

Algoritma ve Programlama - II

Kalıtım (Inheritance)



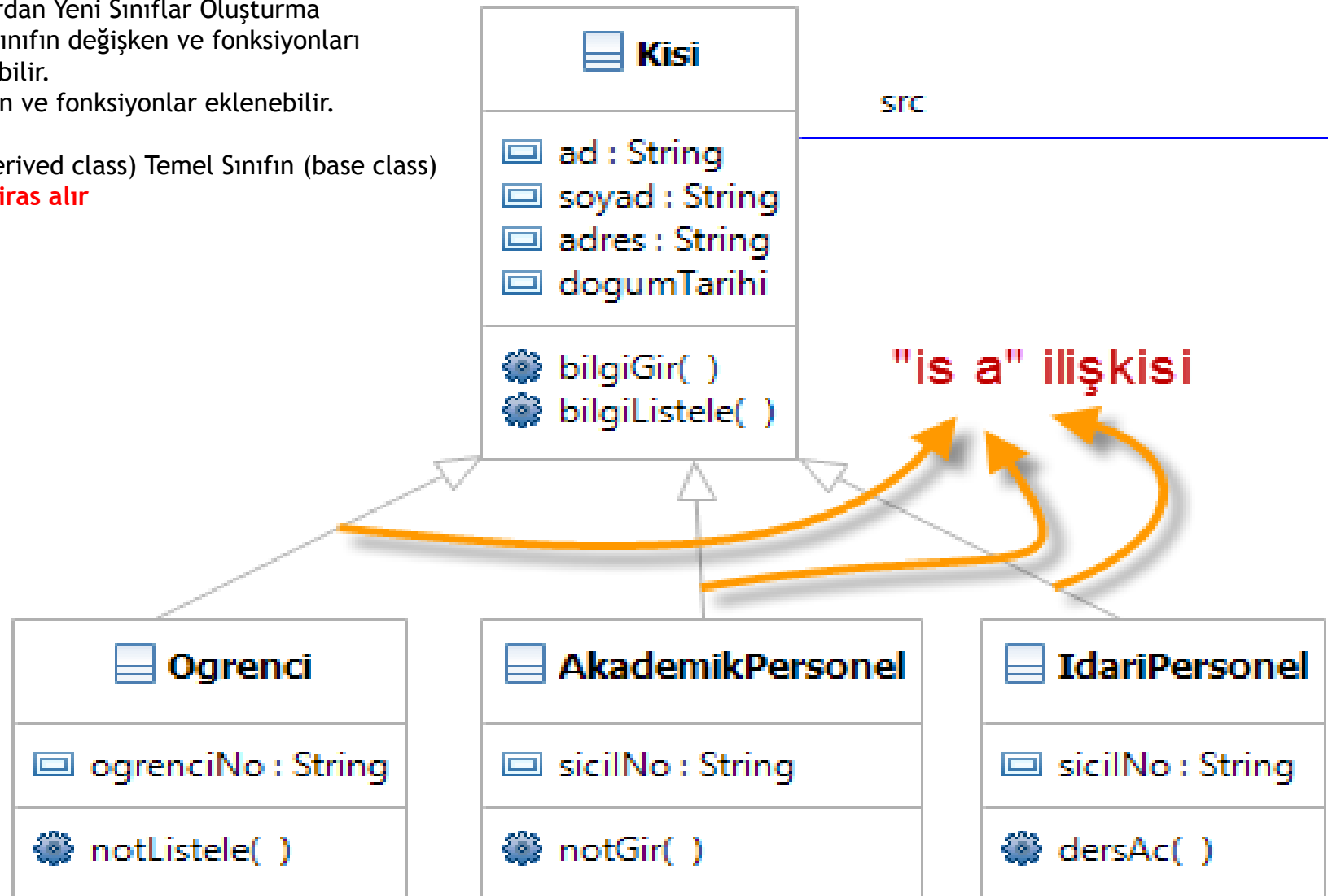
Konular

- ✓ 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
- ✓ Çoklu Kalıtım
- ✓ Çok Şekillilik (Polymorphism)
- ✓ Sanal Fonksiyonlar (Virtual Functions)
- ✓ Veritabanı Bağlantısı
- ✓ Kaynaklar

