



BTK
AKADEMİ

Programlama - II

Doç. Dr. Zafer CÖMERT



Bölüm 13

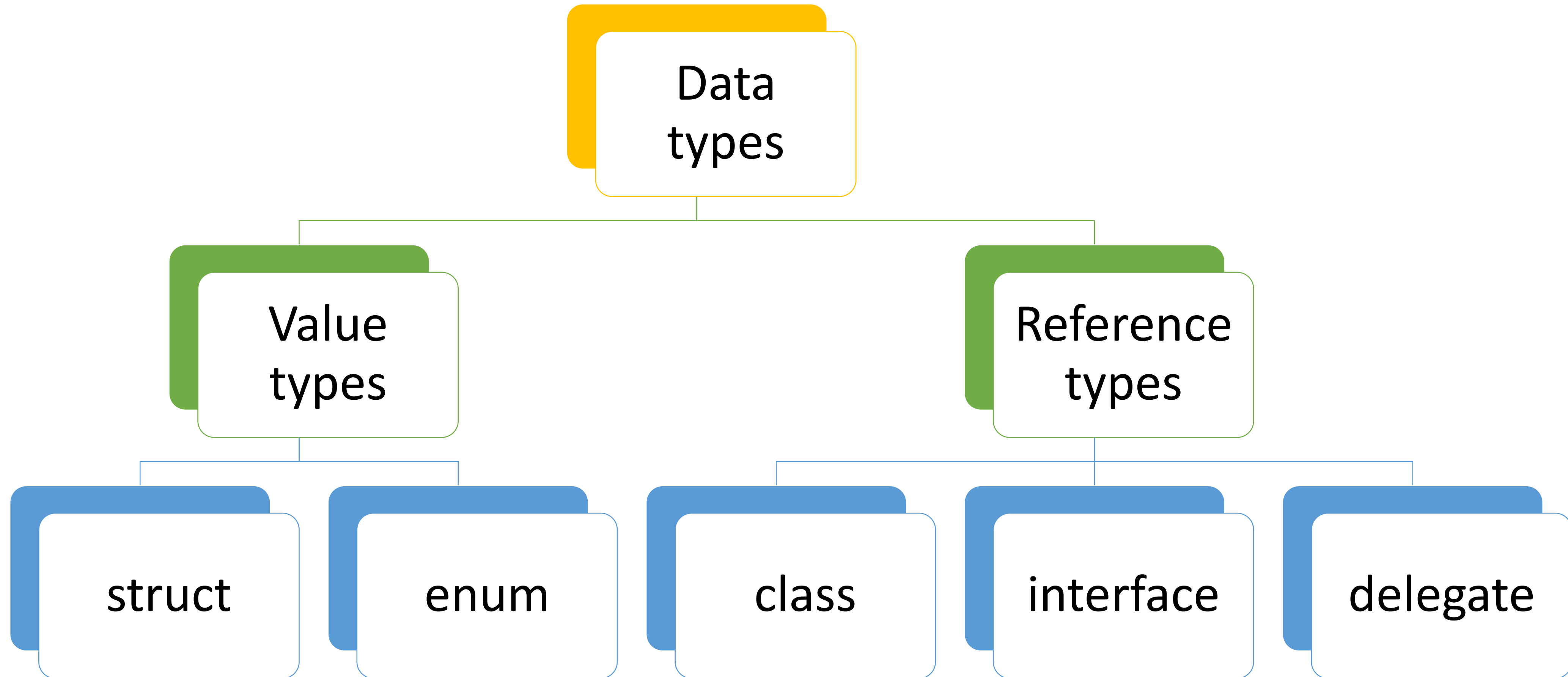
Koleksiyonlar (1)

