

Programlamaya Giriş

HAFTA 10

Nesneye Dayalı Programlama & Kalıtım

Prof. Dr. Cemil ÖZ

Doç. Dr. Cüneyt BAYILMIŞ

Dr. Öğretim Üyesi Gülüzar ÇİT

Konu & İçerik

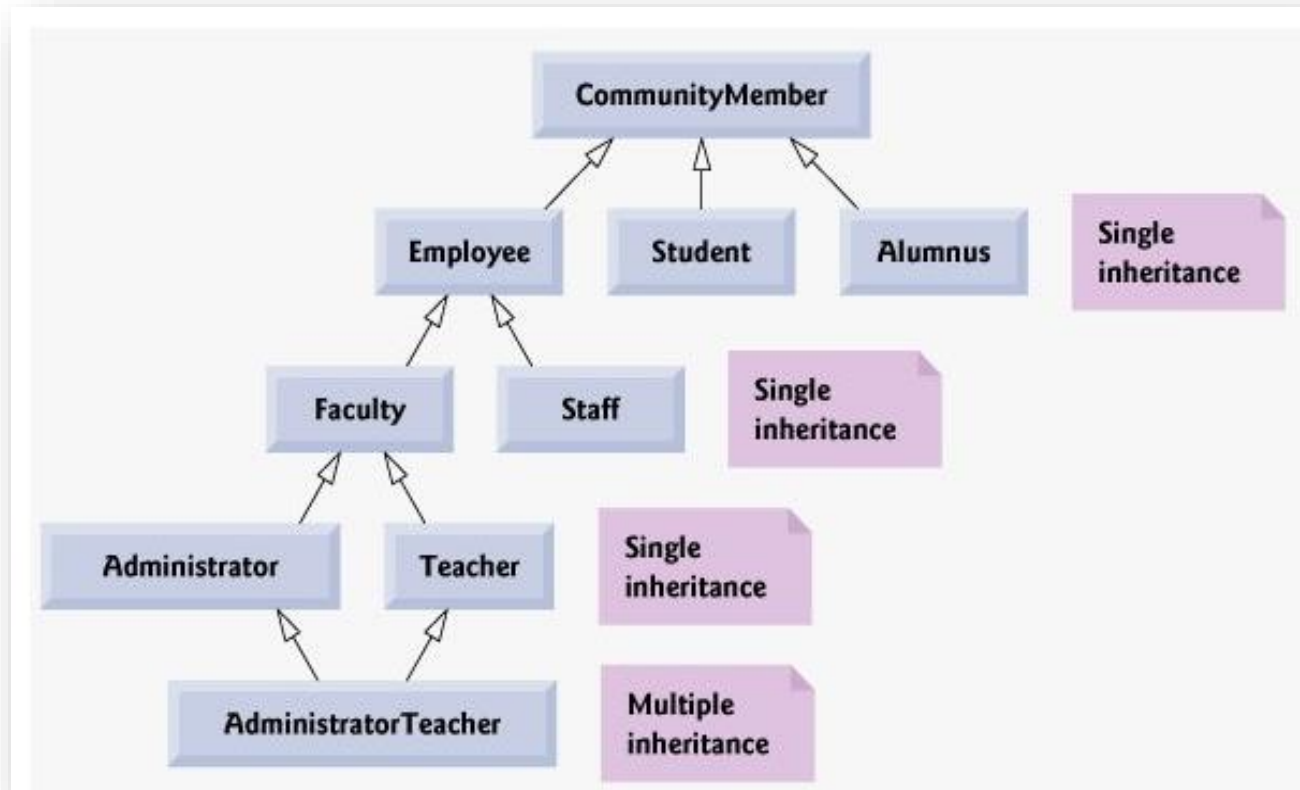
- Kalıtım (Inheritance)
- Kalıtım ve Erişim Düzeyleri (Public Kalıtım)
- Türetilmiş Sınıfta Temel Sınıf Üye Erişilebilirliği
- Nesneler Arası İlişki (Association)
- Kalıtım ve Yapıcılar
- Kalıtım ve Zorlanmış (Overridden) Fonksiyonlar
- Kaynaklar



