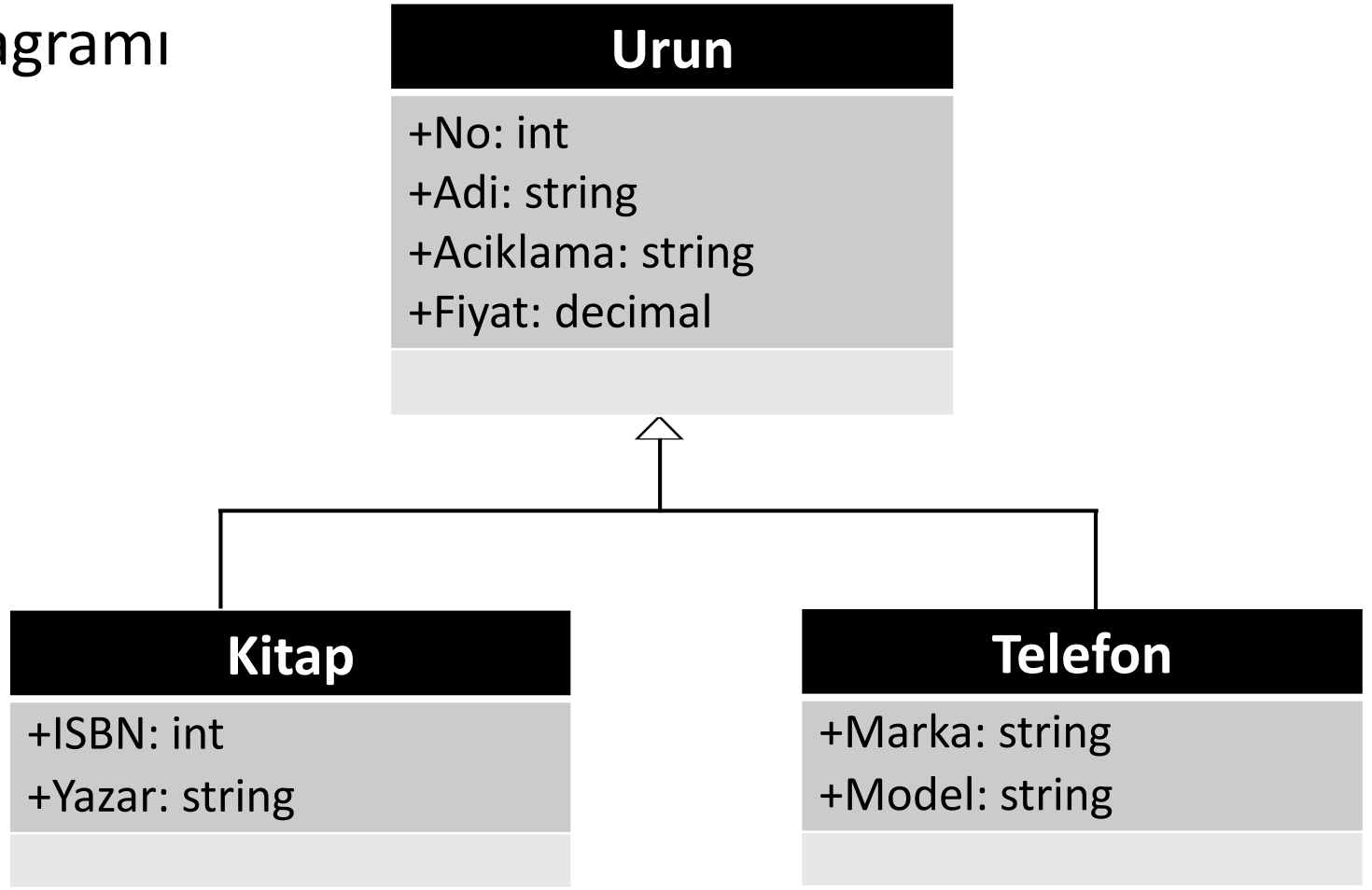
Kalıtım, Miras (Inheritance) Kavramı (devam…)

- Telefon ve Kitap ürünlerinin **ortak özellikleri** nelerdir?