Giriş

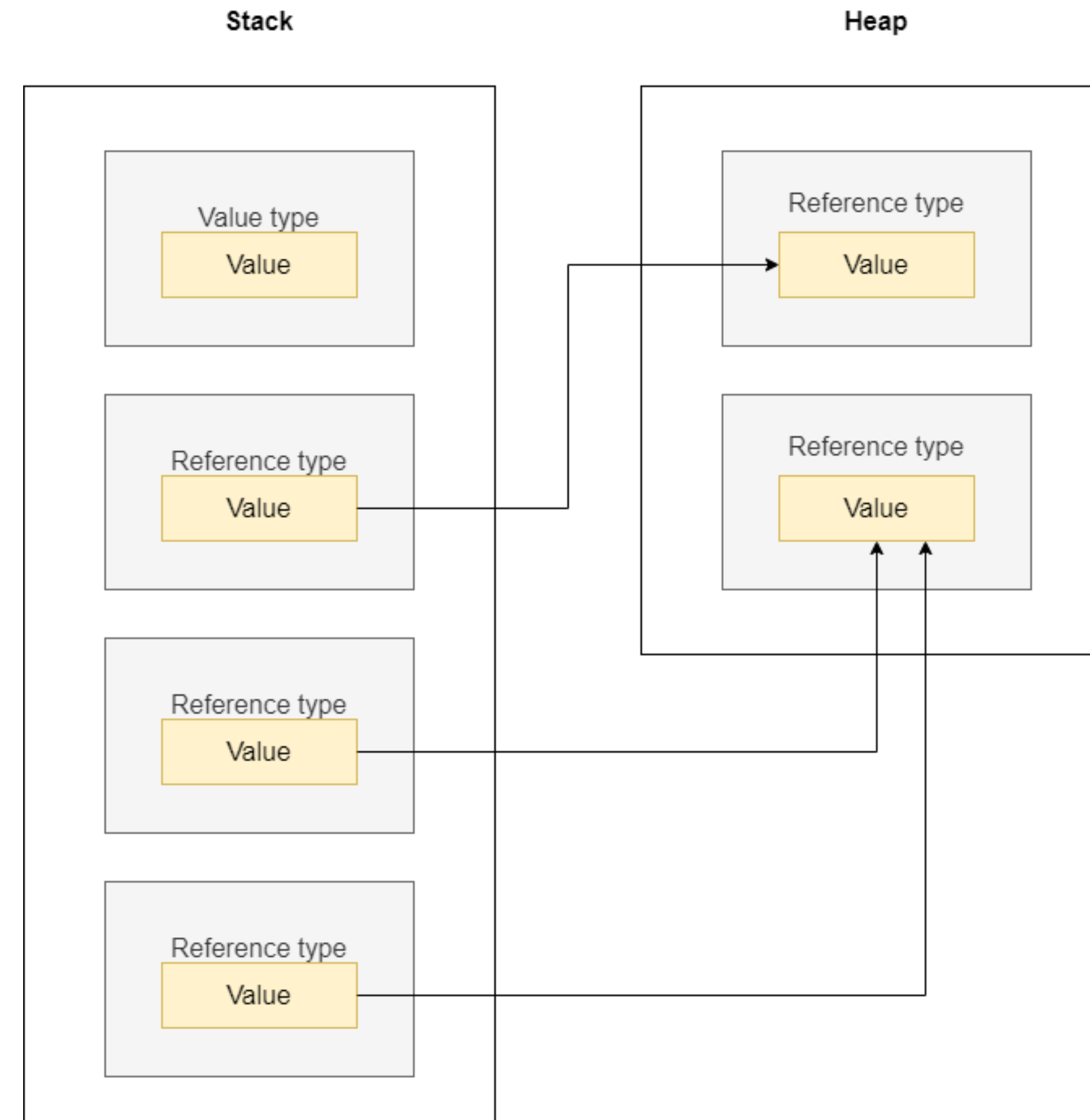
İçerik

- ArrayList
- List<T>
- Generic
- LinkedList<T>
- Queue ve Queue<T>
- Stack ve Stack<T>
- Uygulamalar

