

Bölüm 4

C# Sınıf, ArrayList

Dr. Öğr. Üyesi Murat TAŞYÜREK (kayubmprogramlama1@gmail.com)

13 Mart 2024

Kayseri Üniversitesi, Bilgisayar Mühendisliği Bölümü

- Sınıflar
- List
- ArrayList
- List Foreach
- List Sort
- Hata yakalama (Exception Handle)
- Ödev

- Günler için bir sınıf tanımlayalım ve sadece gün isimlerini tutsun.
- Aylar için bir sınıf tanımlayalım.
- Aylar için tanımlanan sınıfta ayın günleri (gün sınıfından bir dizi), ayın ismi, ayın kaçınıcı ay olduğu ve hangi yıla ait olduğu bilgileri olsun.
- Ayrıca ayara gün set edilebilsin.
- Ayın günlerini göstermek için de `AyinGunleriniGoster` isminde bir metodumuz olsun.
- Ayın günleri `13.03.2024` formatında gösterebiliriz.

Zaman Sınıfı Tanımlama

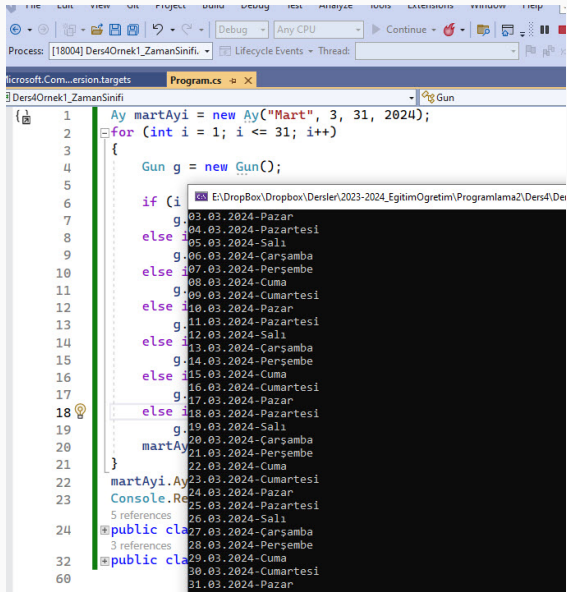
```
23 public class Gun
24 {
25     public string gunAdi;
26     public void setGunAdi(string gun)
27     {
28         this.gunAdi = gun;
29     }
30 }
31 public class Ay
32 {
33     public string ayAdi;
34     public int ayRakam;
35     public Gun[] ayinGunleri;
36     public int Yili;
37     public Ay(string ayAdi, int ayRakam, int ayinGunSayisi, int yil)
38     {
39         this.ayAdi = ayAdi;
40         this.ayRakam = ayRakam;
41         this.ayinGunleri = new Gun[ayinGunSayisi];
42         this.Yili = yil;
43     }
44     public void GunSetEt(Gun g, int index)
45     {
46         this.ayinGunleri[index] = g;
47     }
48     public void AyinGuleriniGoster()
49     {
50         Console.WriteLine(this.ayAdi + " - Günleri");
51         for (int i = 1; i <= ayinGunleri.Length; i++)
52         {
53             string GunFormati = String.Format("{0:00}", i);
54             string AyFormati = String.Format("{0:00}", this.ayRakam);
55             Console.WriteLine(GunFormati + "," + AyFormati + "," + this.Yili + "--" + ayinGunleri[i - 1].gunAdi);
56         }
57     }
58 }
```

Zaman Sınıfı Gün Ekleme

```
Microsoft.Com...ersion.targets  Program.cs  X
Ders4Ornek1_ZamanSinifi  Gun

1  Ay martAyi = new Ay("Mart", 3, 31, 2024);
2  for (int i = 1; i <= 31; i++)
3  {
4      Gun g = new Gun();
5
6      if (i % 7 == 1)
7          g.setGunAdi("Cuma");
8      else if (i % 7 == 2)
9          g.setGunAdi("Cumartesi");
10     else if (i % 7 == 3)
11         g.setGunAdi("Pazar");
12     else if (i % 7 == 4)
13         g.setGunAdi("Pazartesi");
14     else if (i % 7 == 5)
15         g.setGunAdi("Salı");
16     else if (i % 7 == 6)
17         g.setGunAdi("Çarşamba");
18     else if (i % 7 == 0)
19         g.setGunAdi("Perşembe");
20     martAyi.GunSetEt(g, i - 1);
21 }
22 martAyi.AyinGuleriniGoster();
23 Console.ReadKey();
24 public class Gun...
32 public class Ay...
```