UrunOrtakOzellikler
+No: int +Adi: string +Aciklama: string +Fiyat: decimal

Örnek1: İlk Kalıtım Örneği

- UML Sınıf Diyagramı



Kalıtımda Kullanılan Terimler

- Kalıtım için temel alınan sınıflara, Urun sınıfı gibi, temel sınıflar (ing.: **base classes**) denir.
- Temel sınıftan kalıtılarak oluşturulmuş sınıfa, Kitap gibi,
  - ✓kalıtılmış sınıf (ing.: **derived class**) veya
  - ✓genişletilmiş sınıf (ing.: **extended class**) denir.
  - ✓miras alınmış sınıf (ing.: **inherited class**) denir.

Kalıtımda Kullanılan Terimler (devam…)

- Ayrıca superclass ve subclass terimleri de temel sınıf ve kalıtılmış sınıflar için kullanılmaktadır.
  - ✓Kitap sınıfı Urun superclass'ının subclass'ıdır.
- Buna benzer bir kullanım ayrıca ana (ing.: **parent**) ve yavru (ing.: **child**) sınıf kavramları da kullanılmaktadır.
  - ✓Kitap sınıfı Urun ana sınıfının yavru sınıfıdır.

Sınıfların Genişletilmesi

- Başka ***bir sınıftan kalıtım ile yeni genişletilmiş bir sınıf yaratmak için*** (O sınıfın bir yavru sınıfını oluşturmak için) sınıf başlığında yavru sınıfın adı, iki nokta üst üste, ana sınıf adı yazılarak tanımlanır.

```
class [yavruSinifAdi] : [anaSinifAdi]
{
}
class Kitap : Urun
{
}
```

- Kitap bir Üründür.
- Book **is a** Product.

