

CENG 201 Veri Yapıları 1: C# Tekrarı

Öğr.Gör. Şevket Umut ÇAKIR

Pamukkale Üniversitesi

Hafta 1

Anahat

① Ders Bilgisi

② Giriş

C# Hatırlatma

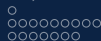
Kitap, Üye ve Ödünç Sınıfları

Kütüphane Sınıfı

③ Generic Tipler ve Koleksiyonlar < T >

Ders Saatleri

- Ders Saatleri
 - Pazartesi 14:25 - 16:55 tüm şubeler için teorik ders (ÖİDB BB 109)
 - Salı 08:55 - 10:35 Şube 1 pratik ders (A0434)
 - Salı 10:45 - 12:25 Şube 2 pratik ders (A0434)



Ders Materyalleri

- 1 Ders Sunumları
- 2 Weiss, M. A., Data Structures and Algorithm Analysis in Java, 3rd Ed., Pearson Education, 2012
- 3 Yıldız O. T., C & Java ile Veri Yapılarına Giriş, Boğaziçi Üniversitesi Yayınevi, 2013
- 4 Çölkesen R., Veri Yapıları ve Algoritmalar, Papatya Bilim Yayınevi, 2014
- 5 McMillan M., Data Structures and Algorithms Using C#, Cambridge Press, 2007

Değerlendirme Yöntemleri

- Ara sınav: % 35
- Dönem sonu sınavı: % 40
- Ödevler: % 25

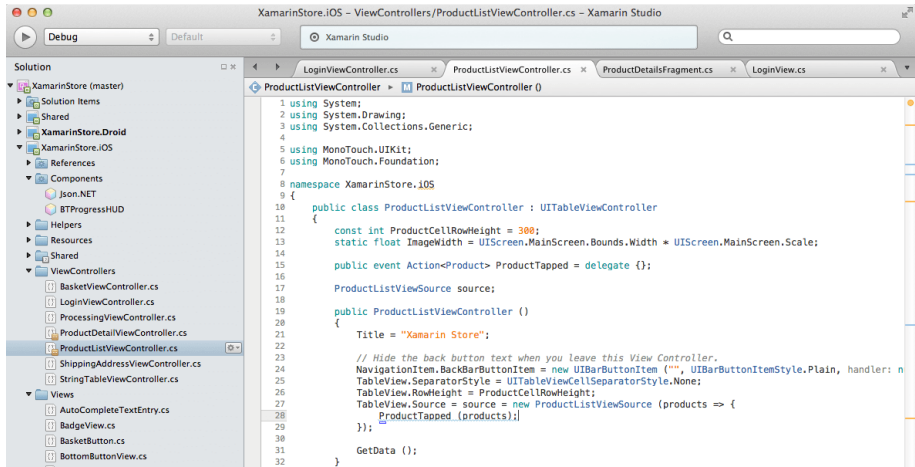
Kodlar

<https://github.com/sevketcakir/ds2017>



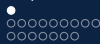
Kullanılacak IDE

Xamarin Studio/Monodevelop



Xamarin/Monodevelop Özellikleri

Özellik	Linux	Mac	Windows
C#	Yes	Yes	Yes
Visual Basic	Yes	Yes	Yes
C/C++	Yes	Yes	No
Python	Yes	No	No
ASP.NET Project	Yes	Yes	Yes
ASP.NET MVC Project	Yes	Yes	Yes
iPhone Project	No	Yes	No
Gtk# designer	Yes	Yes	Yes
Database Explorer	Yes	Yes	No
Debugging (managed)	Yes	Yes	Yes
ASP.NET Debugging	Yes	Yes	Yes

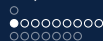


Nesneye Yönelik Kütüphane Yazılımı

Kütüphane Yazılımı

Kitapların ve üyelerin kayıtlarının tutulduğu, üyelerin kitapları ödünç aldığı bir kütüphane yazılımı yapılmak istenmektedir. Bu yazılım için gerekli aşağıdaki sınıflar yazılacaktır.

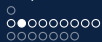
- Kitap
- Uye
- Odunc
- Kutuphane



Kitap Sınıfı

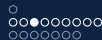
Kitap sınıfının aşağıdaki özellikleri bulunacaktır. Kurucu metodlar(constructor) ve her bir eleman için bir *get* ve *set* metodları yazılacaktır.