Zaman Sınıfı Çıktıları



```
1  Ay martAyi = new Ay("Mart", 3, 31, 2024);
2  for (int i = 1; i <= 31; i++)
3  {
4      Gun g = new Gun();
5
6      if (i == 3) g = new Gun("03.03.2024-Pazar");
7      else if (i == 4) g = new Gun("04.03.2024-Pazartesi");
8      else if (i == 5) g = new Gun("05.03.2024-Salı");
9      else if (i == 6) g = new Gun("06.03.2024-Çarşamba");
10     else if (i == 7) g = new Gun("07.03.2024-Perşembe");
11     else if (i == 8) g = new Gun("08.03.2024-Cuma");
12     else if (i == 9) g = new Gun("09.03.2024-Cumartesi");
13     else if (i == 10) g = new Gun("10.03.2024-Pazar");
14     else if (i == 11) g = new Gun("11.03.2024-Pazartesi");
15     else if (i == 12) g = new Gun("12.03.2024-Salı");
16     else if (i == 13) g = new Gun("13.03.2024-Çarşamba");
17     else if (i == 14) g = new Gun("14.03.2024-Perşembe");
18     else if (i == 15) g = new Gun("15.03.2024-Cuma");
19     else if (i == 16) g = new Gun("16.03.2024-Cumartesi");
20     else if (i == 17) g = new Gun("17.03.2024-Pazar");
21     else if (i == 18) g = new Gun("18.03.2024-Pazartesi");
22     else if (i == 19) g = new Gun("19.03.2024-Salı");
23     else if (i == 20) g = new Gun("20.03.2024-Çarşamba");
24     else if (i == 21) g = new Gun("21.03.2024-Perşembe");
25     else if (i == 22) g = new Gun("22.03.2024-Cuma");
26     else if (i == 23) g = new Gun("23.03.2024-Cumartesi");
27     else if (i == 24) g = new Gun("24.03.2024-Pazar");
28     else if (i == 25) g = new Gun("25.03.2024-Pazartesi");
29     else if (i == 26) g = new Gun("26.03.2024-Salı");
30     else if (i == 27) g = new Gun("27.03.2024-Çarşamba");
31     else if (i == 28) g = new Gun("28.03.2024-Perşembe");
32     else if (i == 29) g = new Gun("29.03.2024-Cuma");
33     else if (i == 30) g = new Gun("30.03.2024-Cumartesi");
34     else if (i == 31) g = new Gun("31.03.2024-Pazar");
35 }
36 martAyi.Ay = g;
37 Console.WriteLine(martAyi.ToString());
38 }
```

- Diziler sabit uzunlukta tanımlanır.
- Yeni öge eklenerek dizilerin boyutu uzatılmaz.
- Uzatılmak istendiğinde yeni dizi tanımlamak gerekir.
- Dinamik uzunlukta dizi ihtiyacı olduğu zaman genellikle **List** veya **ArrayList** kullanılır.

List ve ArrayList

- C# dilinde, **List** ve **ArrayList** sınıfları, koleksiyonlar içinde veri saklamak ve işlemek için kullanılan veri yapılarıdır.
- List aynı veri tipindeki elemanları içermektedir. Örneğin int tipinde oluşturduğumuz bir dizide sadece tam sayı tutabiliriz.
- Ancak ArrayList'de böyle bir kısıtlama bulunmamaktadır. Farklı veri tipindeki öğeleri ArrayList'e ekleyebiliriz.
- **List**, dizin tarafından erişilebilen ve listeyi sıralama, arama ve değiştirme yöntemlerine sahip, türü kesin olarak belirlenmiş nesnelerin bir koleksiyonudur. **System.Collections.Generic** ad alanı altında gelen **ArrayList**'in genel sürümüdür.