is a relation

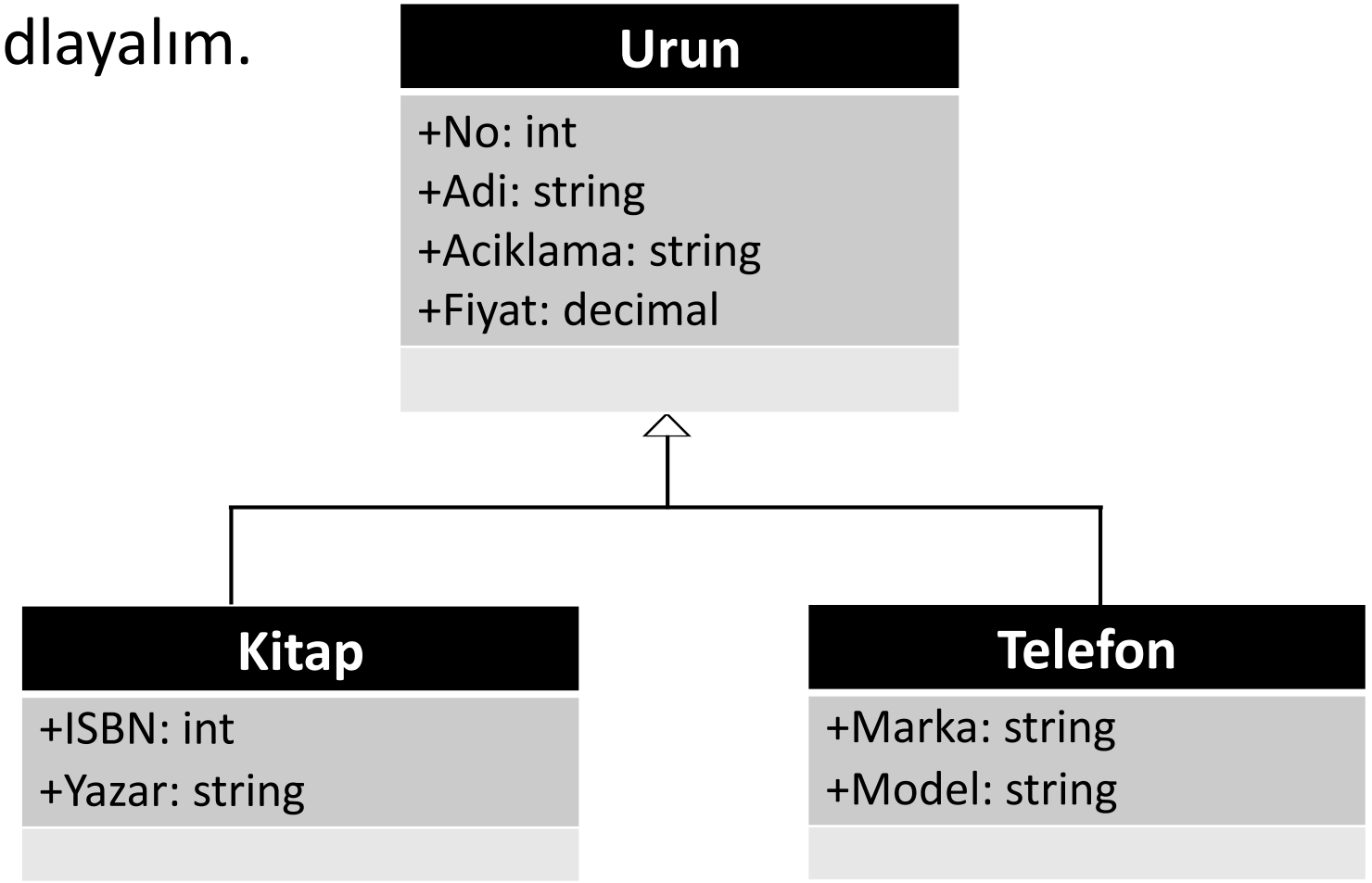
Örnek1: İlk Kalıtım Örneği

Senaryo:

- Sizden **kitap** ve **telefon** satan bir e-ticaret sistemi geliştirmeniz isteniyor. Sistemdeki her **kitap**; ad, açıklama, fiyat, numara, ISBN ve yazar adı bilgisine sahip olmalıdır. Sistemde kayıtlı olacak her **telefon**; ad, açıklama, fiyat, numara, marka ve model bilgilerini barındırmalıdır.

Örnek1: İlk Kalıtım Örneği (devam…)

- Adım adım kodlayalım.



Örnek1: İlk Kalıtım Örneği (devam…)

Adım 1

- Üç sınıfı da ayrı ayrı yaratalım (**Kalıtım bilmiyoruz**).
  - ✓Urun
  - ✓Kitap
  - ✓Telefon
- Form üzerinde üç sınıftan birer tane **nesne** *oluşturalım*.
- Her nesne kaç özelliğe sahip?

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 1

```
public class Urun
{
    0 references
    public int No { get; set; }
    0 references
    public string Adi { get; set; }
    0 references
    public string Aciklama { get; set; }
    0 references
    public decimal Fiyat { get; set; }
}
```

```
public class Kitap
{
    0 references
    public int ISBN { get; set; }
    0 references
    public string Yazar { get; set; }
}
```

```
public class Telefon
{
    0 references
    public string Marka { get; set; }
    0 references
    public string Model { get; set; }
}
```

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 1

```
private void BtnTest_Click(object sender, EventArgs e)
{
    Urun urun = new Urun();
    urun.No = 1;
    urun.Adi = "Ürün 1";

    Kitap kitap = new Kitap();
    kitap.ISBN = 8728323;
    kitap.Yazar = "Yazar1";

    Telefon telefon = new Telefon();
    telefon.Marka = "Samsung";
    telefon.Model = "Galaxy xxx";
}
```

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 2

- Kalıtım / Miras işlemini gerçekleştirelim.
  - ✓Kitap ve Telefon sınıflarını Urun sınıfından miras alalım.
- Form üzerinde oluşturulan nesnelerin özelliklerini **gözlemleyin**.
  - ✓Şimdi her nesne kaç özelliğe sahip?

