

VERİ YAPILARILARI VE ALGORİTMALAR

Generic Programming

Generic

- Tipden bağımsız metotlar, yapılar, sınıflar ve interface yapıları oluşturmak için kullanılabilir.



- Tip güvenlidir.
- Kod şişkinliğinin (**code bloating**) önüne geçer.

Generic



- Intermediate Language (**IL**) ile tümleşik çalışır.
- Kutulama (**boxing**) ve kutundan çıkarma (**unboxing**) işlemlerine göre daha performanslıdır.
- Generic ifadeler Just-In-Time (**JIT**) compiler ile derlenirler (**binary code reuse**).

Generic

A screenshot of the Visual Studio IDE showing a C# program. The code is in a file named 'Program.cs' and is part of a project named 'AlgoritmaTasarimi.Temeller'. The code defines a class 'Program' with a static method 'Main'. Inside 'Main', an 'ArrayList' is created and the value 44 is added. Then, the first element is cast to an 'int' and printed. Comments explain the boxing and unboxing process. The status bar at the bottom indicates 'No issues found' and the current cursor position is 'Ln: 11 Ch: 9'.

Generic

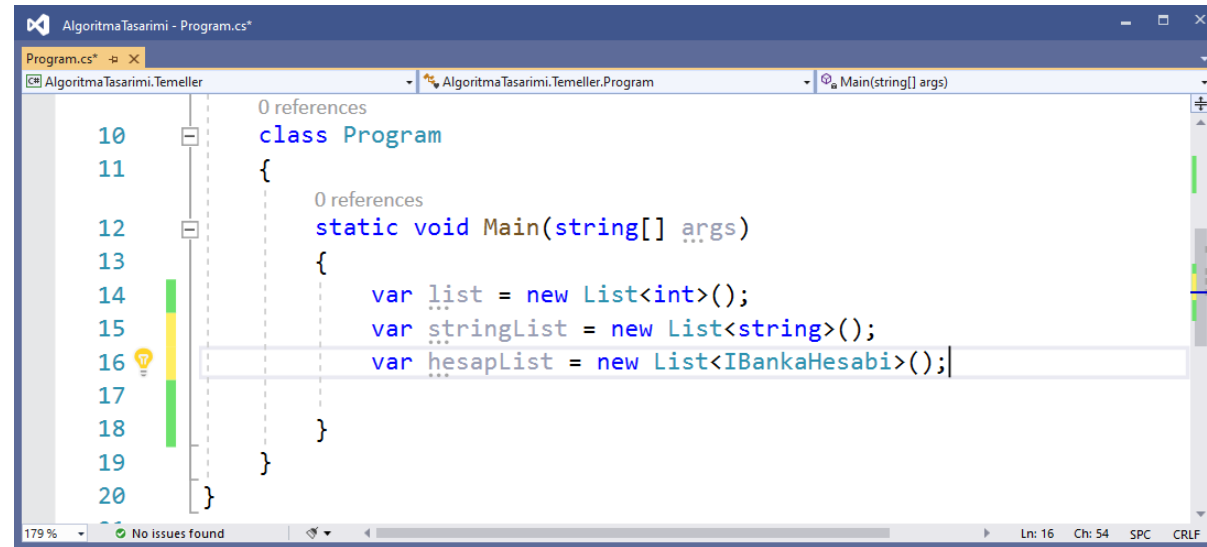


```
AlgoritmaTasarimi - Program.cs
Program.cs
AlgoritmaTasarimi.Temeller
AlgoritmaTasarimi.Temeller.Program
Main(string[] args)

0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        var list = new List<int>();
        list.Add(44); // no boxing
        int i1 = list[0];
        foreach (int i in list)
        {
            Console.WriteLine(i);
        }
    }
}
```

179 % No issues found Ln: 19 Ch: 38 SPC CRLF

Generic



The screenshot shows a Visual Studio window with the file 'Program.cs' open. The code defines a class 'Program' with a static 'Main' method. Inside 'Main', three lists are created: 'list' of type 'List<int>', 'stringList' of type 'List<string>', and 'hesapList' of type 'List<IBankaHesabi>'. The 'hesapList' line is highlighted with a light blue selection bar. The interface includes a Solution Explorer on the left, a breadcrumb trail at the top, and a status bar at the bottom indicating 'No issues found'.

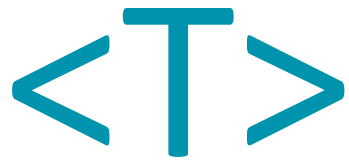
```
10 0 references
11 class Program
12 {
13     0 references
14     static void Main(string[] args)
15     {
16         var list = new List<int>();
17         var stringList = new List<string>();
18         var hesapList = new List<IBankaHesabi>();
19     }
20 }
```

Generic



- Varsayılan değerler (default values).
- Kısıtlayıcılar (constraints)
- Kalıtım (inheritance)
- Statik üyeler (static members)

Generic



Constraint

Tanımı

where T: struct

T değer tipli olmalı.

where T: class

T referans tipli olmalı.

where T: IFoo

T IFoo interface implemente etmeli.

where T: Foo

T Foo temel sınıfından türetilmeli.

where T: new()

T default bir constructor sahip olmalı.

where T1: T2

T1, T2 tipinden türetilmeli.

Generic



- Farklı senaryolar için farklı interface yapıları kullanılır:
 - `IEnumerable<out T>`
 - `IEnumerator<out T>`
 - `IComparable<T>`
 - `ICollection<T>`

Generic

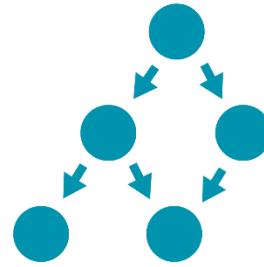


- `System.Collections`
- `System.Collections.Generic`
- `System.Collections.Specialized`
- `System.Collections.Concurrent`
- `System.Collections.Immutable`

Generic

- Dizilerin yerine koleksiyon sınıfları kullanılabilir.
- Bunlar
 - `List<T>`
 - `Queue<T>`
 - `Stack<T>`
 - `LinkedList<T>`
 - `Dictionary<TKey, TValue>`
 - `HashSet<T>`
 - `SortedSet<T>`





Veri Yapıları ve Algoritmalar

ZAFER CÖMERT

Öğretim Üyesi