

VERİ YAPILARILARI VE ALGORİTMALAR

Generic Programming

Generic

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



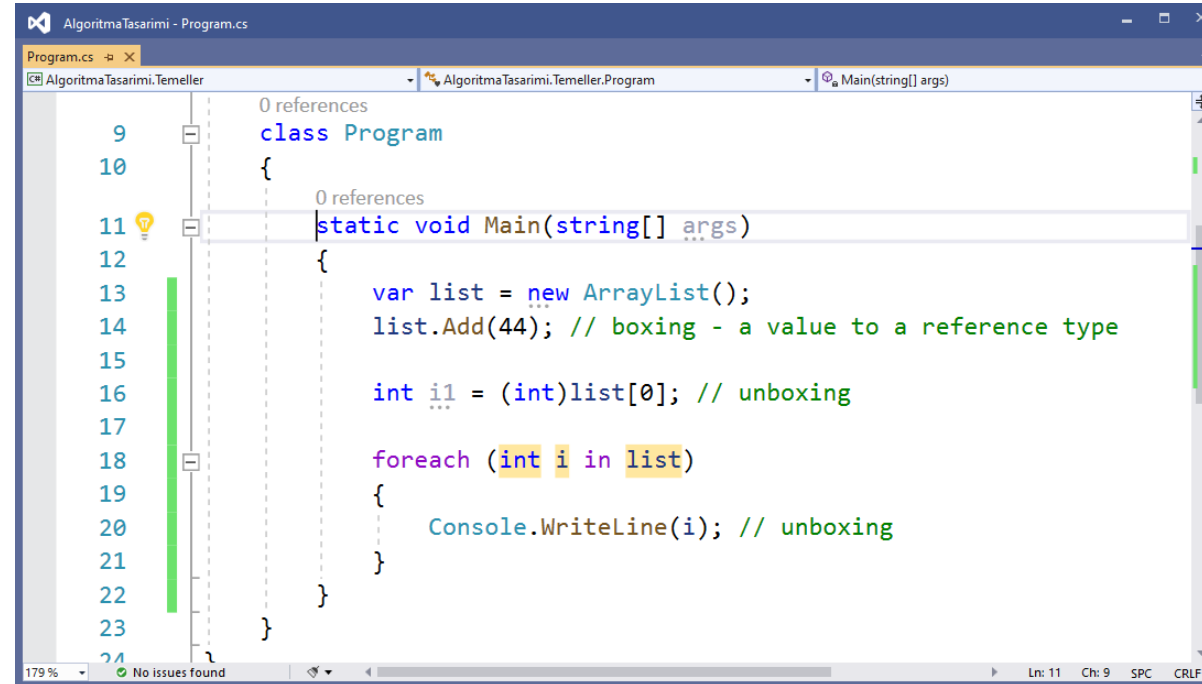
- Tip güvenlidir.
- Kod şişkinliğinin (**code bloating**) önüne geçer.

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 code editor showing a C# program. The file is named 'Program.cs' and is part of a project 'AlgoritmaTasarimi'. The code defines a class 'Program' with a static method 'Main'. The 'Main' method creates an 'ArrayList', adds the value 44, and then iterates over the list, printing each element. Comments indicate 'boxing' for adding a value to a reference type and 'unboxing' for casting the value back to an integer. The status bar at the bottom shows '179 %', 'No issues found', and 'Ln: 11 Ch: 9 SPC CRLF'.

```
9  class Program
10 {
11     static void Main(string[] args)
12     {
13         var list = new ArrayList();
14         list.Add(44); // boxing - a value to a reference type
15
16         int i1 = (int)list[0]; // unboxing
17
18         foreach (int i in list)
19         {
20             Console.WriteLine(i); // unboxing
21         }
22     }
23 }
```

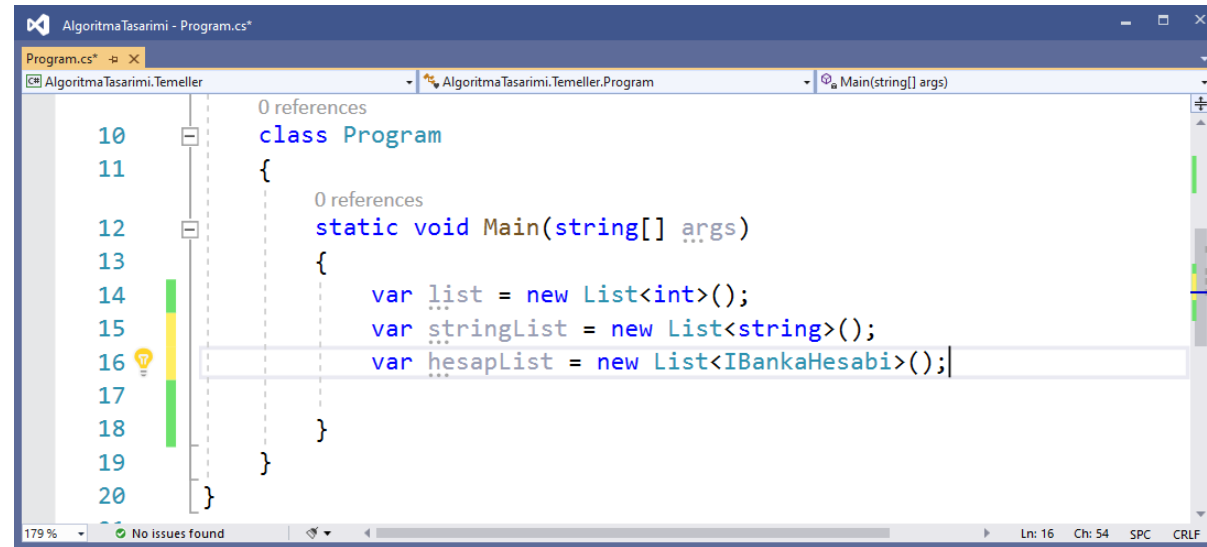
Generic

A screenshot of the Visual Studio IDE showing a C# program. The file explorer on the left shows 'Program.cs' and 'AlgorithmTasarimi.Temeller'. The code editor displays the following code:

```
10 class Program
11 {
12     static void Main(string[] args)
13     {
14         var list = new List<int>();
15         list.Add(44); // no boxing
16         int i1 = list[0];
17         foreach (int i in list)
18         {
19             Console.WriteLine(i);
20         }
21     }
22 }
23
24
```

The status bar at the bottom indicates '179 %', 'No issues found', and '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 'int', 'stringList' of type 'string', and 'hesapList' of type '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 class Program
11 {
12     static void Main(string[] args)
13     {
14         var list = new List<int>();
15         var stringList = new List<string>();
16         var hesapList = new List<IBankaHesabi>();
17     }
18 }
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

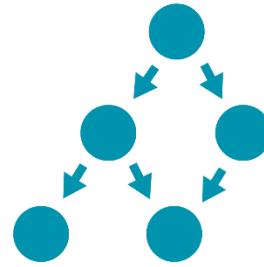


- `System.Collections`
- `System.Collections.Generic`
- `System.Collections.Specialized`
- `System.Collections.Concurrent`
- `System.Collections.Immutable`

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