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 2

```
private void BtnTest_Click(object sender, EventArgs e)
{
    Urun urun = new Urun();
    urun.No = 1;
    urun.Adi = "Ürün 1";

    Kitap kitap = new Kitap();
    kitap.ISBN = 8728323;
    kitap.Yazar = "Yazar1";
    kitap.Adi = "Sindrella";
    kitap.Fiyat = 23 ;

    Telefon telefon = new Telefon();
    telefon.Marka = "Samsung";
    telefon.Model = "Galaxy xxx";
    telefon.Adi = "S6 Edge";
    telefon.Fiyat = 900;
}
```

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 2 Açıklama

- Kitap ve Telefon sınıflarından oluşturulan her bir nesne otomatik olarak Urun sınıfının erişim belirleyicisi **public** olan tüm *özelliklerini* içermektedir.
- Kalıtım **tek yönlü** çalışmaktadır:
  - ✓Yavru sınıf, ana sınıftan kalıtılarak oluşturulur, *ters yönde* **oluşturulamaz**.
  - ✓Program içerisinde bir **Urun nesnesi** oluşturduğunuzda **Kitap sınıfının** özelliklerine veya metotlarına erişemez.

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 3

- Urun sınıfında No özelliğini **read-only** yapalım.
- Urun sınıfına bir tane **Constructor** ekleyelim ve burada No özelliğinin **random** olarak dolmasını sağlayalım.
- Form'da her nesneye ait **No** özellik değerini gösterelim.

Urun
+No: int {Read-Only} +Adi: string +Aciklama: string +Fiyat: decimal
<<Constructor>>+Urun()

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 3

```
public class Urun
{
    4 references
    public int No { get; private set; }
    6 references
    public string Adi { get; set; }
    0 references
    public string Aciklama { get; set; }
    2 references
    public decimal Fiyat { get; set; }
    1 reference
    public Urun()
    {
        Random random = new Random();
        int sayi = random.Next(1, 10000);
        this.No = sayi;
    }
}
```

```
private void BtnTest_Click(object sender, EventArgs e)
{
    Urun urun = new Urun();
    urun.Adi = "Ürün 1";
    MessageBox.Show(urun.Adi + ": " + urun.No);

    Kitap kitap = new Kitap();
    kitap.ISBN = 8728323;
    kitap.Yazar = "Yazar1";
    kitap.Adi = "Sindrella";
    kitap.Fiyat = 23 ;
    MessageBox.Show(kitap.Adi + ": " + kitap.No);

    Telefon telefon = new Telefon();
    telefon.Marka = "Samsung";
    telefon.Model = "Galaxy xxx";
    telefon.Adi = "S6 Edge";
    telefon.Fiyat = 900;
    MessageBox.Show(telefon.Adi + ": " + telefon.No);
}
```

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 3 Açıklama

- *Ana sınıfta* **No özelliği** read-only yapılınca *yavru sınıflarda* da bu özellik read-only oldu.
- Sadece Urun sınıfından oluşturulan nesne mi No özellik değeri **aldı**? (Cevap: **Hayır**)
- Kitap ve Telefon sınıflarından oluşturulan nesneler de birer No **özellik değeri** aldılar.
- **Kalıtım** sadece özelliklerin değil aynı zamanda **metotlar ve kurucuların** da *ana sınıftan* (Urun) miras alınarak yavru sınıflara (Kitap, Telefon) aktarılmasını sağlar.

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 4

- Kitap ve Telefon sınıflarına da **Constructor** ekleyelim.
- Nesneler form üzerinde oluşturulduğunda **SIRAYLA** hangi **Constructların** çalıştığını **DEBUG** işlemi yaparak gözlemleyelim.