Kalıtım (Inheritance)

- İki ya da daha fazla sınıf farklı olmakla birlikte benzer özelliklere sahip ise, ortak özelliklerini içeren temel sınıf tanımlanarak diğer sınıflar bu sınıftan türetilir.
- Kodların çok kez kullanımı (Software reusability)
 - Mevcut Sınıflardan Yeni Sınıflar Oluşturma
 - Mevcut sınıfın değişken ve fonksiyonları kullanılabilir.
 - Yeni değişken ve fonksiyonlar eklenebilir.
 - **Temel Sınıf** (Base class) – Mevcut Sınıf
 - **Türetilmiş Sınıf** (Derived class) – Yeni sınıf
 - Türetilmiş sınıf, temel sınıfın özelliklerini **miras alır**

Kalıtım (Inheritance)...



Kalıtım (Inheritance)...

Temel Sınıf Erişim Belirleyicisi

Kalıtım Alınan Temel Sınıf

```
class AsagiSayici : public Sayici
{
public:
    void operator -- ()
    {
        sayi--;
    }
};
```

Kalıtım (Inheritance)...

➤ ÖRNEK: ⇒ [1]_sayici.cpp

```
class Sayici
{
protected:
    unsigned int sayi;
public:
    Sayici() : sayi(0)
    { }
    Sayici(int c) : sayi(c)
    { }
    unsigned int getSayi() const
    {
        return sayi;
    }

    void setSayi(unsigned int sayi)
    {
        this->sayi = sayi;
    }

    void operator ++ ()
    {
        sayi++;
    }
};
```

```
class AsagiSayici : public Sayici
{
public:
    void operator -- ()
    {
        sayi--;
    }
};
```

```
AsagiSayici c1;

cout << "\nc1=" << c1.getSayi();

++c1; ++c1; ++c1;
cout << "\nc1=" << c1.getSayi();

--c1; --c1;
cout << "\nc1=" << c1.getSayi();
cout << endl;
```

Kalıtım (Inheritance)...

➤ Kalıtım ve Erişim Düzeyleri (Public Kalıtım)

- Türetilmiş sınıf üyeleri temel sınıfın private üyelerine doğrudan erişemezler!...

	public	protected	private
Sınıf Üyeleri	X	X	X
Türetilmiş Sınıf Üyeleri	X	X	Gizli
Üye Olmayanlar	X	Gizli	Gizli

Kalıtım (Inheritance)...

➤ Türetilmiş Sınıfta Temel Sınıf Üye Erişilebilirliği

- Türetilmiş sınıf üyeleri tarafından doğrudan erişilebilir.
- Türetilmiş sınıf üyeleri tarafından doğrudan erişilemez.
- Temel sınıfın **public** ve **protected** üyeleri kullanılarak erişilir.

		Kalıtım Türü		
		public	protected	private
Temel sınıf üye erişim belirleyicisi	public	Türetilmiş sınıfta public olur	Türetilmiş sınıfta protected olur	Türetilmiş sınıfta private olur
	protected	Türetilmiş sınıfta protected olur	Türetilmiş sınıfta protected olur	Türetilmiş sınıfta private olur
	private	Türetilmiş sınıfta gizli olur	Türetilmiş sınıfta gizli olur	Türetilmiş sınıfta gizli olur

Kalıtım (Inheritance)...

