

# Nesneye Yönelimli Programlama – BLM 205

## Hafta 7: Çok Biçimlilik



Fenerbahçe Üniversitesi

# Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: [emre.levent@fbu.edu.tr](mailto:emre.levent@fbu.edu.tr)

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 307

Email: [ugur.ozbalkan@fbu.edu.tr](mailto:ugur.ozbalkan@fbu.edu.tr)

Asistan: Arş. Gör. Ecenur Alioğulları

Ofis: 307

Email: [ecenur.aliogullari@fbu.edu.tr](mailto:ecenur.aliogullari@fbu.edu.tr)

# Ders Planı

- Çok Biçimlilik
  - Polymorfik fonksiyonlar
  - Sınıflar ve polymorfizm
  - Metod geçersiz kılma

## Çok Biçimlilik

- Çok biçimlilik, genel olarak bir nesnenin farklı durumlar altında farklı davranışlar sergilemesidir.
- Örneğin, bir insan hem baba, hem eş hem de çalışan olabilir. Bu insan'ın konuşma hareketi, farklı durumlarda farklı olacaktır.

# Çok Biçimlilik

- Objeye yönelik programlama dillerinde polymorphism, aynı isimli fonksiyonların birden çok gerçekleşmesinin olabileceğidir.

# Çok Biçimlilik

- İki farklı sınıfta aynı isimde method'lar olabilir. Sınıflardan bir liste oluşturulup, bir döngüde methodlar çağırılabilir.

# Çok Biçimlilik

## Sınıf metodlarının çok biçimliliği

### Örnek Kod Parçasığı

```
class Turkiye():  
    def baskent(self):  
        print("Ankara")  
  
    def dili(self):  
        print("Turkce")  
  
class ABD():  
    def baskent(self):  
        print("Washington")  
  
    def dili(self):  
        print("Ingilizce")  
  
ulkeler = [Turkiye() , ABD()]  
  
for ulke in (ulkeler):  
    ulke.baskent()  
    ulke.dili()
```

### Çıktı

```
Ankara  
Turkce  
Washington  
Ingilizce
```

# Çok Biçimlilik

- Çalışma esnasında, aynı isimli method çağırılmasına rağmen elemanın türüne göre farklı fonksiyonlar çağrılmaktadır.



## Çok Biçimlilik

- Bir sınıf, ana sınıftan türetilirken, ana sınıftaki fonksiyon ismi ile kendinde de fonksiyon barındırıyor olabilir.
- Kalıtım yapıldığında, ana sınıftaki fonksiyon geçersiz kılınır.

# Çok Biçimlilik

## Metod Geçersiz Kılma

### Örnek Kod Parçasığı

```
class kuslar:  
    def testFonksiyonu(self):  
        print("Cok sayida kus turu vardır")  
  
    def ucus(self):  
        print("Bazi kuslar uçamaz")  
  
class guvercin(kuslar):  
    def ucus(self):  
        print("Guvercinler Uçabilir")  
  
class tavuk(kuslar):  
    def ucus(self):  
        print("Tavuklar uçamaz")
```

### Örnek Kod Parçasığı

```
kusObj = kuslar()  
guvercinObj = guvercin()  
tavukObj = tavuk()  
  
kusObj.testFonksiyonu()  
kusObj.ucus()  
  
guvercinObj.testFonksiyonu()  
guvercinObj.ucus()  
  
tavukObj.testFonksiyonu()  
tavukObj.ucus()
```

Ana sınıftaki ucus fonksiyonunu geçersiz kıldılar

### Çıktı

Cok sayida kus turu vardır  
Bazi kuslar uçamaz  
Cok sayida kus turu vardır  
Guvercinler Uçabilir  
Cok sayida kus turu vardır  
Tavuklar uçamaz

# Çok Biçimlilik

- Fonksiyonlarda çok biçimlilik, kendisine verilen farklı türde argümanlara göre çalışabilmesidir.

# Çok Biçimlilik

## Fonksiyonlarda Çok Biçimlilik

### Örnek Kod Parçasığı

```
class Turkiye():  
    def baskent(self):  
        print("Ankara")  
  
    def dili(self):  
        print("Turkce")  
  
class ABD():  
    def baskent(self):  
        print("Washington")  
  
    def dili(self):  
        print("Ingilizce")
```

### Örnek Kod Parçasığı

```
def func(obj):  
    obj.baskent()  
    obj.dili()  
  
turkObj = Turkiye()  
abdObj = ABD()  
  
func(turkObj)  
func(abdObj)
```

### Çıktı

```
Ankara  
Turkce  
Washington  
Ingilizce
```

Fonksiyon çalışma zamanında farklı türde argümanlar alıyor

# Çok Biçimlilik

## Örnek

### Örnek Kod Parçasığı

```
from math import pi

class sekiller( object ):
    def __init__(self, isim):
        self.isim = isim

    def alan(self):
        pass

class kare(sekiller):
    def __init__(self, uzunluk):
        sekiller.__init__(self, "Kare")
        self.uzunluk = uzunluk

    def alan(self):
        return self.uzunluk**2
```

### Örnek Kod Parçasığı

```
class cember(sekiller):
    def __init__(self, cap):
        sekiller.__init__(self,
"Cember")
        self.cap = cap

    def alan(self):
        return pi*self.cap**2

a = kare(4)
b = cember(7)

print(a.alan())
print(b.alan())
```

### Çıktı

```
16
153.93804002589985
```

Soyut (Abstract) fonksiyon, kendi başlarına kullanılmaları anlamsızdır.