Kitap
+ISBN: int +Yazar: string
<<Constructor>>+Kitap()

Telefon
+Marka: string +Model: string
<<Constructor>>+Telefon()

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 4

```
Kitap kitap = new Kitap();   ≤ 3.210ms elapsed
kitap.ISBN = 8728323;
kitap.Yazar = "Yazar1";
kitap.Adi = "Sindrella";
kitap.Fiyat = 23 ;
MessageBox.Show(kitap.Adi + ": " + kitap.No);
```

```
1 reference
public Kitap()   ≤ 1ms elapsed
{

}
```

```
public Urun()
{
    Random random = new Random();   ≤ 1ms elapsed
    int sayi = random.Next(1, 10000);
    this.No = sayi;
}
```

```
1 reference
public Kitap()
{

}   ≤ 1ms elapsed
```

Kitap nesnesi yaratılma aşaması.

Kitap Constructor'ın ilk satırına düşer ancak **tamamlamadan** Urun sınıfının **Constructor'ına** gider.

Urun sınıfının **Constructor'ı tamamlanır.**

Kitap sınıfının **Constructor'ı tamamlanır.**

Örnek1: İlk Kalıtım Örneği (devam…)

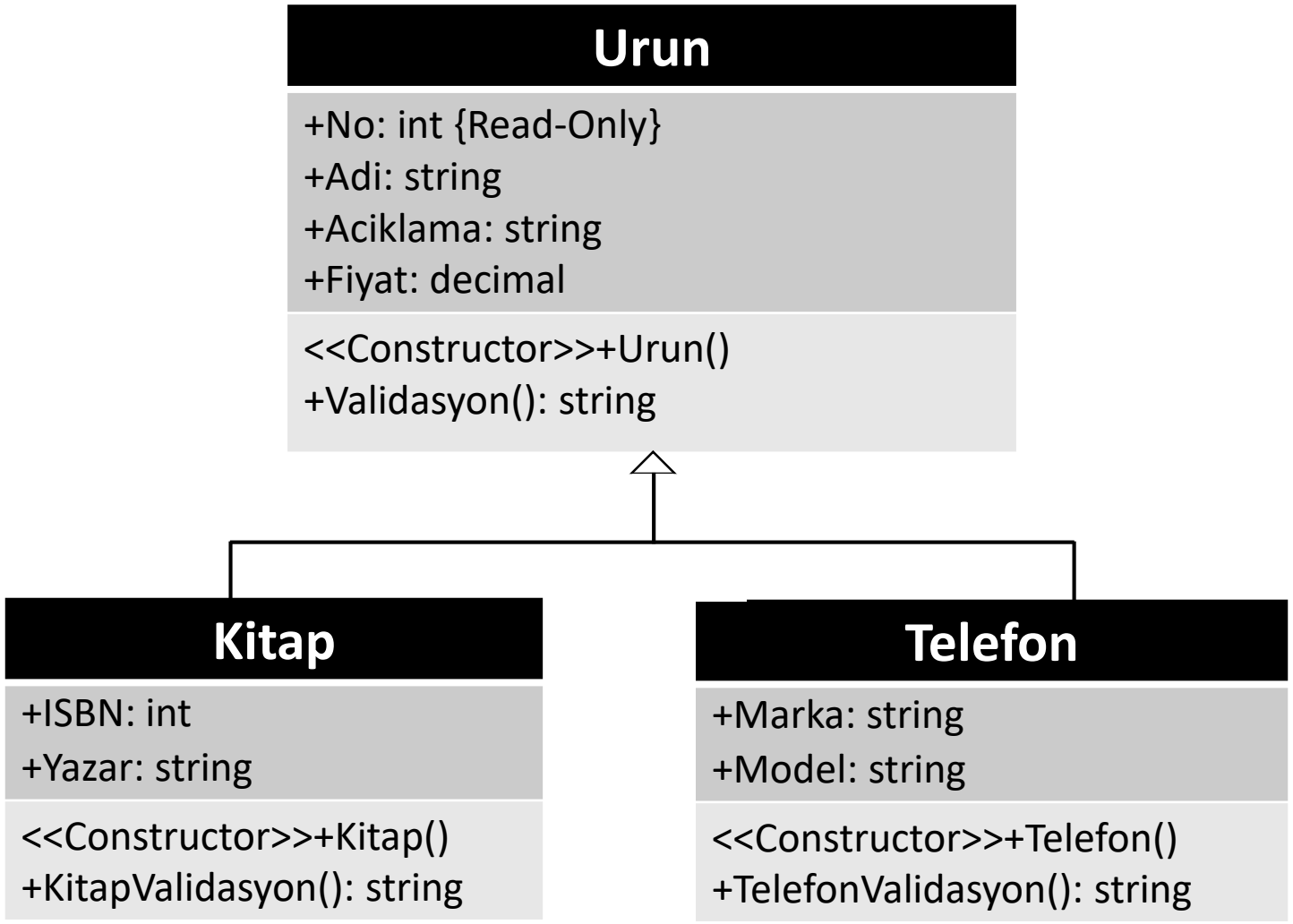
Adım 5

- Bir ürünün **Adı** ve **Fiyatı** boş geçilemez.
  - ✓Bu ürün Kitapsa **ISBN** ve **Yazar** adı,
  - ✓Telefon ise **Model** ve **Marka** özellikleri

  de ayrıca boş olamaz.
- Ürün sınıfına *Validasyon()* isimli bir metot ekleyelim.
- Kitap sınıfına *KitapValidasyon()* isimli bir metot ekleyelim.
  - ✓Temel sınıftan **Validasyon()** metodunu da çağırsın.
- Telefon sınıfına *TelefonValidasyon()* isimli bir metot ekleyelim.