➤ Nesneler Arası İlişki

➤ 'is a' ilişkisi (kind of)

➤ türetilmiş sınıf nesnesi temel sınıf nesnesi gibidir

➤ araba 'is a' araç

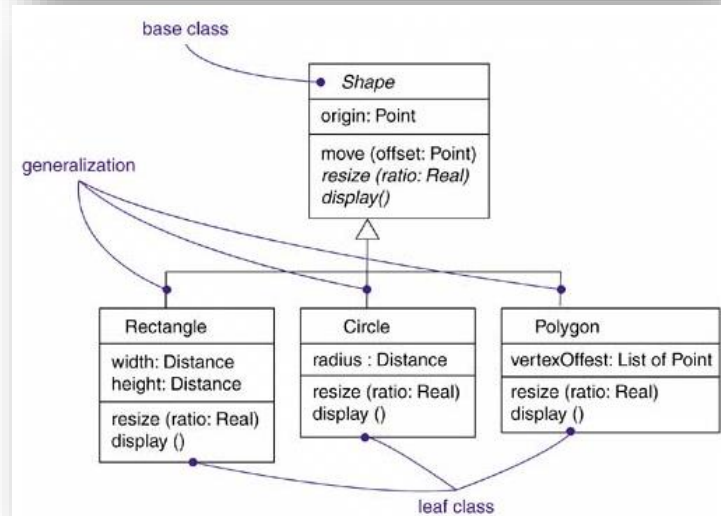
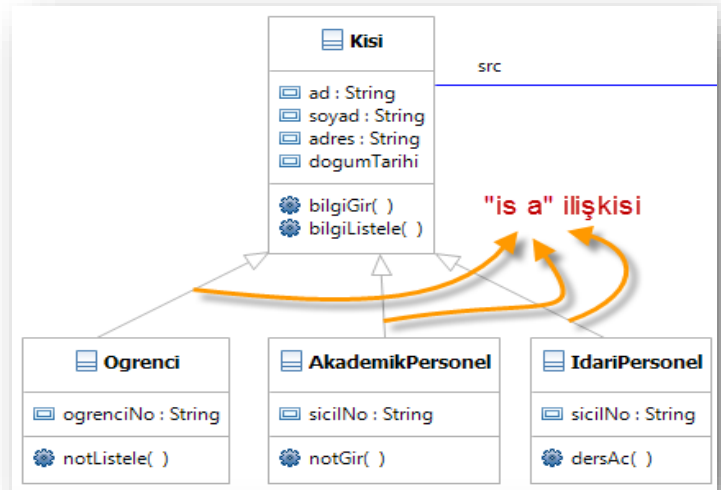
➤ ÖRNEK:

➤ (Temel Sınıf ⇒ Türetilmiş Sınıf)

Dörtgen ⇒ Kare, Dikdörtgen, Paralel Kenar

Araç ⇒ Araba, Kamyon, Tır, Tren

Personel ⇒ Kadrolu, Sözleşmeli, Mevsimlik



Kalıtım (Inheritance)...

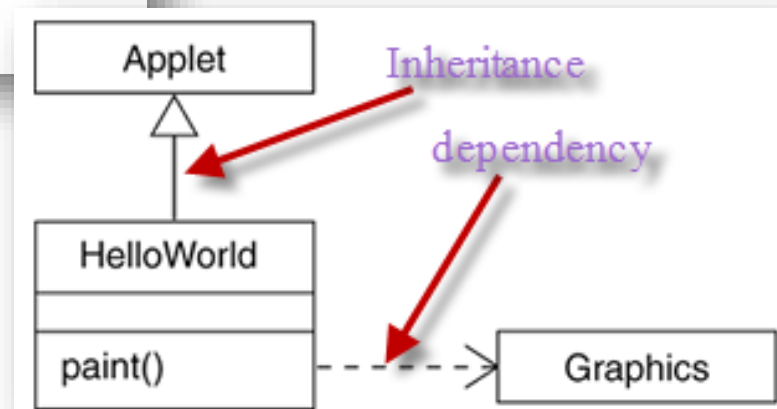
➤ Nesneler Arası İlişki...