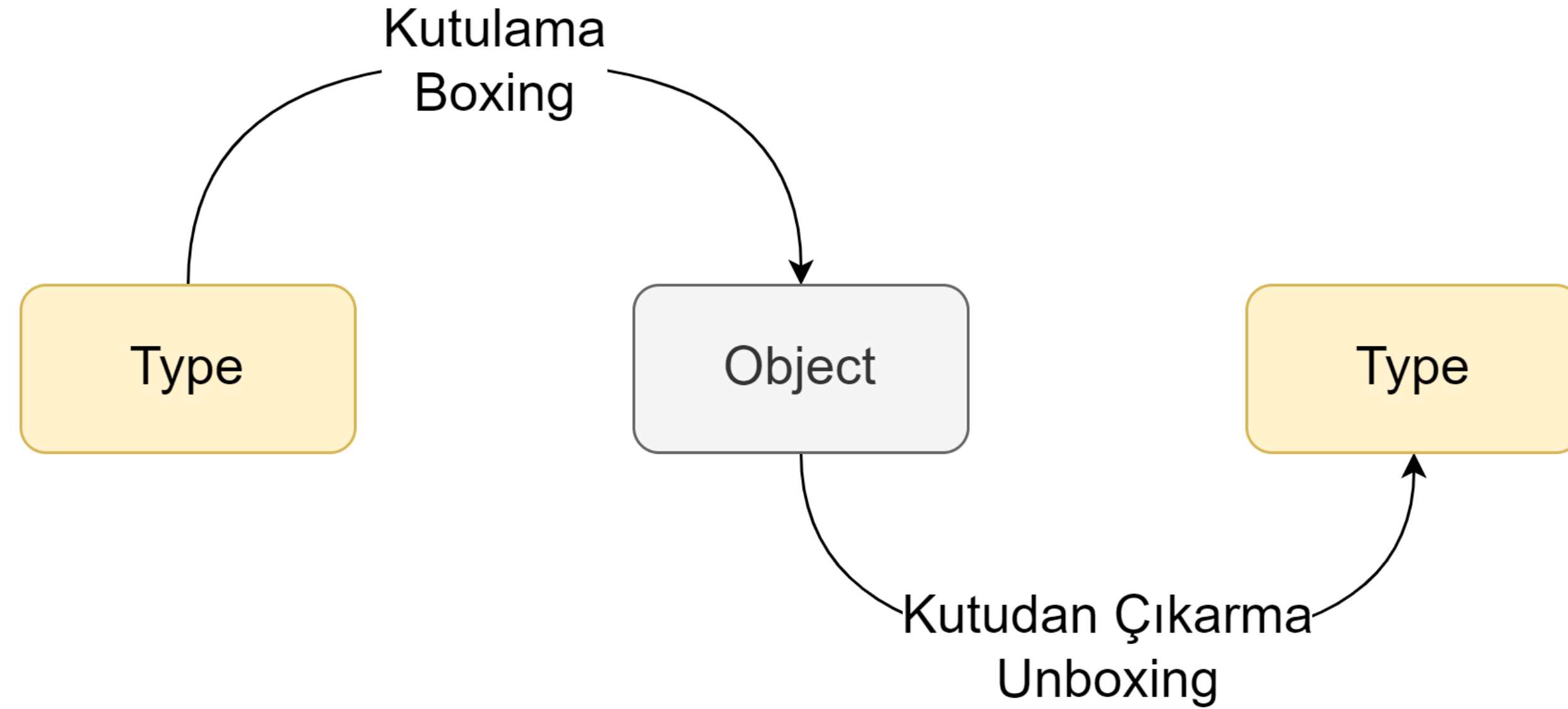
Veri Türleri



Değer ve Referans Tipler



Kutulama ve Kutudan Çıkarma

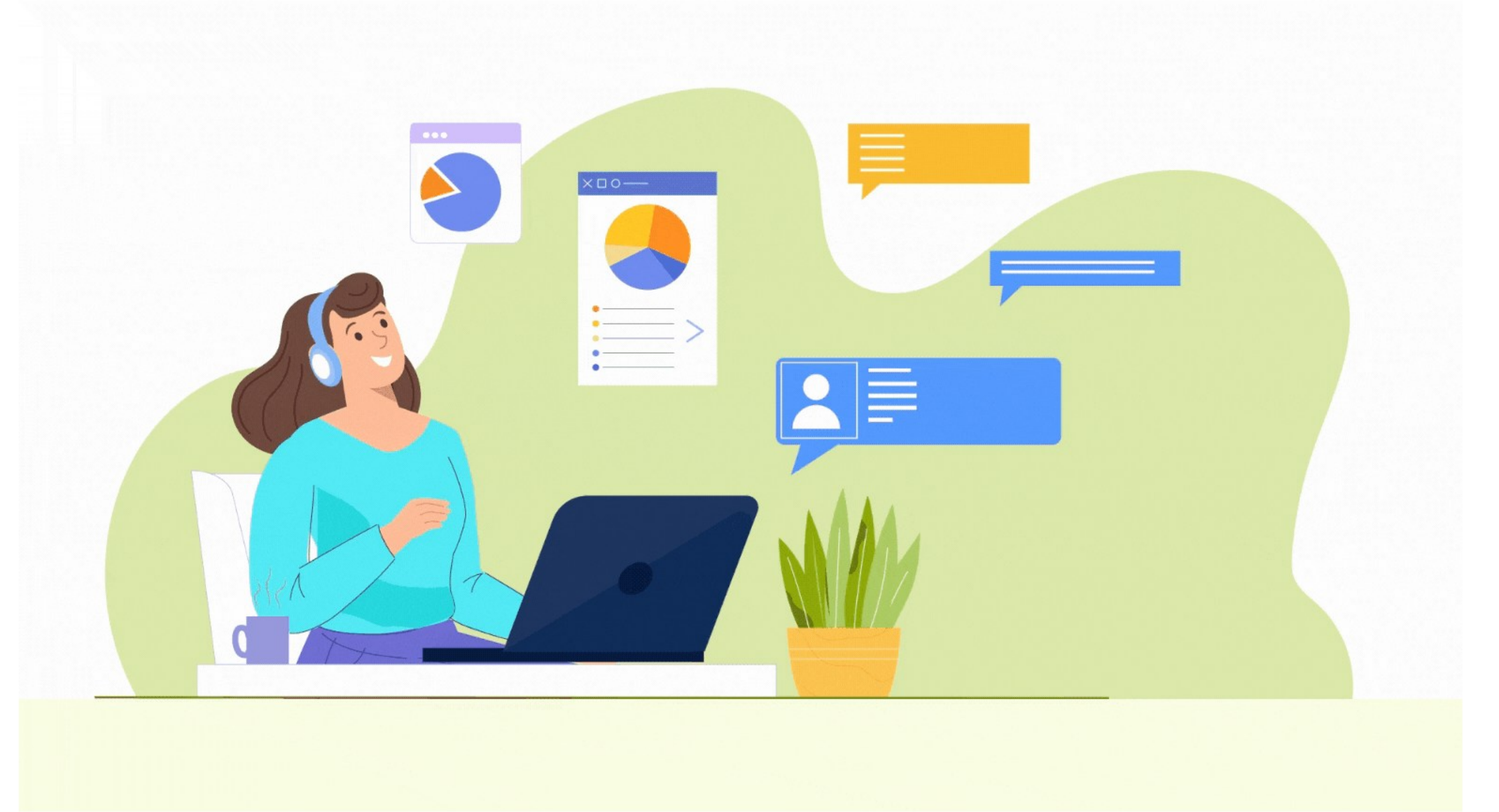
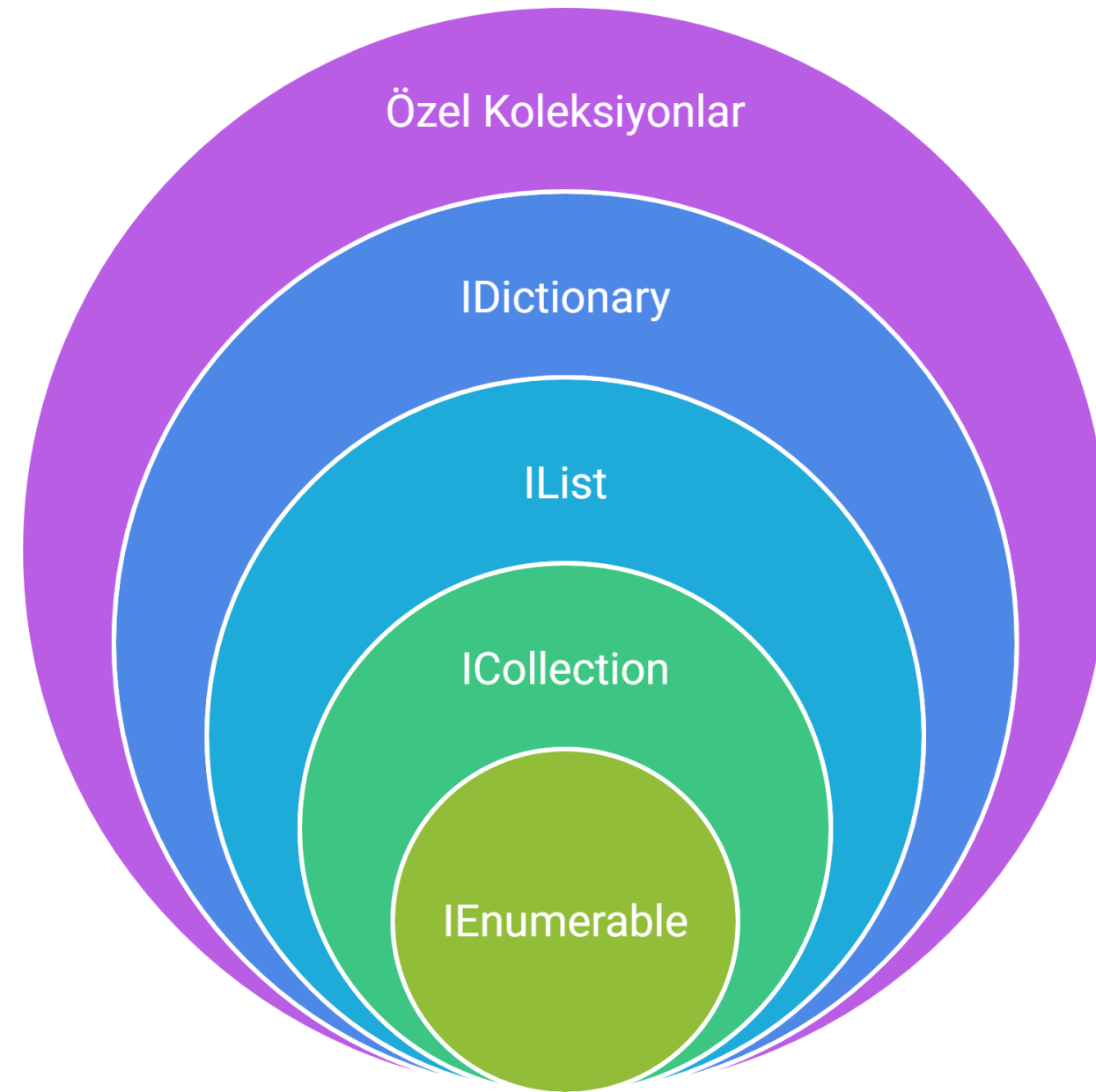


Kutulama, değer tiplerinin referans tipine dönüştürülmesi işlemidir. Yani bellekte stack alanında duran bir değeri alıp, heap alanına taşıyarak object türü gibi bir referans tip içinde saklamak anlamına gelir. Bu işlem sırasında aslında değerın bir kopyası oluşturulur ve bu kopya kutu olarak adlandırabileceğimiz bir referans yapısının içine yerleştirilir. Böylece değer tipi, artık bir referans tipi gibi davranmaya başlar.