  - ✓Temel sınıftan **Validasyon()** metodunu da çağırsın.

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 5



Örnek1: İlk Kalıtım Örneği (devam…)

Adım 5.1 Çözüm

Ürün sınıfındaki *Validasyon()* metodu

```
public string Validasyon()
{
    string hataMesaji = "";
    if (this.Adi == "")
        hataMesaji += "Ad özelliği boş olamaz.";
    if (this.Fiyat == 0)
        hataMesaji += "Fiyat özelliği 0 olamaz.";
    return hataMesaji;
}
```

Telefon sınıfındaki *TelefonValidasyon()* metodu

```
public string TelefonValidasyon()
{
    return this.Validasyon();
}
```

Kitap sınıfındaki *KitapValidasyon()* metodu

```
public string KitapValidasyon()
{
    return this.Validasyon();
}
```

```
private void BtnTest_Click(object sender, EventArgs e)
{
    Urun urun = new Urun();
    urun.Adi = "Ürün 1";
    //MessageBox.Show(urun.Adi + ": " + urun.No);
    MessageBox.Show("Ürün: " + urun.Validasyon());

    Kitap kitap = new Kitap();
    kitap.ISBN = 8728323;
    kitap.Yazar = "Yazar1";
    kitap.Adi = "";
    kitap.Fiyat = 23 ;
    //MessageBox.Show(kitap.Adi + ": " + kitap.No);
    MessageBox.Show("Kitap: " + kitap.Validasyon());

    Telefon telefon = new Telefon();
    telefon.Marka = "Samsung";
    telefon.Model = "Galaxy xxx";
    telefon.Adi = "S6 Edge";
    telefon.Fiyat = 0;
    //MessageBox.Show(telefon.Adi + ": " + telefon.No);
    MessageBox.Show("Telefon: " + telefon.Validasyon());
}
```