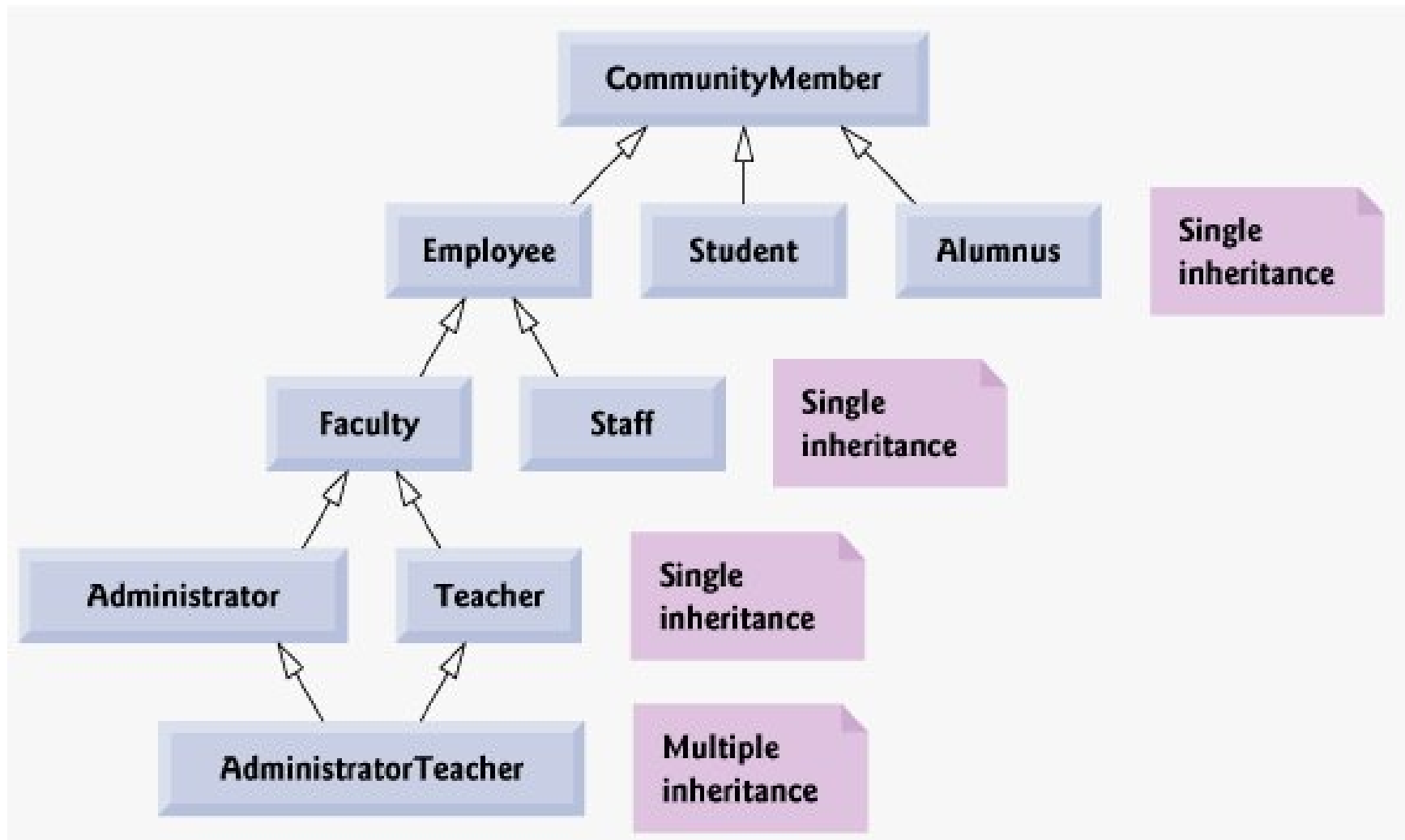
Kalıtım (Inheritance)

Kodların çok kez kullanımı (Software reusability)
Mevcut Sınıflardan Yeni Sınıflar Oluşturma
Mevcut sınıfın değişken ve fonksiyonları kullanılabilir.
Yeni değişken ve fonksiyonlar eklenebilir.

Türetilmiş sınıf (Derived class) Temel Sınıfın (base class) özelliklerini **miras alır**



Kalıtım (Inheritance)



Deitel, C++ How To Program, Prentice Hall kitabından alınmıştır

Kalıtım (Inheritance)

Kalıtım ve Erişim Düzeyleri (Public Kalıtım)

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

Tüderilmiş sınıf üyeleri temel sınıfın private üyelerine doğrudan erişemezler.

Türetilmiş Sınıfta Temel Sınıf Üye Erişilebilirliği

		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.

Türetilmiş sınıf üyeleri tarafından doğrudan erişilebilir.