Koleksiyonlar

Koleksiyonlar

- Queue, Stack ve BitArray gibi özel yapılar
- Anahtar-değer veri depolamayı destekler
- İndeks tabanlı erişimi sağlar
- Temel koleksiyon işlemlerini tanımlar
- Tüm koleksiyonların temel arayüzü



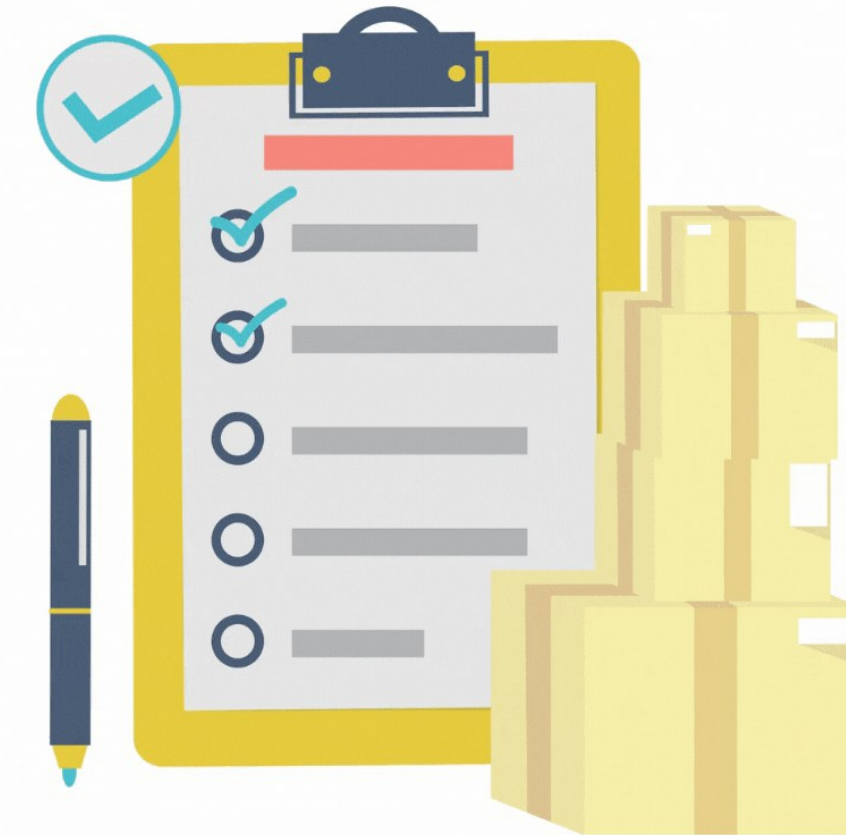
List<T>

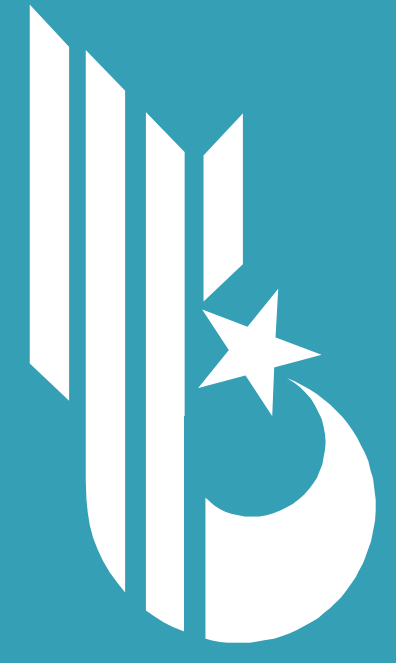
Generic bir koleksiyondur

Dinamik boyutludur

Index tabanlıdır

Hızlı erişim ve arama sağlar