- ISBN: Kitabın uluslararası standart kitap numarası
- Ad: Kitabın adı
- Yazar: Kitabı yazan
- Alan: Kitabın alanı/kategorisi



Kitap.cs I

```
1 using System;
2
3 namespace VYDers1
4 {
5     public class Kitap
6     {
7         string isbn;
8         public string ISBN {
9             get { return this.isbn; }
10            set { isbn = value; }
11        }
12        string ad;
13        public string Ad {
14            get { return this.ad; }
15            set { ad = value; }
16        }
17        string yazar;
18        public string Yazar {
19            get { return this.yazar; }
20            set { yazar = value; }
21        }
22    }
23 }
```



Kitap.cs II

```
23     public string Alan {
24         get { return this.alan; }
25         set { alan = value; }
26     }
27     public Kitap(string isbn, string ad, string yazar, string alan)
28     {
29         this.isbn = isbn;
30         this.ad = ad;
31         this.yazar = yazar;
32         this.alan = alan;
33     }
34     public Kitap () {} //Default constructor
35     public override string ToString()
36     {
37         return String.Format ("{0}, {1}, {2}, {3}", isbn, ad, yazar,
↵     alan);
38     }
39     } //class sonu
40 } //namespace sonu
```



Üye Sınıfı

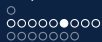
Üye sınıfının aşağıdaki özellikleri bulunacaktır. Kurucu metodlar(constructor) ve her bir eleman için bir *get* ve *set* metodları yazılacaktır.

- uyeNo: Üyenin numarasını tutacaktır
- Ad: Üyenin adı
- Adres: Üyenin adresi



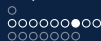
Uye.cs I

```
1 using System;
2
3 namespace VYDers1
4 {
5     public class Uye
6     {
7         int uyeNo;
8         string ad;
9         string adres;
10        public int UyeNo {
11            get { return this.uyeNo; }
12            set { uyeNo = value; }
13        }
14
15        public string Ad {
16            get { return this.ad; }
17            set { ad = value; }
18        }
19    }
```



Uye.cs II

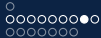
```
20     public string Adres {
21         get { return this.adres; }
22         set { adres = value; }
23     }
24     public Uye (int uyeNo, string ad, string adres)
25     {
26         this.uyeNo = uyeNo;
27         this.ad = ad;
28         this.adres = adres;
29     }
30
31     public Uye () {}
32     public override string ToString ()
33     {
34         return string.Format ("[Uye: UyeNo={0}, Ad={1}, Adres={2}]",
↪      UyeNo, Ad, Adres);
35     }
36
37 }
38 }
```



Ödunc Sınıfı

Ödünç sınıfı kütüphaneden ödünç alınan kayıtları saklayacaktır ve aşağıdaki elemanları içerir. Kurucu metodlar(constructor) ve her bir eleman için bir *get* ve *set* metodları yazılacaktır.

- Uye: Ödünç alan üyenin nesnesi
- Kitap: Ödünç alınan üyenin nesnesi
- Zaman: Ödünç alma zamanı. Kurucu metodla otomatik olarak oluşturulacaktır



Odunc.cs I

```
1 using System;
2
3 namespace VYDers1
4 {
5     public class Odunc
6     {
7         Uye uye;
8         Kitap kitap;
9         DateTime zaman;
10        public Odunc () { }
11        public Odunc (Uye uye, Kitap kitap)
12        {
13            this.uye = uye;
14            this.kitap = kitap;
15            zaman = DateTime.Now;
16        }
17
18        public Uye Uye {
19            get { return this.uye; }
20            set { uye = value; }
21        }
22    }
```



Odunc.cs II

```
22
23     public Kitap Kitap {
24         get { return this.kitap; }
25         set { kitap = value; }
26     }
27
28     public override string ToString ()
29     {
30         return string.Format ("[Odunc: zaman={0}, Uye={1}, Kitap={2}]",
↪      zaman, Uye.Ad, Kitap.Ad);
31     }
32
33 }
34 }
```



Kutuphane Sınıfı

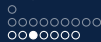
Kütüphane sınıfı üyeleri, kitapları ve ödünç almaları içinde barındıran bir sınıftır ve aşağıdaki elemanları içerir. Kurucu metodlar(constructor) ve her bir eleman için bir *get* ve *set* metodları yazılacaktır.