List vs ArrayList

- Listeler sadece tek bir türde nesne içerdiğinden dolayı ArrayList'e göre daha hızlı çalışır.
- Her iki listenin de kullanımı aynıdır.

```
1 using System.Collections.Generic;  
  
1 // List with default capacity  
2 List<Int16> list = new List<Int16>();  
3 // List with capacity = 5  
4 List<string> authors = new List<string>(5);  
5 string[] animals = { "Cow", "Camel", "Elephant" };  
6 List<string> animalsList = new List<string>(animals);
```

List Fonksiyonlar

- **Add:** koleksiyona yeni bir öğe ekler.
- **Remove:** belirtilen öğeyi koleksiyondan kaldırır.
- **Contains:** belirli bir öğenin koleksiyonda olup olmadığını kontrol eder.
- **Count:** koleksiyonun içindeki öğe sayısını alır.
- **Clear:** koleksiyondaki tüm öğeleri temizler.
- **Indexer:** indeksleme kullanılarak öğelere erişim sağlar.
- **Sort:** koleksiyonu belirli bir sıraya göre sıralar.

Örnek List Kodlayalım

```
Program.cs
Ders4CoreOrnekList2

1  List<string> liste = new List<string>();
2  liste.Add("1");
3  liste.Add("3");
4  liste.Insert(1, "2");
5  Console.WriteLine("Sadece eklenmiş hali");
6  foreach (string s in liste)
7  {
8      Console.WriteLine(s);
9  }
10
11  liste.Add("11");
12  liste.Add("7");
13  liste.Add("5");
14  liste.Remove("3");
15  Console.WriteLine("Silinmiş yeni hali");
16  foreach (string s in liste)
17  {
18      Console.WriteLine(s);
19  }
20
21  liste.Sort();
22  Console.WriteLine("Sıralı hali (string olarak eklendiğinden metinsel sıralar)");
23  foreach (string s in liste)
24  {
25      Console.WriteLine(s);
26  }
27  Console.ReadKey();
28
```

List Çıktı

```
ogram.cs -p x
Ders4CoreOrnekList2
{
1   List<string> liste = new List<string>();
2   liste.Add("1");
3   liste.Add("3");
4   liste.Insert(1, "2");
5   E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders4\Ders4NetCore\I
6   Sadece eklenmiş hali
7   1
8   2
9   3
10  Silinmiş yeni hali
11  1
12  2
13  11
14  7
15  5
16  Sıralı hali (string olarak eklendiğinden metinsel sıralar)
17  1
18  11
19  2
20  5
21  7
22
```

- List kullanarak, Günler için bir sınıf tanımlayalım ve sadece gün isimlerini tutsun.
- Aylar için bir sınıf tanımlayalım.
- Aylar için tanımlanan sınıfta ayın günleri (gün sınıfından bir List), ayın ismi, ayın kaçınıcı ay olduğu ve hangi yıla ait olduğu bilgileri olsun.
- Ayrıca ayara gün set edilebilsin.
- Ayın günlerini göstermek için de AyinGunleriniGoster isminde bir metodumuz olsun.
- Günleri hangi sırada eklersek ekleyelim sıralı gösterebilirsin.
- Ayın günleri 29.02.2024 formatında gösterebilirsin.

```
6 references
public class Gun
{
    public string gunAdi;
    public int ayinKacinciGunu;
    7 references
    public void setGunAdi(string gun, int kacinciGun)
    {
        this.gunAdi = gun;
        this.ayinKacinciGunu = kacinciGun;
    }
}
```