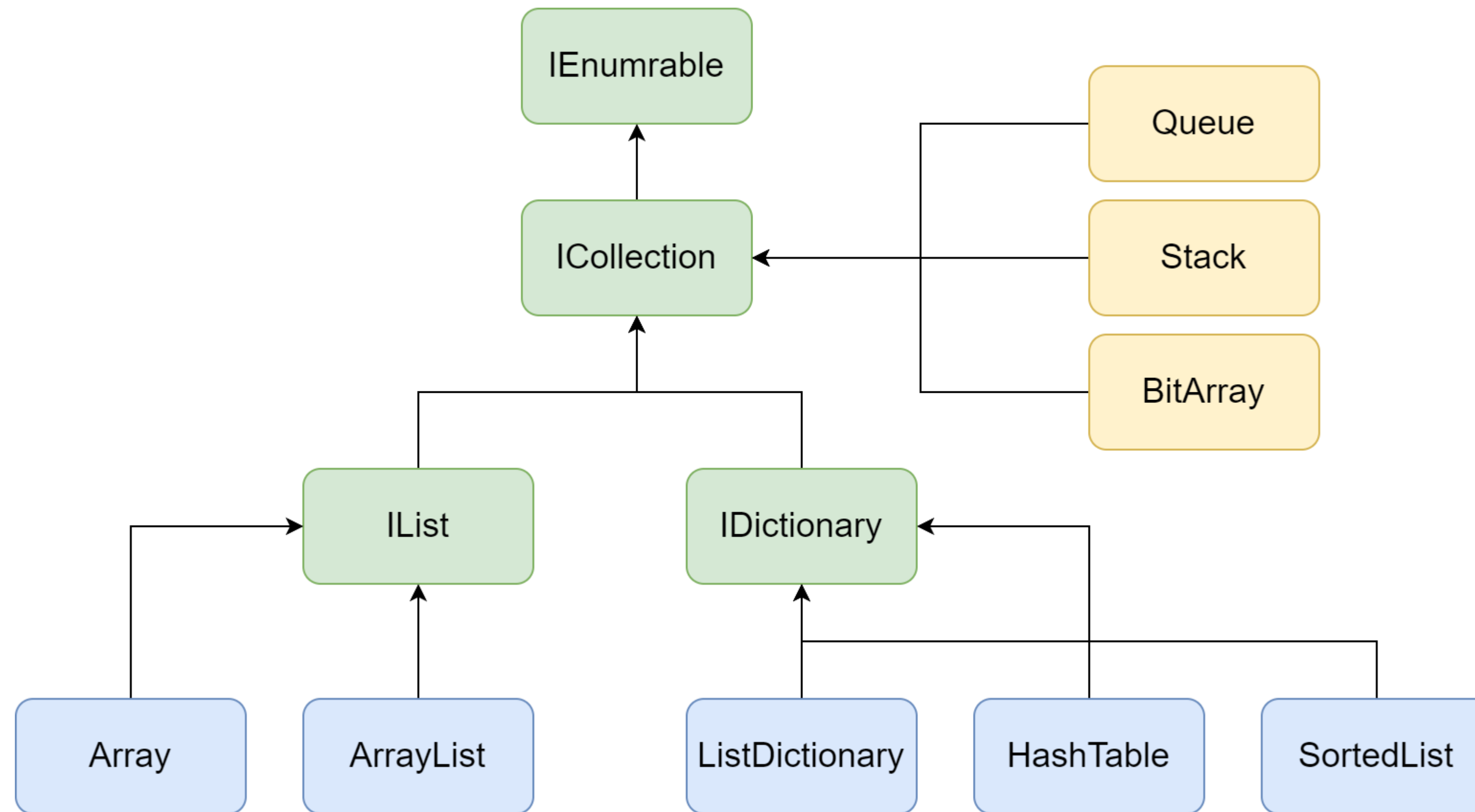




BTK
AKADEMİ

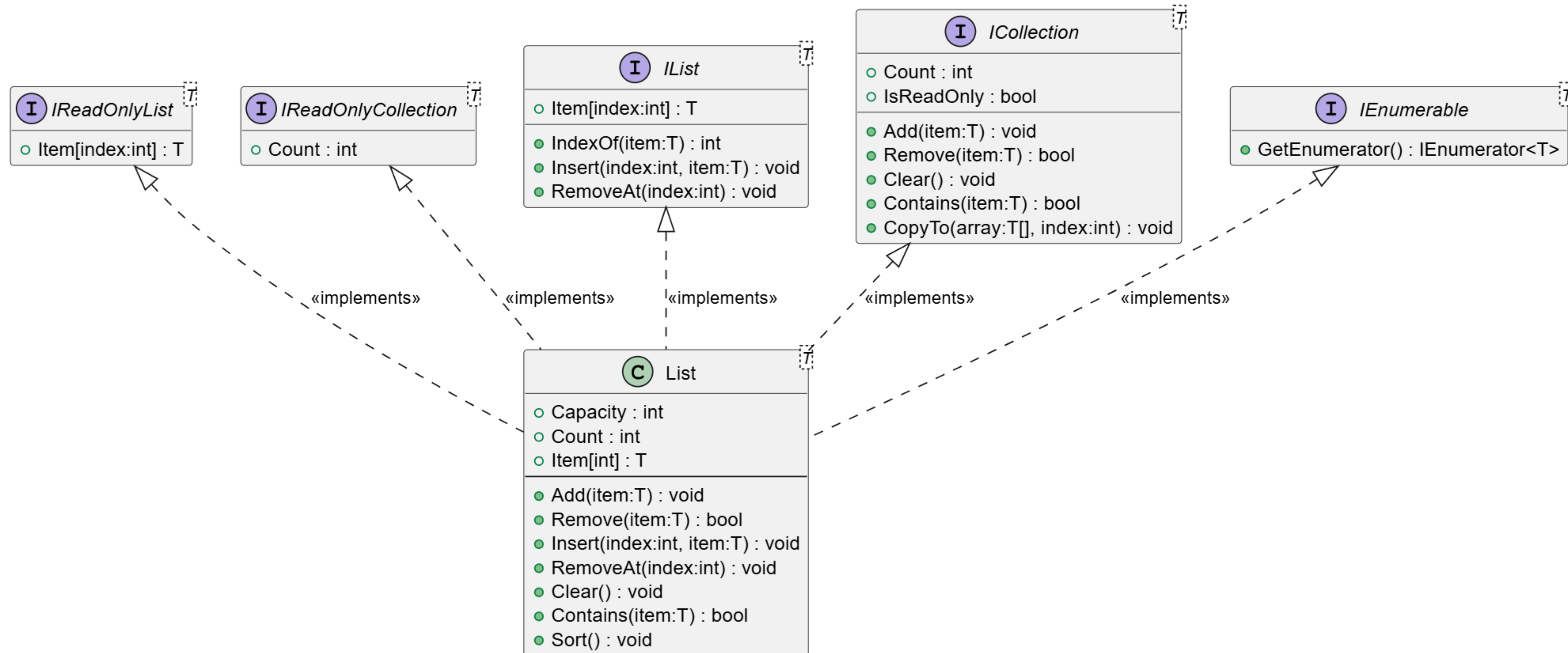
*Generic kullanımı kod şişmesinin
(**code bloat**) önüne geçmektedir.*

Koleksiyonlar



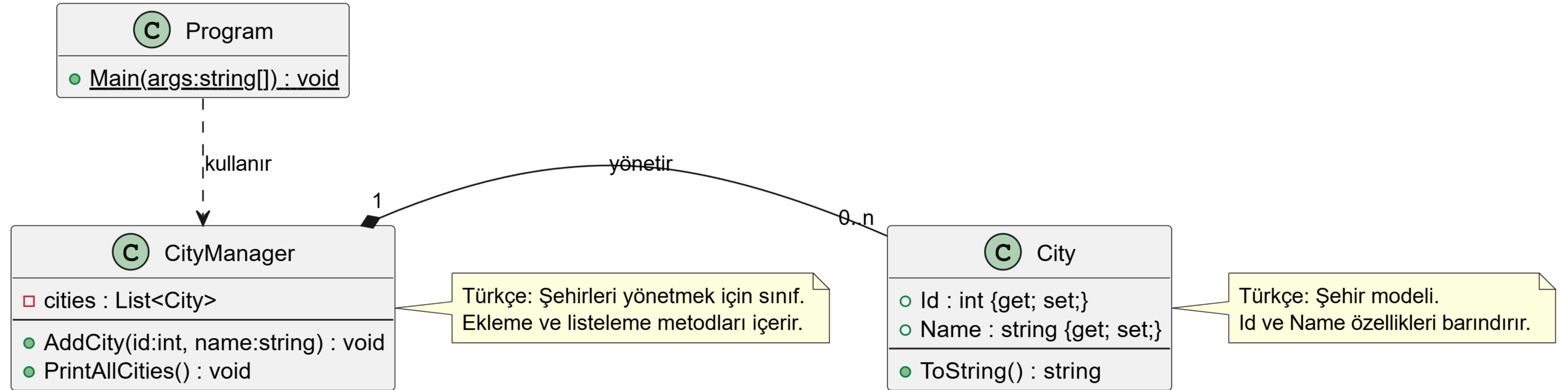
List<T>

List<T> UML Diagram



List<T>

CityManager Demo - Class Diagram (C#)



Parametrik çok biçimlilik (Parametric polymorphism)



Aynı kod yapısının farklı veri tipleriyle çalışmasını sağlar.