- uyeler: Tüm üyelerin listesini tutar
- kitaplar: Tüm kitapların listesini tutar
- oduncler: Ödünç alınan kitapların listesini tutar
- Metodlar:
 - `void uyeEkle(Uye u)`: Kütüphaneye üye ekler
 - `void uyeleriGoster()`: Üyeleri listeler
 - `Uye uyeBul (string uyeAdi)`: Üye arama
 - `void kitapEkle(Kitap k)`: Kitap ekler
 - `Kitap kitapBul (string kitapAdi)`: Kitap arama
 - `void oduncListele()`: Ödünçleri listeler
 - `void oduncAl(string uyeAdi, string kitapAdi)`
 - `void oduncAl(Uye uye, Kitap kitap)`



Kutuphane.cs I

```
1 using System;
2 using System.Collections;
3
4 namespace VYDers1
5 {
6     public class Kutuphane
7     {
8         ArrayList uyeler;
9         ArrayList kitaplar;
10        ArrayList oduncler;
11        public Kutuphane ()
12        {
13            uyeler = new ArrayList ();
14            kitaplar = new ArrayList ();
15            oduncler = new ArrayList ();
16        }
17        public void uyeEkle (Uye u)
18        {
19            uyeler.Add (u);
20        }
21    }
```



Kutuphane.cs II

```
21 public void uyeleriGoster ()
22 {
23     Console.WriteLine ("Kütüphane Üyeleri:");
24     foreach (Uye u in uyeler)
25         Console.WriteLine (u);
26 }
27 public Uye uyeBul (string uyeAdi)
28 {
29     foreach (Uye u in uyeler)
30         if (u.Ad.Equals (uyeAdi))
31             return u;
32     return null;
33 }
34 public void kitapEkle(Kitap k)
35 {
36     kitaplar.Add (k);
37 }
```



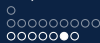
Kutuphane.cs III

```
38     public Kitap kitapBul (string kitapAdi)
39     {
40         foreach (Kitap k in kitaplar)
41             if (k.Ad.Equals (kitapAdi))
42                 return k;
43         return null;
44     }
45     public void oduncListele()
46     {
47         foreach (Odunc o in oduncler)
48             Console.WriteLine (o);
49     }
```



Kutuphane.cs IV

```
50 public void oduncAl(string uyeAdi, string kitapAdi)
51 {
52     Uye u = uyeBul (uyeAdi);
53     Kitap k = kitapBul (kitapAdi);
54     if (u == null || k==null)
55         throw new Exception ("Üye veya kitap bulunamadı!");
56     oduncler.Add (new Odunc (u, k));
57 }
58 public void oduncAl(Uye uye, Kitap kitap)
59 {
60     oduncler.Add (new Odunc (uye, kitap));
61 }
62 }
63 }
```



Program.cs I

```
1 using System;
2
3 namespace VYders1
4 {
5     class MainClass
6     {
7         public static void Main (string[] args)
8         {
9             Kitap k1 = new Kitap ("978-0132576277",
↪ "Data Structures and Algorithm Analysis in Java", "Mark Allen Weiss",
↪ "Bilgisayar Bilimleri");
10             Kitap k2 = new Kitap ("978-6054787012",
↪ "C && Java ile Veri Yapılarına Giriş", "Olca Taner Yıldız",
↪ "Bilgisayar");
11             Kitap k3 = new Kitap ("1234", "Çalılıkusu", "Reşat Nuri Güntekin",
↪ "Roman");
12             Console.WriteLine (k1.ToString());
13             Console.WriteLine (k2+"\n"+k3);
14             Uye u1 = new Uye (1, "Şevket Umut Çakır", "Denizli");
15             Uye u2 = new Uye (2, "Selahattin Akkaş", "Manisa");
16             Uye u3 = new Uye (3, "Mustafa Tosun", "Antalya");
```




Program.cs II

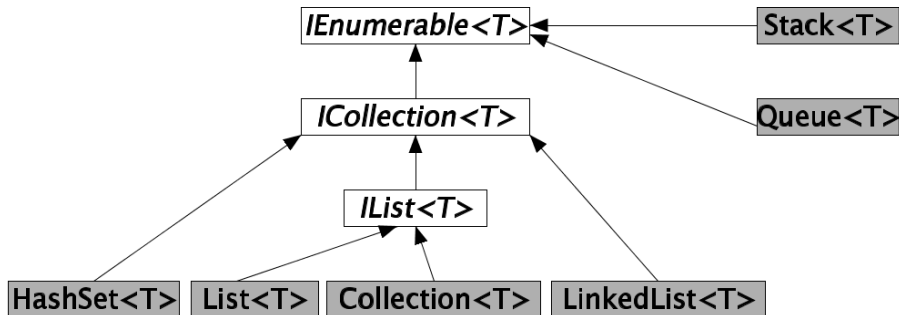
```
17         Console.WriteLine (u1);
18         Console.WriteLine (u2+"\n"+u3);
19         Kutuphane kutuphane = new Kutuphane ();
20         kutuphane.kitapEkle (k1);
21         kutuphane.kitapEkle (k2);
22         kutuphane.kitapEkle (k3);
23         kutuphane.uyeEkle (u1);
24         kutuphane.uyeEkle (u2);
25         kutuphane.uyeEkle (u3);
26         kutuphane.uyeleriGoster ();
27         kutuphane.oduncAl ("Şevket Umut Çakır",
↪      "Data Structures and Algorithm Analysis in Java");
28         kutuphane.oduncAl (u3, k2);
29         kutuphane.oduncListele ();
30     }
31 }
32 }
```