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

Nesneler Arası İlişki

1. Kalıtım (Inheritance)

2 ya da daha fazla sınıf farklı olmakla birlikte benzer özelliklere sahip ise, ortak özelliklerini içeren temel sınıf tanımlanarak diğer sınıflar bu sınıftan türetilir.

'is a' ilişkisi (kind of) →

kalıtım

türetilmiş sınıf nesnesi temel sınıf nesnesi gibidir
araba 'is a' araç

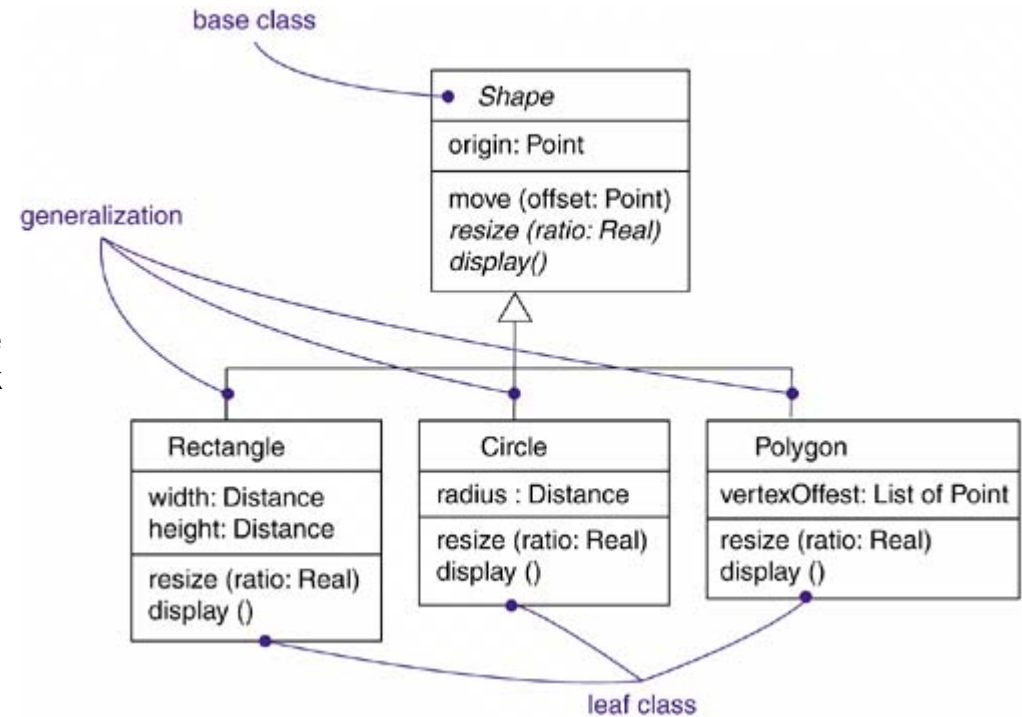
Kalıtım Örnekler

Temel Sınıf : Türetilmiş Sınıf

Dörtgen : Kare, Dikdörtgen, Paralel Kenar

Araç : Araba, Kamyon, Tır, Tren

Personel: Kadrolu, Sözleşmeli, Mevsimlik

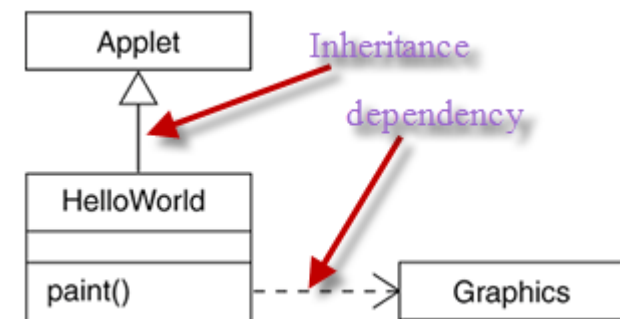


```
import java.awt.Graphics;
class HelloWorld extends java.applet.Applet {
    public void paint (Graphics g) {
        g.drawString("Hello, World!", 10, 10);
    }
}
```