Generic sınıflar, metotlar ve arayüzler aracılığıyla uygulanır

Tip güvenliği sunar

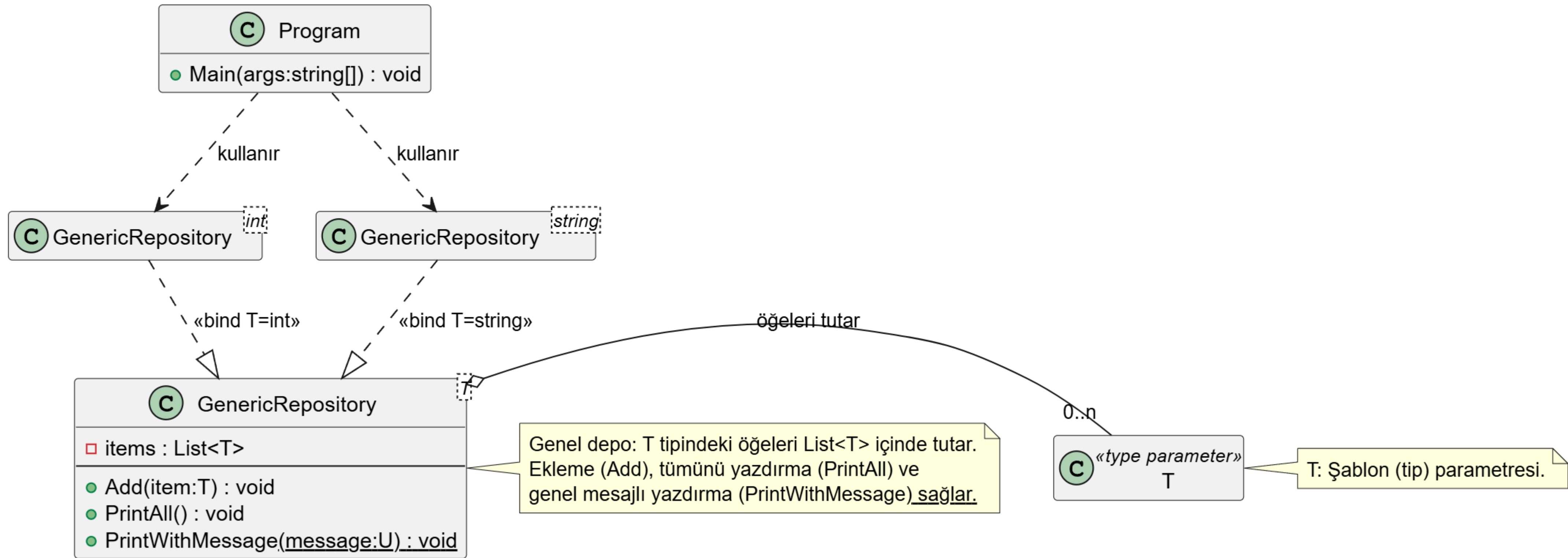
Yeniden kullanılabilirlik sağlar

Genel (Generic)

Constraint	Tanımı
<i>where T: struct</i>	<i>T</i> değer tipli olmalı.
<i>where T: class</i>	<i>T</i> referans tipli olmalı.
<i>where T: IFoo</i>	<i>T IFoo interface</i> implemente etmeli.
<i>where T: Foo</i>	<i>T Foo</i> temel sınıfından türetilmeli.
<i>where T: new()</i>	<i>T default</i> bir <i>constructor</i> sahip olmalı.
<i>where T1: T2</i>	T1, T2 tipinden türetilmeli.