➤ Dependency

➤ 'uses a' ilişkisi

➤ Bir nesne diğer nesneye ait fonksiyon içerisinde kullanılır

```
import java.awt.Graphics;  
class HelloWorld extends java.applet.Applet {  
    public void paint (Graphics g) {  
        g.drawString("Hello, World!", 10, 10);  
    }  
}
```

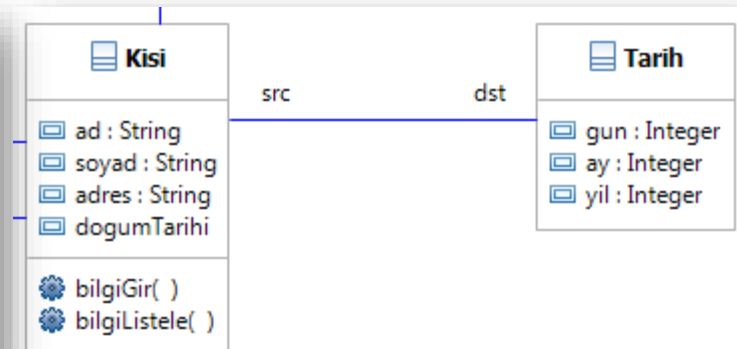
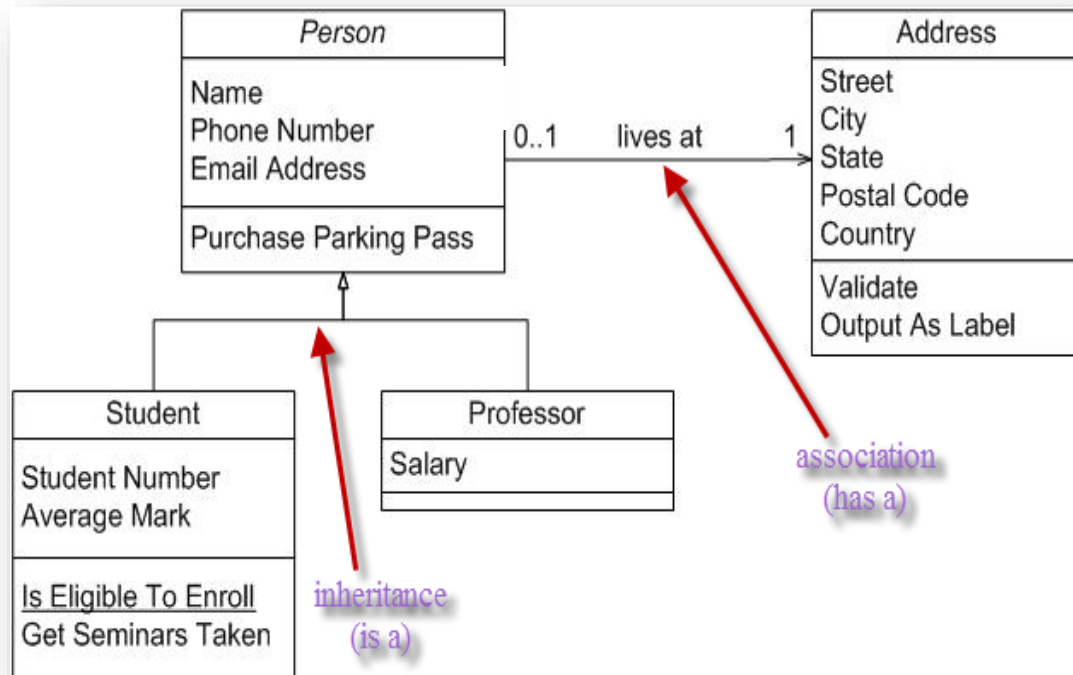


Kalıtım (Inheritance)...

➤ Nesneler Arası İlişki...