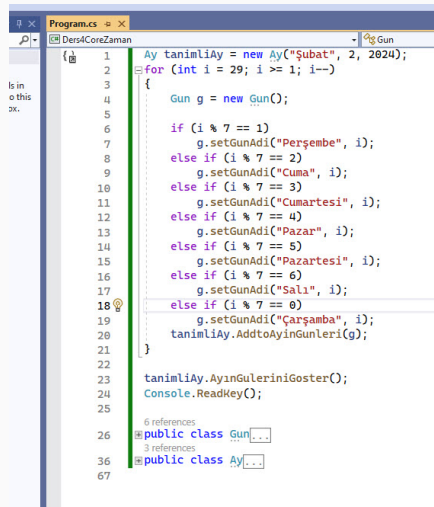
Ay Sınıfı

```
3 references
public class Ay
{
    public string ayAdi;
    public int ayRakam;
    public List<Gun> ayinGunleri;
    public int Yili;

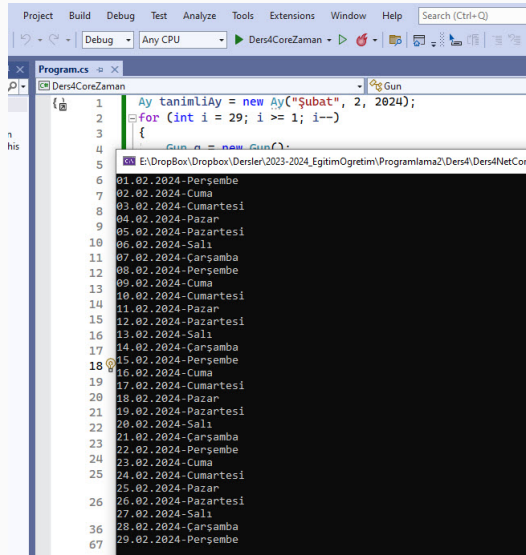
    1 reference
    public Ay(string ayAdi, int ayRakam, int yil)
    {
        this.ayAdi = ayAdi;
        this.ayRakam = ayRakam;
        this.ayinGunleri = new List<Gun>();
        this.Yili = yil;
    }

    1 reference
    public void AddtoAyinGunleri(Gun g)
    {
        this.ayinGunleri.Add(g);
    }

    1 reference
    public void AyinGuleriniGoster()
    {
        Console.WriteLine(this.ayAdi + " - Günleri");
        this.ayinGunleri = ayinGunleri.OrderBy(g => g.ayinKacinciGunu).ToList();
        foreach (Gun g in ayinGunleri)
        {
            string GunFormati = String.Format("{0:00}", g.ayinKacinciGunu);
            string AyFormati = String.Format("{0:00}", this.ayRakam);
            Console.WriteLine(GunFormati + "." + AyFormati + "." + this.Yili + "-" + g.gunAdi);
        }
    }
}
```



```
1  Ay tanimliAy = new Ay("Şubat", 2, 2024);
2  for (int i = 29; i >= 1; i--)
3  {
4      Gun g = new Gun();
5
6      if (i % 7 == 1)
7          g.setGunAdi("Perşembe", i);
8      else if (i % 7 == 2)
9          g.setGunAdi("Cuma", i);
10     else if (i % 7 == 3)
11         g.setGunAdi("Cumartesi", i);
12     else if (i % 7 == 4)
13         g.setGunAdi("Pazar", i);
14     else if (i % 7 == 5)
15         g.setGunAdi("Pazartesi", i);
16     else if (i % 7 == 6)
17         g.setGunAdi("Salı", i);
18     else if (i % 7 == 0)
19         g.setGunAdi("Çarşamba", i);
20     tanimliAy.AddtoAyinGunleri(g);
21 }
22
23 tanimliAy.AyinGuleriniGoster();
24 Console.ReadKey();
25
26 public class Gun...
27
28 public class Ay...
```

