

# **ÖZET ve TEKRAR**

Doç. Dr. Aybars UĞUR

# **Koleksiyonlar** (veya Topluluklar)

Collections

# ArrayList Sınıfı

```
// using System.Collections; kullanmak gerekir!
```

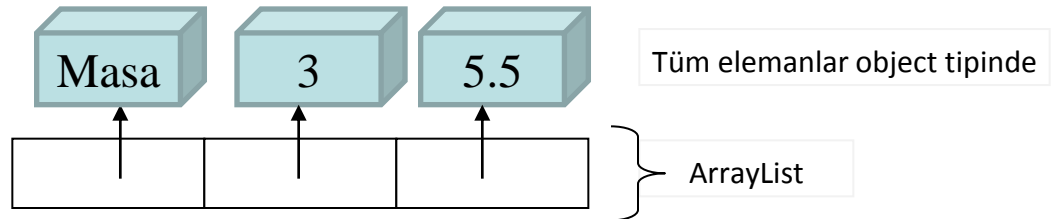
```
ArrayList liste = new ArrayList();
```

```
// Her Tür elemanı alabilir : object tipinde
```

```
liste.Add("Masa");
```

```
liste.Add(3);
```

```
liste.Add(5.5);
```



```
// İlgili indis elemanına doğrudan erişilir
```

```
Console.WriteLine(liste[2]); // Java'da [] ile erişme yok!
```

```
// elementAt(2) gibi bir metot aracılığı ile indis erişimi var
```

```
// Operator Overloading ve İndeksleyici
```

```
// Ancak, double sayi = liste[2]; olmaz.
```

```
// Kutudan çıkarmak gerekir,
```

```
// tüm elemanları objecte dönüştüğünden
```

```
double sayi = (double) liste[2];
```

# List Sınıfı

Listemiz tek tip elemandan oluşuyorsa, kutulama ve kutudan çıkarma için zaman harcamamıza, koda bu işlemleri eklememize gerek yok. Generic List sınıfını kullanırız. Hata yapma olasılığını da aza indirir.

```
List<string> liste2 = new List<string>();  
liste2.Add("Merhaba");  
Console.WriteLine(liste2[0].ToUpper());
```

Ekran Çıktısı :  
MERHABA



object sınıfının ToUpper metodu yok. Ancak listenin tüm elemanları string, böylece kutudan çıkarmadan string sınıfı metotlarına erişebiliyoruz.

# Dizi kullansak olmaz mı?

Dizilerin de tüm elemanları aynı tiptendir.

Ancak, boyutları daraltılıp genişletilemez. Araya eleman ekleme ve çıkarmalarda ekstra işlem yükü ve kod yazmayı gerektirir.

# **Yeni Bir Programlama Dili Öğrenmek Başlangıç**

- Dersleri dinleyip olabildiğince anlamak.
- Kaynakların yerini bilmek ve ulaşabilir olmak (fiziksel veya internet üzerinden).
- Programlama ortamını karıştırmak ve denemeler, küçük örnekler yapmak.
- Ders kitabı veya diğer kaynaklardan da okumak.
- Problemleri koda dökerken, önce kağıt üzerinde tasarlamak, gerekiyorsa hazır örnekleri incelemek, anlamaya çalışmak, nedenlerini düşünmek.
- Uygun kişilere soru sormak.

# Öneriler - Özet

- Kendinize göre olabildiğince çalışma ortamınızı ayarlayın.
- Günü gününe ve planlı çalışın, ders öncesi veya sonrasında ders notları ile ders kitabından ilgili bölümleri okuyun.
- Gerektiğinde soru çözerek, yazarak, özet çıkararak çalışın.
- Kodları bilgisayara doğrudan yazmayın, öncesinde kağıda (A4) veya deftere yazarak tasarlayın. Neden?
- Öğrendikçe, kağıt üzerinde tasarıma ihtiyaç da azalacaktır.
- Değerleri program içerisinde yazdırmak yerine Debug özelliklerini etkin kullanın.

# **Form Application, GUI, Controls**