➤ Association

➤ 'has a' ilişkisi

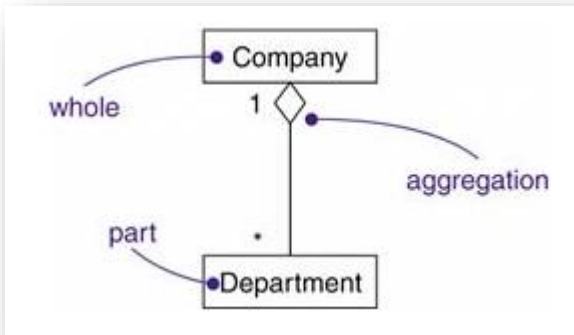


Kalıtım (Inheritance)...

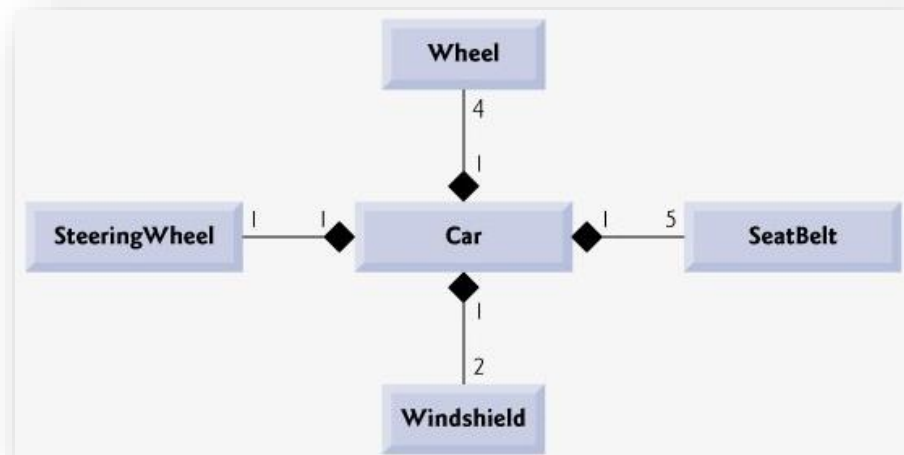
➤ Nesneler Arası İlişki...

➤ Aggregation ve Composition

- **'owns a'** ilişkisi
- **'is part of'** (whole/part)
- Nesneler diğer sınıfların nesnelerini üye olarak içerirler.
- Araba; tekerlek, direksiyon, emniyet kemeri, gaz pedalı v.s.... den oluşur



Daha kuvvetli bir ilişki (Bölüm tek başına da kullanılabilir...)



Kalıtım (Inheritance)...

➤ Kalıtım ve Yapıcılar

- Türetilmiş sınıfa ait bir nesne oluşturulduğunda türetilmiş sınıf nesnesinin yapıcısı çağrılır.
- Bu yapıcının komutları işletilmeden önce temel sınıf yapıcısı çalıştırılır. (Bu işlem kalıtım hiyerarşisinin en üst basamağına kadar yapılır). Türetilmiş sınıf içerisinde temel sınıf yapıcısı çağrılmamışsa varsayılan yapıcı, aksi takdire çağrılan yapıcı çalıştırılır.
- Temel sınıf yapıcısının çalışması bittikten sonra türetilmiş sınıf yapıcısı çalıştırılır.
- Türetilmiş sınıf içerisinde temel sınıf yapıcısı çağırmak için adını ve parametresini yazmak yeterlidir.
- Son çalıştırılan yapıcı, ilk çağrılanıdır.

Kalıtım (Inheritance)...