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 5.2 Çözüm

```
public string KitapValidasyon()
{
    string hataMesaji = "";
    if (this.ISBN == 0)
        hataMesaji += "ISBN özelliği boş olamaz.";
    if (this.Yazar == "")
        hataMesaji += "Yazar özelliği boş olamaz.";
    return this.Validasyon() + " / " + hataMesaji;
}
```

```
public string TelefonValidasyon()
{
    string hataMesaji = "";
    if (this.Marka == "")
        hataMesaji += "Marka özelliği boş olamaz.";
    if (this.Model == "")
        hataMesaji += "Model özelliği boş olamaz.";
    return this.Validasyon() + " / " + hataMesaji;
}
```

```
private void BtnTest_Click(object sender, EventArgs e)
{
    Urun urun = new Urun();
    urun.Adi = "Ürün 1";
    //MessageBox.Show(urun.Adi + ": " + urun.No);
    MessageBox.Show("Ürün: " + urun.Validasyon());

    Kitap kitap = new Kitap();
    kitap.ISBN = 8728323;
    kitap.Yazar = "Yazar1";
    kitap.Adi = "";
    kitap.Fiyat = 23 ;
    //MessageBox.Show(kitap.Adi + ": " + kitap.No);
    MessageBox.Show("Kitap: " + kitap.Validasyon());

    Telefon telefon = new Telefon();
    telefon.Marka = "Samsung";
    telefon.Model = "Galaxy xxx";
    telefon.Adi = "S6 Edge";
    telefon.Fiyat = 0;
    //MessageBox.Show(telefon.Adi + ": " + telefon.No);
    MessageBox.Show("Telefon: " + telefon.Validasyon());
}
```

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 6

- Ürün sınıfındaki *Validasyon()* metodunun erişim belirleyicisini **public'ten protected'a** *çekersek* ne olur?
  - ✓**Soru1:** ***Validasyon()*** metodu hala Form üzerinden çağrılabilir mi?
  - ✓**Soru2:** *KitapValidasyon()* bu metodu çağırabilir mi?
  - ✓**Soru3:** *TelefonValidasyon()* bu metodu çağırabilir mi?

Değiştirip Görelim !!!

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 6

- **Soru1:** *Hayır*
- **Soru2:** *Evet*
- **Soru3:** *Evet*

```
protected string Validasyon()
{
    string hataMesaji = "";
    if (this.Adi == "")
        hataMesaji += "Ad özelliği boş olamaz.";
    if (this.Fiyat == 0)
        hataMesaji += "Fiyat özelliği 0 olamaz.";
    return hataMesaji;
}
```

Örnek1: İlk Kalıtım Örneği (devam…)