2. Dependency

“uses a” ilişkisi

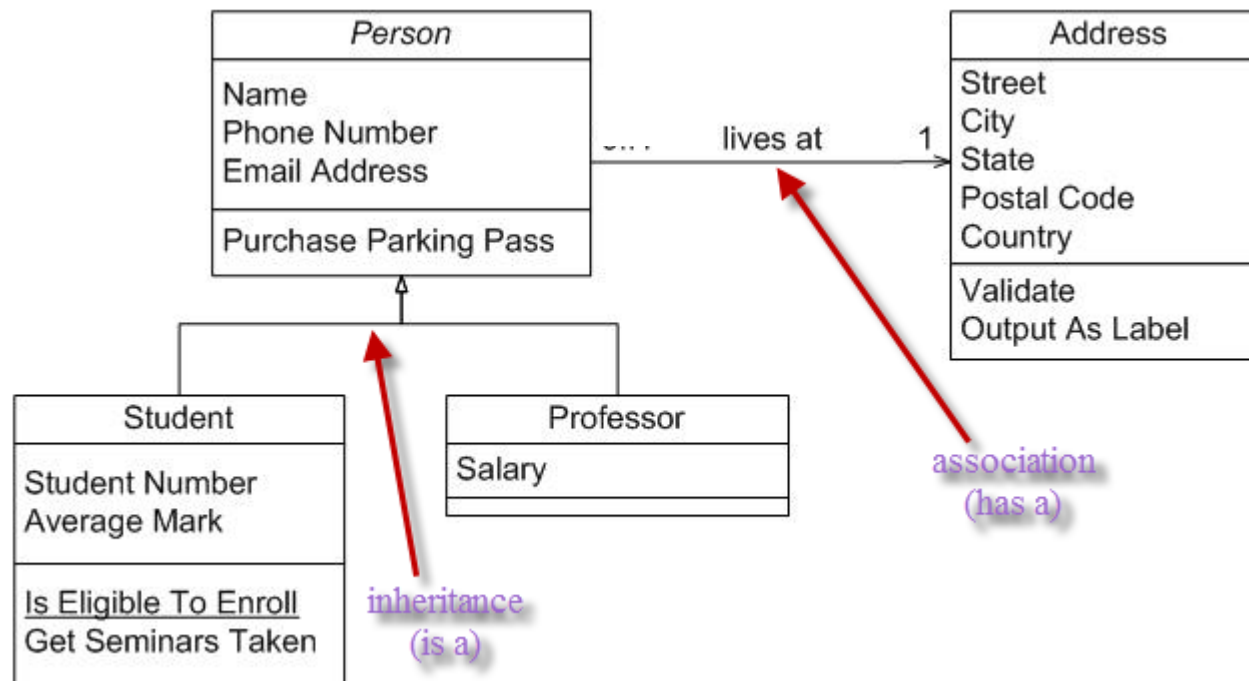
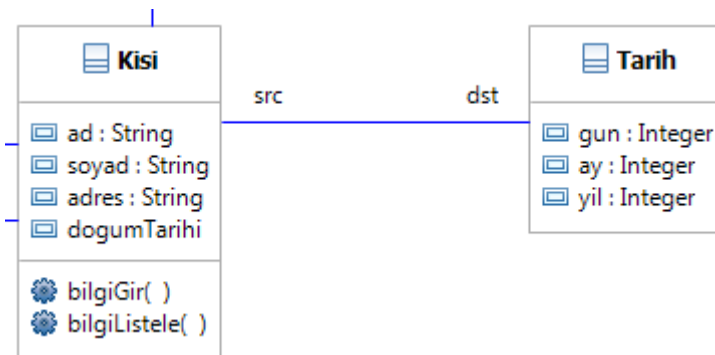
Bir nesne diğer nesneye ait fonksiyon içerisinde kullanılır



Nesneler Arası İlişki (Association)

3. Association

'has a' ilişkisi



Nesneler Arası İlişki (Association)

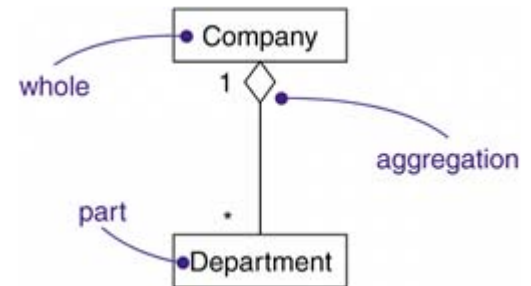
4. Aggregation ve Composition

“owns a”