➤ Kalıtım ve Geçersiz Kılınmış (Overridden) Fonksiyonlar

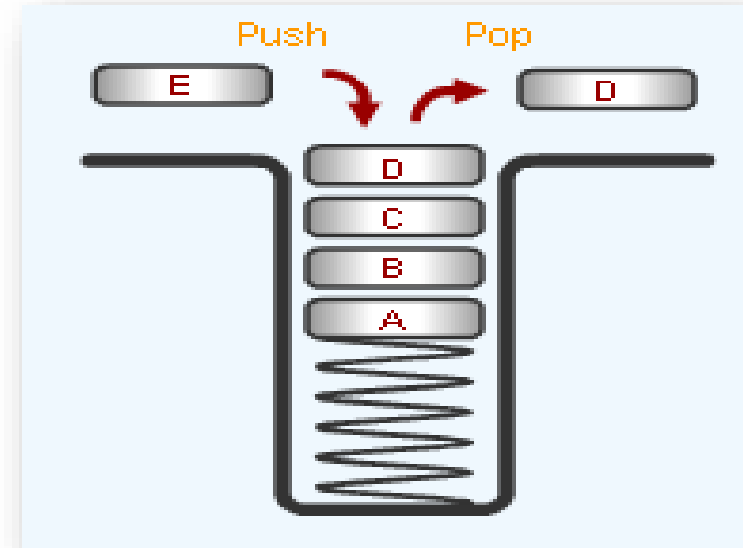
- Bir üye fonksiyonun hem temel sınıf hem de türetilmiş sınıf içerisinde tanımlanmasına denir.
- Bir üye fonksiyon hem temel sınıf hem de türetilmiş sınıf içerisinde tanımlanırsa, türetilmiş sınıftan oluşturulan nesne türetilmiş sınıf içerisindeki fonksiyonu kullanır.
- Türetilmiş sınıf içerisindeki üye fonksiyondan temel sınıfa ait üye fonksiyon çağrılabilir. Bunun için **Temel Sınıf::Üye fonksiyon** yapısı kullanılır.
 - `stack::push(12)`
- Temel sınıfa ait bir nesne oluşturulursa bu nesne türetilmiş sınıf üyelerini kullanamaz.

Kalıtım (Inheritance)...

➤ Yığıt (Stack)

➤ Günlük hayatımızda:

- Kütüphanede raflara konulmak üzere kutuya üst üste yerleştirilmiş kitaplar yığıt yapısı oluşturur.
- Yemekhanede üst üste dizilmiş olarak bulunan tabaklar da yine yığıt yapısına örnek olarak gösterilebilir.
- Yazılımda kullanan yığıt veri yapısında da, gerçek hayattaki örneklerle benzer şekilde, sadece en üstte bulunan elemana erişim imkanı vardır .
- Yığına son giren eleman ilk olarak çıkar (LIFO - Last In First Out).



Kalıtım (Inheritance)...

➤ ÖRNEK: ⇒ [2]_yigit.cpp, yigit.h, gelismisyigit.h

```
typedef int YigitVeriTipi;

const int KAPASITE = 4;

class Yigit
{
private:    int elemanSayisi;
           YigitVeriTipi depolamaBirimi[KAPASITE];
public:
    Yigit();
    bool bosmu() const;
    bool dolumu() const;
    YigitVeriTipi pop();
    void push(YigitVeriTipi);
    int getElemanSayisi() const;
    void setElemanSayisi(int elemanSayisi);
};
```

```
class GelismisYigit : public Yigit
{
public:
    GelismisYigit();
    YigitVeriTipi pop();
    void push(YigitVeriTipi);
};
```

```
GelismisYigit veriYapisi;

//veriYapisi.pop();
veriYapisi.push(10);
veriYapisi.push(20);
veriYapisi.push(30);
veriYapisi.push(40);
veriYapisi.push(50);
while (!veriYapisi.bosmu())
    cout << veriYapisi.pop() << endl;
```


KAYNAKLAR

- Deitel, C++ How To Program, Prentice Hall
- Horstmann, C., Budd, T., Big C++, Jhon Wiley & Sons, Inc.
- Robert Lafore, Object Oriented Programming in C++, Macmillan Computer Publishing
- Prof. Dr. Celal ÇEKEN, Programlamaya Giriş Ders Notları
- Prof. Dr. Cemil ÖZ, Programlamaya Giriş Ders Notları