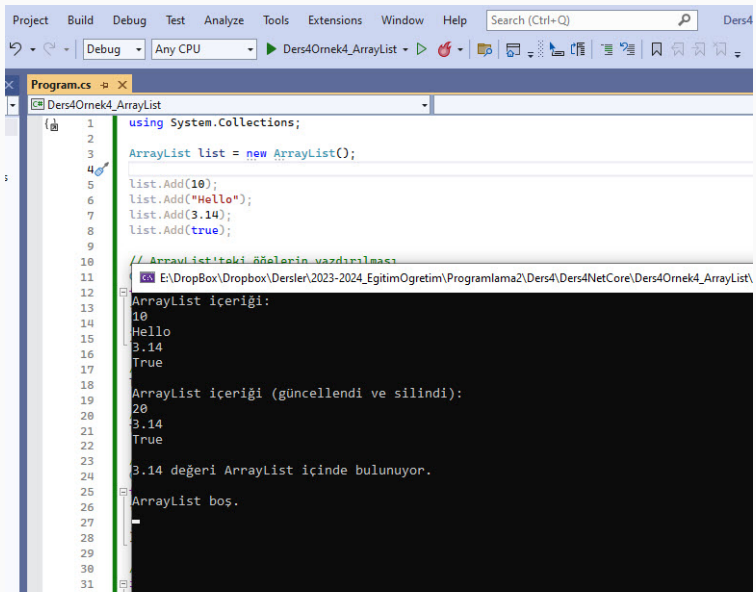
The screenshot shows a C# program in Visual Studio. The code defines a class `Ay` and a `for` loop that iterates from 29 down to 1, creating `Ay` objects for each day of the month of Şubat 2024. The output window displays the resulting dates and days of the week.

```
1 Ay tanimliAy = new Ay("Şubat", 2, 2024);
2 for (int i = 29; i >= 1; i--)
3 {
4     Gun g = new Gun();
5     Console.WriteLine(g.ToString());
6     01.02.2024 - Perşembe
7     02.02.2024 - Cuma
8     03.02.2024 - Cumartesi
9     04.02.2024 - Pazar
10    05.02.2024 - Pazartesi
11    06.02.2024 - Salı
12    07.02.2024 - Çarşamba
13    08.02.2024 - Perşembe
14    09.02.2024 - Cuma
15    10.02.2024 - Cumartesi
16    11.02.2024 - Pazar
17    12.02.2024 - Pazartesi
18    13.02.2024 - Salı
19    14.02.2024 - Çarşamba
20    15.02.2024 - Perşembe
21    16.02.2024 - Cuma
22    17.02.2024 - Cumartesi
23    18.02.2024 - Pazar
24    19.02.2024 - Pazartesi
25    20.02.2024 - Salı
26    21.02.2024 - Çarşamba
27    22.02.2024 - Perşembe
28    23.02.2024 - Cuma
29    24.02.2024 - Cumartesi
30    25.02.2024 - Pazar
31    26.02.2024 - Pazartesi
32    27.02.2024 - Salı
33    28.02.2024 - Çarşamba
34    29.02.2024 - Perşembe
```

- **ArrayList**, farklı türlerden öğeleri saklamak için kullanılabilir, ancak her öğe eklenirken veya çıkarılırken uygun türe dönüştürme yapılması gerekebilir.
- **ArrayList** sınıfı, indeksleme kullanarak öğelere erişim sağlar ve koleksiyon üzerinde çeşitli işlemler yapılmasına izin verir.
- Ancak, **List** sınıfı gibi tip güvenliği ve performans avantajlarına sahip değildir, bu nedenle genellikle tercih edilmez.
- Aşağığı örnek ile **ArrayList** kullanımını inceleyelim.

ArrayList Kodu

```
Program.cs
Ders4Omek4_ArrayList
1 using System.Collections;
2
3 ArrayList list = new ArrayList();
4
5 list.Add(10);
6 list.Add("Hello");
7 list.Add(3.14);
8 list.Add(true);
9
10 // ArrayList'teki öğelerin yazdırılması
11 Console.WriteLine("ArrayList içeriği:");
12 foreach (var item in list)
13 {
14     Console.WriteLine(item);
15 }
16
17 // ArrayList'ten öğe silme
18 list.Remove("Hello");
19
20 // ArrayList'teki öğelerin güncellenmesi
21 list[0] = 20;
22
23 // ArrayList'teki öğelerin tekrar yazdırılması
24 Console.WriteLine("\nArrayList içeriği (güncellendi ve silindi):");
25 foreach (var item in list)
26 {
27     Console.WriteLine(item);
28 }
29
30 // ArrayList içinde belirli bir öğenin varlığının kontrol edilmesi
31 if (list.Contains(3.14))
32 {
33     Console.WriteLine("\n3.14 değeri ArrayList içinde bulunuyor.");
34 }
35 else
36 {
37     Console.WriteLine("\n3.14 değeri ArrayList içinde bulunmuyor.");
38 }
39
40 // ArrayList'in boşaltılması
41 list.Clear();
42
43 if (list.Count == 0)
44 {
45     Console.WriteLine("\nArrayList boş.");
46 }
47 else
48 {
49     Console.WriteLine("\nArrayList dolu.");
50 }
51 Console.ReadKey();
```