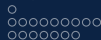
Generic Tanım

Generic tasarlanan interface(arayüz), class(sınıf), metod yada parametrelerin(argümanların) belirli bir tip için değil bir şablon yapısına uyan her tip için çalışmasını sağlayan bir yapıdır.

- Tekrar kullanılabilir kod yazmayı kolaylaştırırlar.
- Çalışma zamanında (run time) gereksiz Cast ve Boxing-Unboxing kullanmasını önlediğinden efektif performans sağlar.
- Derleme zamanında (compile time) (type safe) tip güvenli değişken kullanılmasını zorlayarak çalışma zamanında oluşabilecek tip dönüşüm hatalarını önler.
- Programcıya kod üzerinde daha güçlü esnek bir kontrol sağlar.

Generic Koleksiyonlar





Generic Koleksiyon Kullanımı

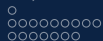
`using System.Collections.Generic;` kullanılarak eklenebilir.

```
1 List<int> myList = new List<int>();
2 myList.Add(3);
3 myList.Add(4);
4 //myList.Add(5.0); //Hatalı işlem
5 int total = 0;
6 foreach(int val in myList)
7 {
8     total = total + val;
9 }
10 Console.WriteLine("Toplam : {0}", total);
```

Collections.Generic Özellikleri

Generic koleksiyonlar üzerinde kalıtım yoluyla gelen aşağıdaki metodlar ve özellikler kullanılabilir.

- Add, Insert, Remove, RemoveAt, RemoveRange, RemoveAll
- Sort (Comparison<T>), Reverse
- ConvertAll (Converter<T,U>), CopyTo(T[])
- Exists, Find, FindAll, FindLast, FindIndex, FindLastIndex, IndexOf, LastIndexOf, TrueForAll
- ForEach (Action<T>)



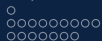
Dictionary Örneği

```
1 Dictionary<string, int> numbers = new Dictionary<string, int>();
2 numbers["Ayşe"] = 235485;
3 numbers["Ali"] = 965748;
4 numbers["Mehmet"] = 987654;
5 foreach (KeyValuePair<string, int> entry in numbers)
6     Console.WriteLine(entry.Key + ":" + entry.Value);
```



Generic Metodlar

```
1 using System;
2 using System.Collections.Generic;
3 namespace Generics
4 {
5     class MainClass
6     {
7         public static void Copy<T>(List<T> source, List<T> destination)
8         {
9             foreach (T obj in source)
10                 destination.Add (obj);
11         }
12         public static void Main (string[] args)
13         {
14             List<int> lst1 = new List<int>();
15             lst1.Add(2);
16             lst1.Add(4);
17             List<int> lst2 = new List<int>();
18             Copy(lst1, lst2);
19             Console.WriteLine(lst2.Count);
20         }
21     }
```



Generic Arayüzler

```
1 using System;
2 using System.Collections.Generic;
3 namespace Generics
4 {
5     class MainClass
6     {
7         public static T Max<T>(T op1, T op2) where T : IComparable
8         {
9             if (op1.CompareTo (op2) < 0)
10                 return op2;
11             return op1;
12         }
13         public static void Main (string[] args)
14         {
15             Console.WriteLine (Max(10,15));
16             Console.WriteLine (Max("Ali","Ayşe"));
17         }
18     }
19 }
```


Generic Arayüzler

```
1 where T : struct           //T tipi değer (value) tipleri olmalıdır.
2 where T : class           //T tipi referans (reference) tipleri olmalıdır.
3 where T : new()           //T tipi yükleyicisi (constructor) parametresiz
    ↪ olan bir tip olmalıdır.
4 where T : class_name      //T tipi oluşturduğunuz bir sınıf yada bu sınıfı
    ↪ ile genişletilmiş alt sınıflar olmalıdır.
5 where T : interface_name //T tipi belirtilen interface ile genişletilmiş bir
    ↪ obje olmalıdır.
```