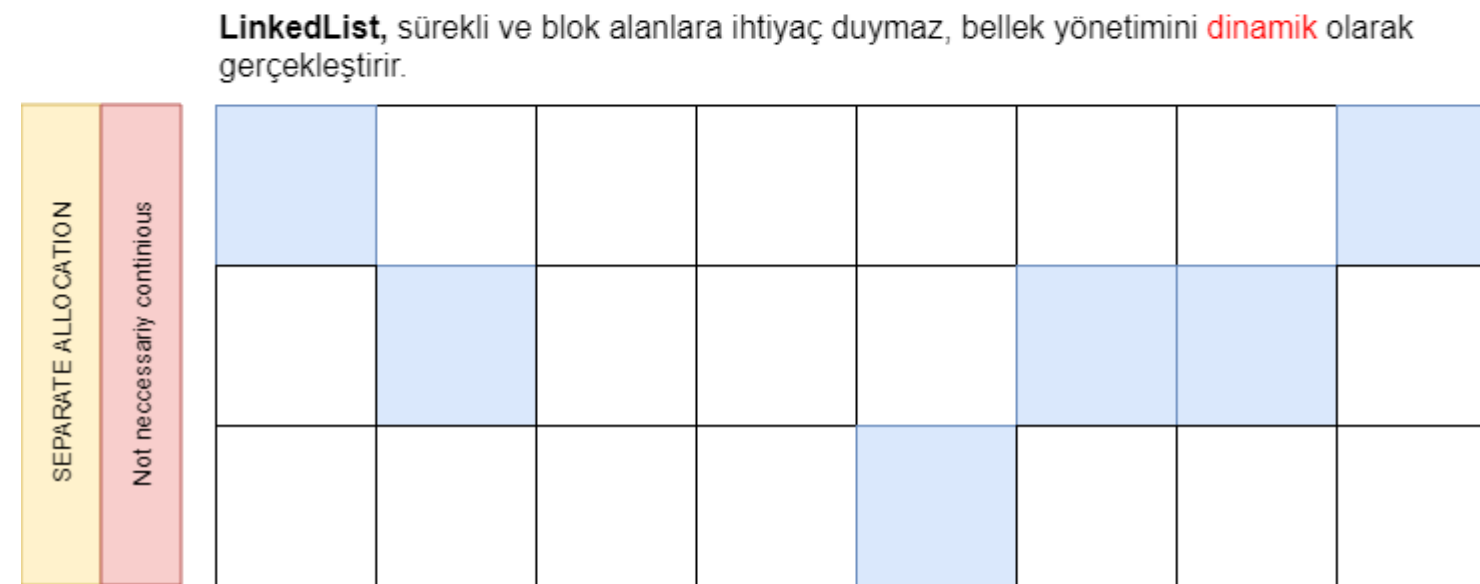
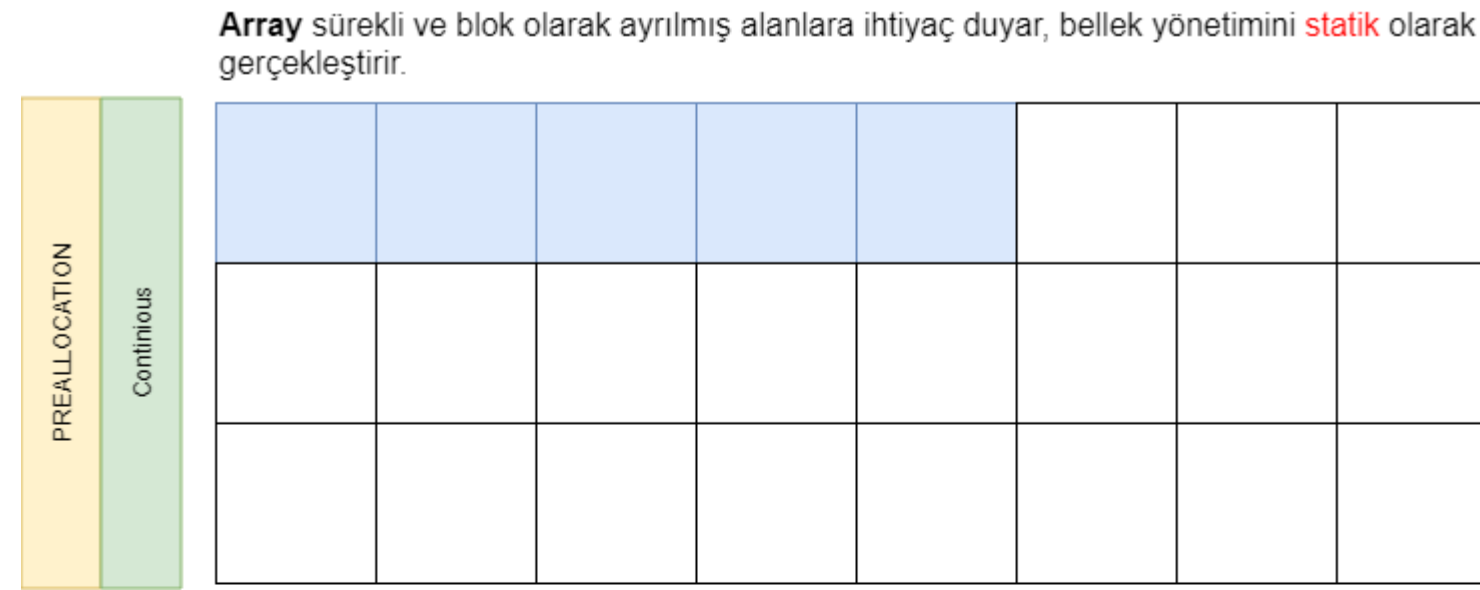
Parametrik çok biçimlilik (Parametric polymorphism)

GenericRepository<T> - Class Diagram (C#)



Bağlı Liste

LinkedList<T>



Generic bir koleksiyondur

Çift yönlüdür

Elemanlar arasında düğüm bağlantısı vardır

LINQ sorguları ile kullanılabilir

Bağlı Liste

LinkedList<T>

google.com

btkakademi.gov.tr

samsun.edu.tr

Bağlı Liste LinkedList<T>

google.com

btkakademi.gov.tr

samsun.edu.tr

← back()

Bağlı Liste LinkedList<T>

google.com

btkakademi.gov.tr

samsun.edu.tr



Bağlı Liste LinkedList<T>

google.com

btkakademi.gov.tr

samsun.edu.tr

← back()

Bağlı Liste

LinkedList<T>

google.com

btkakademi.gov.tr

samsun.edu.tr

forward()



Bağlı Liste LinkedList<T>

google.com

btkakademi.gov.tr

samsun.edu.tr

forward()