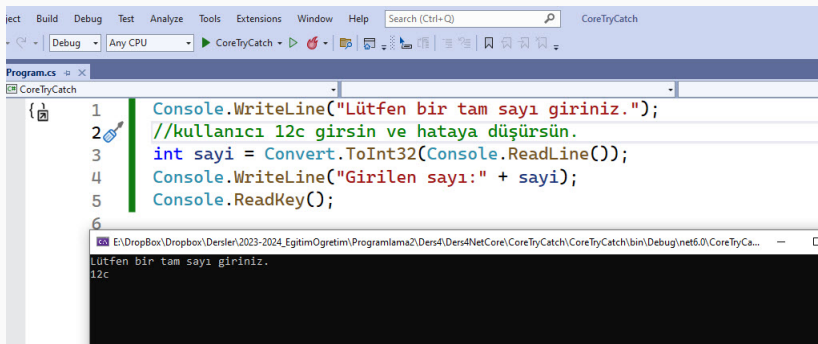
```
Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Ders4
Debug Any CPU Ders4Ornek4_ArrayList
Program.cs
Ders4Ornek4_ArrayList
1 using System.Collections;
2
3 ArrayList list = new ArrayList();
4
5 list.Add(10);
6 list.Add("Hello");
7 list.Add(3.14);
8 list.Add(true);
9
10 // ArrayList'teki öğelerin yazdırılması
11 E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders4\Ders4NetCore\Ders4Ornek4_ArrayList\
12 ArrayList içeriği:
13 10
14 Hello
15 3.14
16 True
17
18 ArrayList içeriği (güncellendi ve silindi):
19 20
20 3.14
21 True
22
23 3.14 değeri ArrayList içinde bulunuyor.
24
25 ArrayList boş.
26
27
28
29
30
31
```

- C# dilinde **try-catch** bloğu, bir kod bloğunda oluşabilecek istisnaları (hataları) yakalamak ve bu hatalarla başa çıkmak için kullanılan bir yapıdır.
- **İstisnalar**, programın normal akışını etkileyen ve beklenmeyen durumları temsil eden durumlardır.
- **try-catch** bloğu, bu tür istisnaların kontrol altına alınmasını ve uygun bir şekilde işlenmesini sağlar.

- **try bloğu:** İşlem yapılacak olan kod bloğunu içerir. Bu blokta herhangi bir hata oluşabilecek kodlar yer alır. Eğer bir hata meydana gelirse, bu hata istisna olarak fırlatılır.
- **catch bloğu:** try bloğundaki kodlar çalışırken ortaya çıkan istisnaları yakalar. **catch bloğu**, belirli bir türdeki istisnaları yakalayarak işlem yapılmasını sağlar. Birden fazla catch bloğu kullanılabilir, böylece farklı türlerdeki istisnalar için farklı işlemler yapılabilir.
- **Exception** parametresi ile hatanın kaynağını gösterebilirsiniz.

- Konsole uygulamasında kullanıcıdan tam sayı değer girmesini isteyen bir uygulama yazalım.
- Kullanıcıdan bu değeri alsın ve integer'a çevirsin.
- Kullanıcı da integer değil metin ifade girsin.
- İlgili Örneği Try-Catch bloğu olmadan ve Try-Catch bloğu ile ayrı ayrı kodlayalım.

Try-Catch bloğu olmadan kod

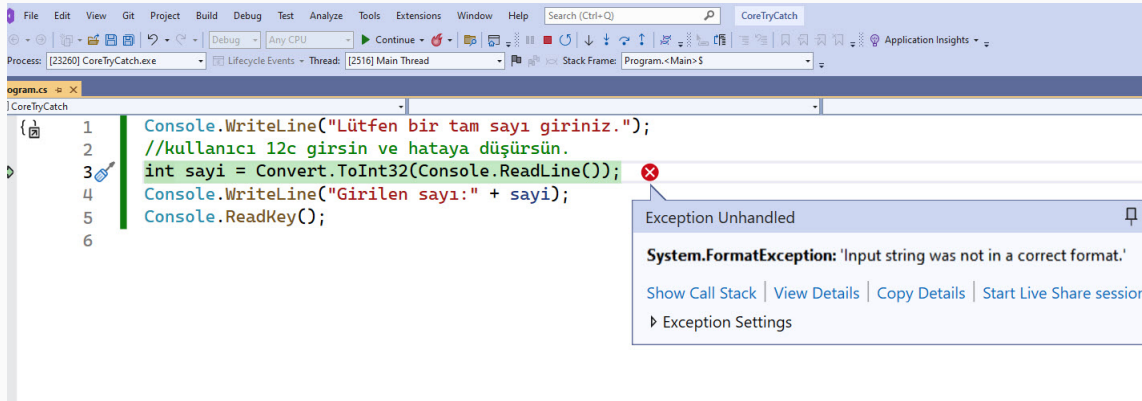


```
1 Console.WriteLine("Lütfen bir tam sayı giriniz.");
2 //kullanıcı 12c girsın ve hataya düşülsün.
3 int sayi = Convert.ToInt32(Console.ReadLine());
4 Console.WriteLine("Girilen sayı:" + sayi);
5 Console.ReadKey();
6
```