Adım 6 Açıklama

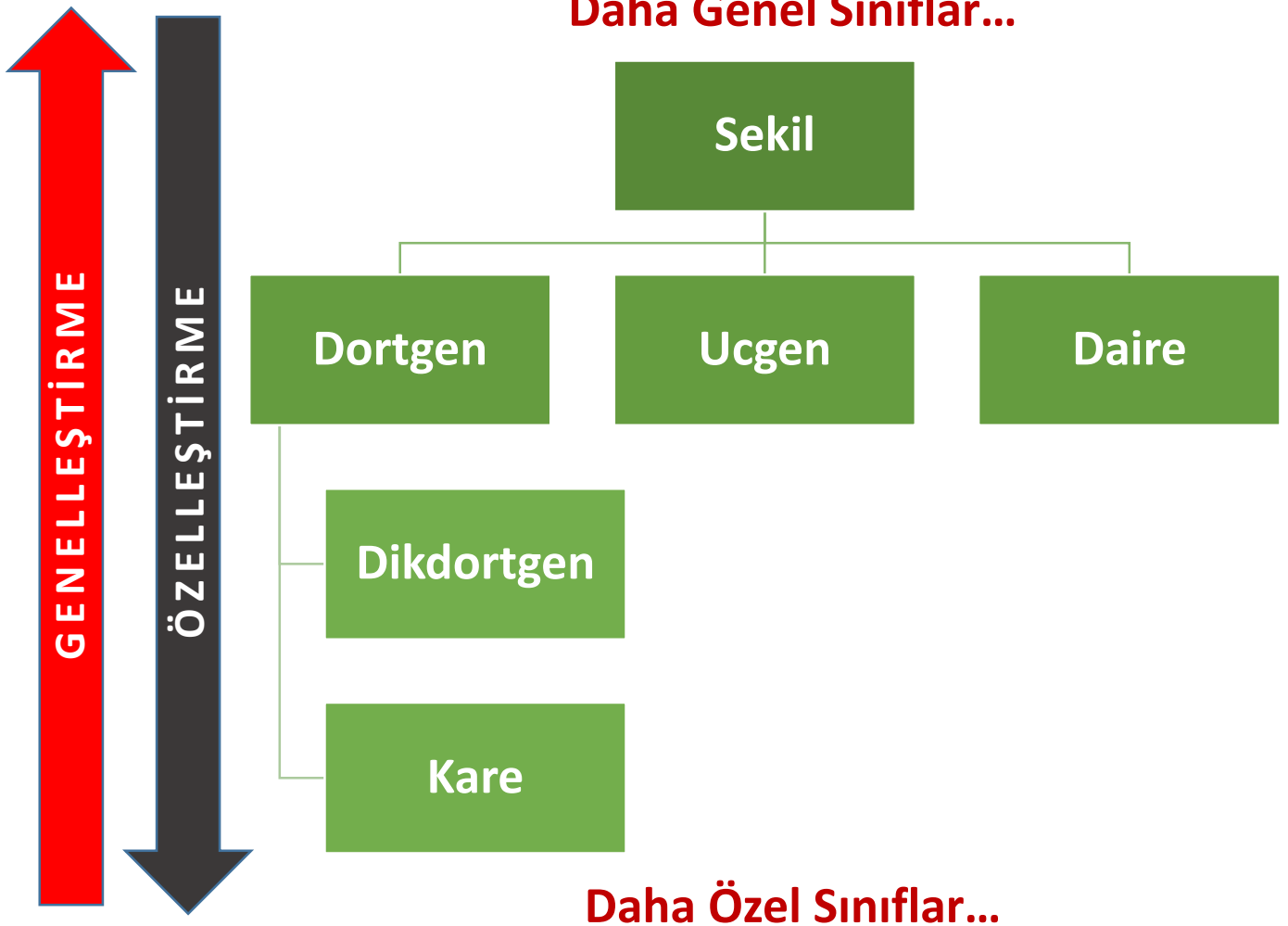
- protected erişim belirleyicisine sahip olan özellikler veya metotlar,
  - Tanımlandıkları sınıfın içerisinde ya da
  - Tanımlı oldukları sınıflardan kalıtımla oluşturulan sınıflar içerisinden erişilebilirler.
- Bu sınıfların **dışında** erişilemezler.
- Diğer bir deyişle, protected erişim belirleyicisine sahip **üyeler** ailenin içerisinde (ana - yavru) *erişilebilirler*.

Sınıfların Genişletilmesi

- Sınıftan türetilerek yeni bir sınıf oluşturulmasını engellemek için sınıf başlığı `sealed` anahtar sözcüğüyle tanımlanır. Hazır olarak gelen String sınıfı sealed sınıflara örnek olarak verilebilir.

```
sealed class sinif
{
}
```

Sınıfların Genişletilmesi (devam…)



- Parent
- SuperClass
- BaseClass

- Child
- SubClass
- DerivedClass

Kalıtım Avantajları

- Kalıtımı kullanabilme kabiliyeti, programı
  - ✓daha kolay ve az kod yazma,
  - ✓daha kolay anlama ve
  - ✓daha az hata ile karşılaşmayı sağlamaktadır.
- Kalıtımı kullanarak, düzgün bir şekilde, hızlıca yeni sınıflar yaratılabilir.

Yararlanılan Kaynaklar

- Sefer Algan , HER YÖNÜYLE C# , Pusula Yayıncılık, İstanbul, 2003
- Volkan Aktaş, HER YÖNÜYLE C# 5.0 , Kodlab Yayıncılık, İstanbul, 2013
- Milli Eğitim Bakanlığı "Nesne Tabanlı Programlama", 2012

İyi Çalışmalar...

Doç. Dr. Deniz Kılınç

deniz.kilinc@bakircay.edu.tr

drdenizkilinc@gmail.com

www.denizkilinc.com