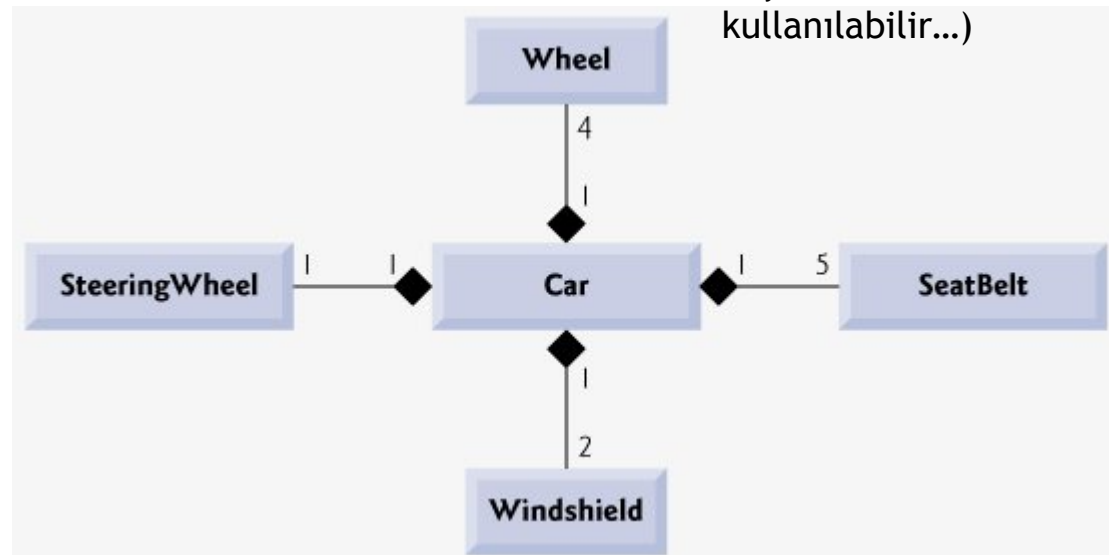
“is part of” (whole/part)→

Nesneler diğer sınıfların nesnelerini üye olarak içerirler.

araba tekerlek, direksiyon, emniyet kemeri, gaz pedalı v.s.... den oluşur



Daha kuvvetli bir ilişki (Bölüm tek başınada kullanılabilir...)



Kalıtım ve Yapıcılar

- ✓ 1. Türetilmiş sınıfa ait bir nesne oluşturulduğunda türetilmiş sınıf nesnesinin yapıcısı çağrılır.
- ✓ 2. 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.
- ✓ 3. Temel sınıf yapıcısının çalışması bittikten sonra türetilmiş sınıf yapıcısı çalıştırılır.
- ✓ 4. Türetilmiş sınıf içerisinde temel sınıf yapıcısı çağırmak için adını ve parametresini yazmak yeterlidir.
- ✓ 5. Son çalıştırılan yapıcı, ilk çağrılanıdır.

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.

Yığıt Uygulaması

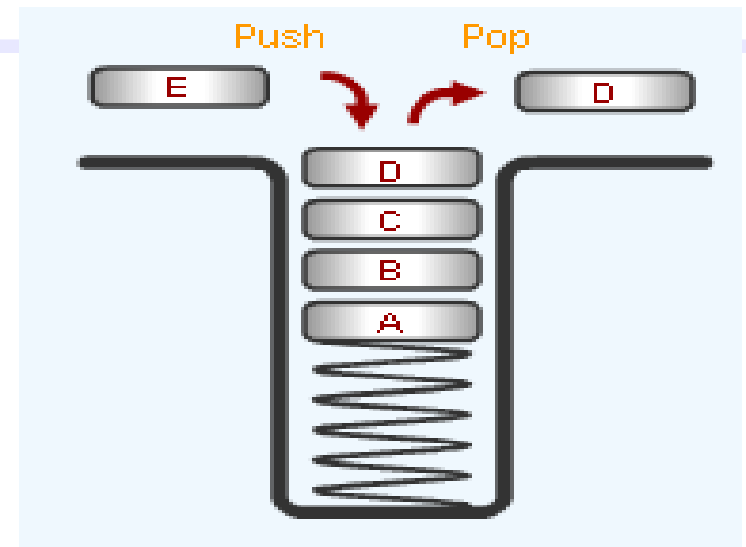
✓Günlük hayatımızda:

- ✓ 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 örneklere benzer şekilde, sadece en üstte bulunan elemana erişim imkanı vardır .

✓Yığına son giren eleman ilk olarak çıkar. LIFO (Last In First Out)

```
typedef double YigitVeriTipi;  
  
enum HataKodu { basarili, bos, dolu};  
  
const int mAXeLEMANsAYISI = 10;  
  
class Yigit  
{  
private:  
    int elemanSayisi;  
    YigitVeriTipi depolamaBirimi[mAXeLEMANsAYISI];  
public:  
    Yigit();  
    bool bos() const;  
    YigitVeriTipi pop();  
    void push(YigitVeriTipi);  
};
```



Kaynaklar

- ✓ Robert Lafore, Object Oriented Programming in C++, Macmillan Computer Publishing
- ✓ Deitel, C++ How To Program, Prentice Hall