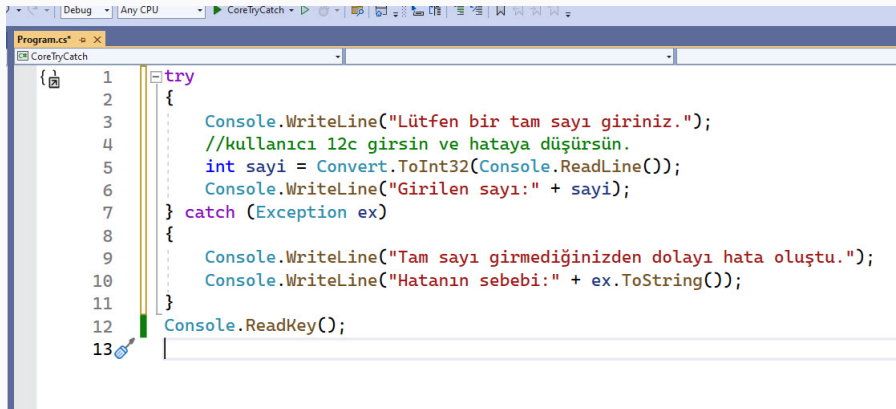
E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders4\Ders4NetCore\CoreTryCatch\CoreTryCatch\bin\Debug\net6.0\CoreTryCa...

Lütfen bir tam sayı giriniz.
12c

Try-Catch bloğu olmadan sonuç



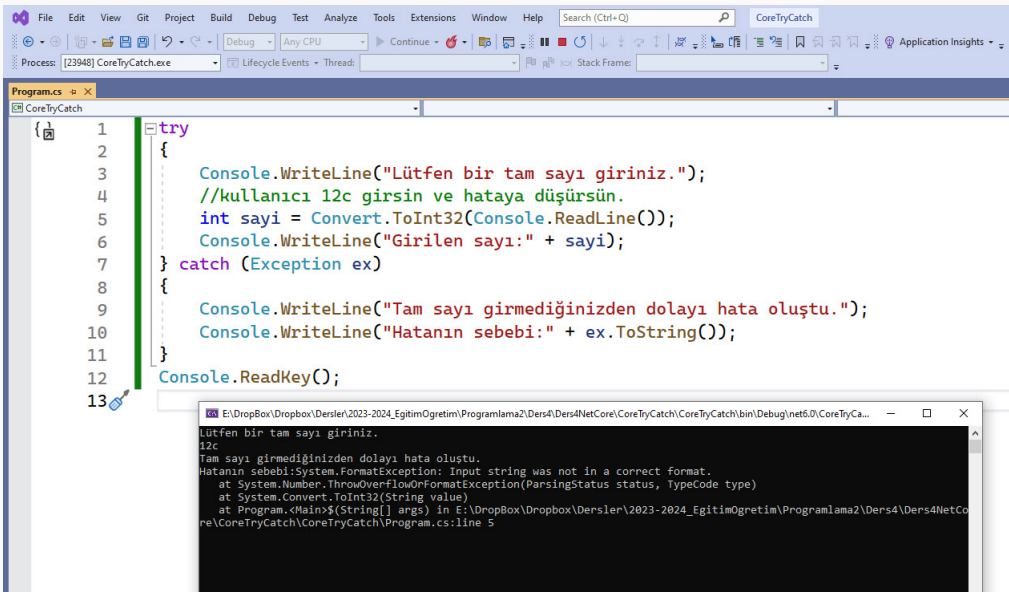
Try-Catch bloğu ile kod



The screenshot shows a Visual Studio code editor window with a C# file named Program.cs. The code is as follows:

```
1 try
2 {
3     Console.WriteLine("Lütfen bir tam sayı giriniz.");
4     //kullanıcı 12c girsin ve hataya düşürsün.
5     int sayi = Convert.ToInt32(Console.ReadLine());
6     Console.WriteLine("Girilen sayı:" + sayi);
7 } catch (Exception ex)
8 {
9     Console.WriteLine("Tam sayı girmedığınızden dolayı hata oluştu.");
10    Console.WriteLine("Hatanın sebebi:" + ex.ToString());
11 }
12 Console.ReadKey();
13
```

Try-Catch bloğu ile sonuç



The screenshot shows the Visual Studio IDE with a C# program named `CoreTryCatch` open. The code is as follows:

```
1 try
2 {
3     Console.WriteLine("Lütfen bir tam sayı giriniz.");
4     //kullanıcı 12c girsin ve hataya düşürsün.
5     int sayi = Convert.ToInt32(Console.ReadLine());
6     Console.WriteLine("Girilen sayı:" + sayi);
7 } catch (Exception ex)
8 {
9     Console.WriteLine("Tam sayı girmedığınızden dolayı hata oluştu.");
10    Console.WriteLine("Hatanın sebebi:" + ex.ToString());
11 }
12 Console.ReadKey();
13
```

The program is running in debug mode. The console output shows the following sequence of events:

```
E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders4\Ders4NetCore\CoreTryCatch\CoreTryCatch\bin\Debug\net6.0\CoreTryCa...
Lütfen bir tam sayı giriniz.
12c
Tam sayı girmedığınızden dolayı hata oluştu.
Hatanın sebebi:System.FormatException: Input string was not in a correct format.
   at System.Number.ThrowOverflowOrFormatException(ParsingStatus status, TypeCode type)
   at System.Convert.ToInt32(String value)
   at Program.<Main>$(String[] args) in E:\DropBox\Dropbox\Dersler\2023-2024_EgitimOgretim\Programlama2\Ders4\Ders4NetCo...
re\CoreTryCatch\CoreTryCatch\Program.cs:line 5
```

Scopa Oyunu

- Oyuncuların ellerindeki kartlarla masadaki kartları yakalamaya veya süpürmeye çalıştıkları bir oyun türüdür.
- Bu oyunun temel amacı, masadaki açık kartları ve eldeki kartları kullanarak 15 puanlık kombinasyonlar oluşturmak ve kazanılan kartlarla en yüksek puanları elde etmektir. Bunu yaparken her zaman özellikle dirhem kartları ve yedililer almaya özen gösterilir. Çünkü Escoba'da en fazla puanı bu kartlar kazandırmaktadır.
- Oynamak için 40 kartlı İspanyol destesi kullanılır.
- İki, üç, dört veya altı oyuncu ile oynanılır. Dört veya altı oyuncu olması durumunda oyuncular iki gruba ayrılır.
- <https://torofun.com/tr/escoba/kurallar> buradan da bilgi alabilirsiniz.

- Özgün ödev olmalı.
- **Scopa**, geleneksel bir İtalyan kart oyunudur ve genellikle bir 40 kartlık İtalyan destesi kullanılarak oynanır. Oyun iki veya daha fazla oyuncu arasında oynanabilir, ancak en yaygın olanı iki kişilik versiyonudur
- İki kişinin oynadığı oyununu kodlayınız. Oyunda adınız bir kişi soyadınız diğer kişi olsun.
- Oyuna başlamadan önce oyuncunun **doğum günü kadar** kartlarda karma işlemi yapın.
- Oyunun sonunda sırasıyla hangi oyuncunun **hangi kartları çektiğini ve kimin kazandığını** gösterin.
- Numarası tek ile bitenler **List**, çift ile bitenler **ArrayList** kullansın.
- 20 Mart 07.59'a kadar gönderilenler 8+2 puan (kayubmprogramlama1@gmail.com). Bu tarihten sonra vize tarihine kadar gönderenler 8 puan.