Bağlı Liste LinkedList<T>

google.com

btkakademi.gov.tr

samsun.edu.tr

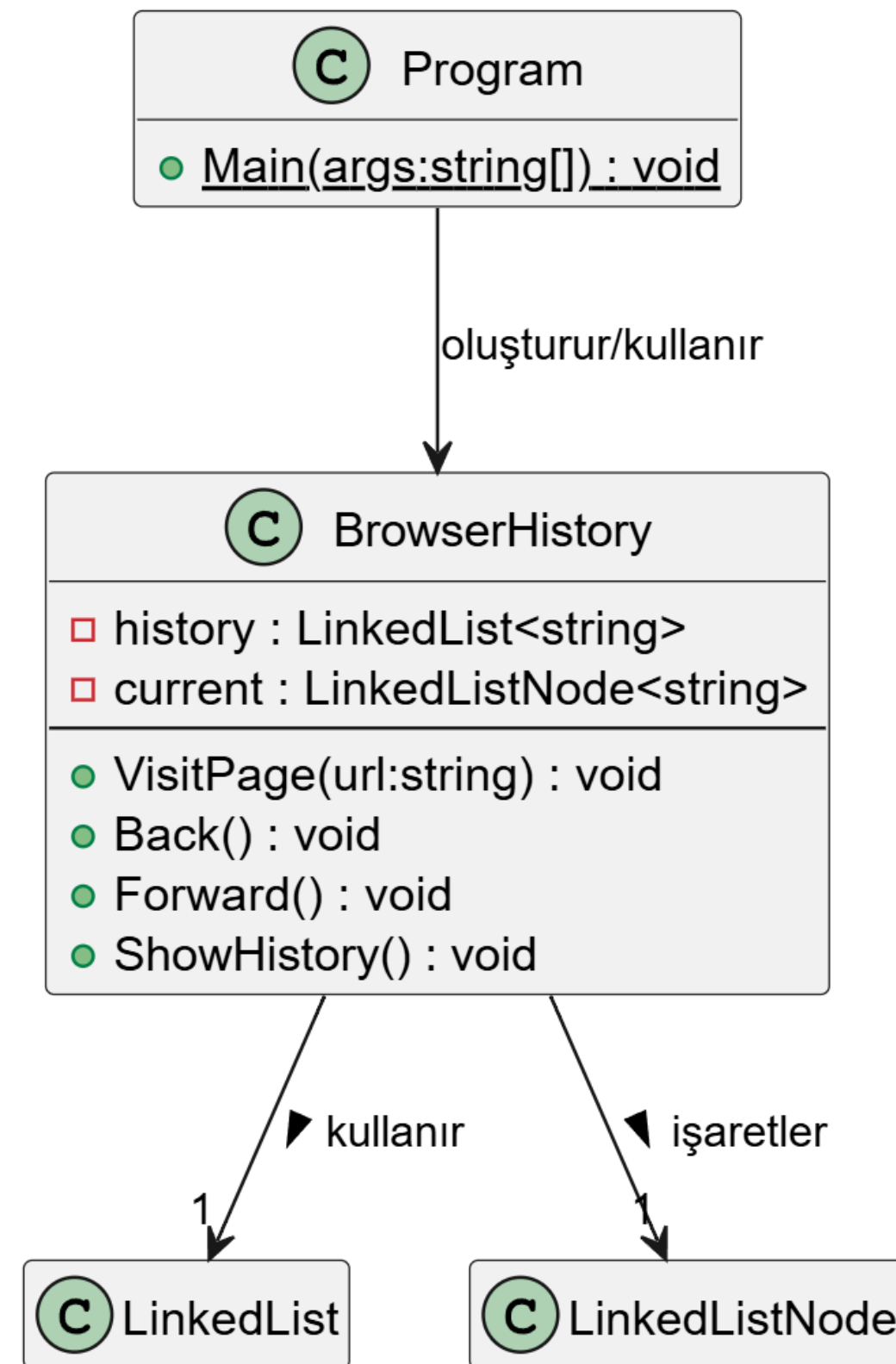
forward()



Bağlı Liste

LinkedList<T>

Browser History Demo - Sınıf Diyagramı (C#)



Yığın Stack<T>

Soyut Veri Türü

Push(item)

Pop()

Peek()

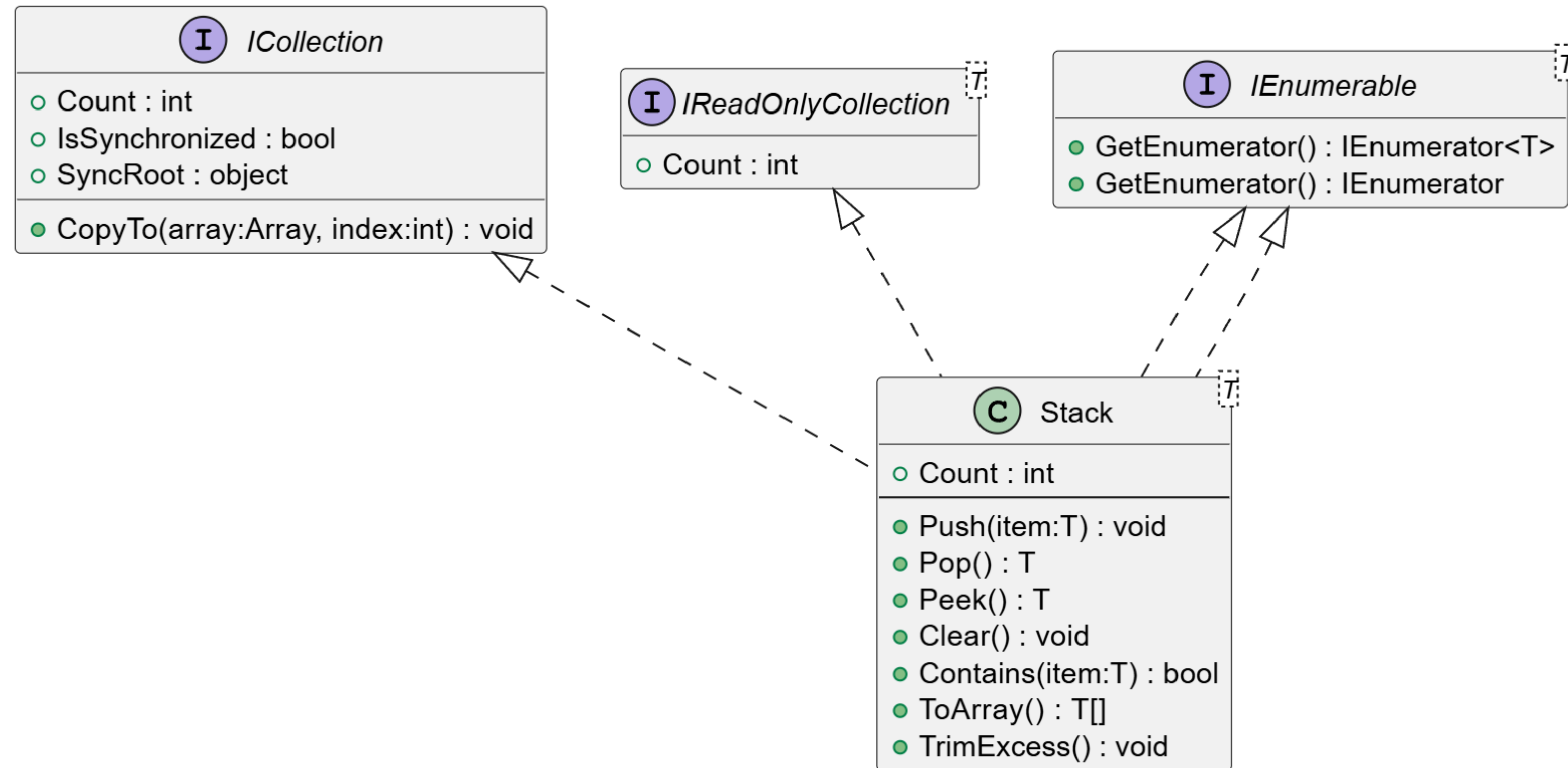
Count



Son giren, ilk çıkar
Last-In, First-Out (LIFO)

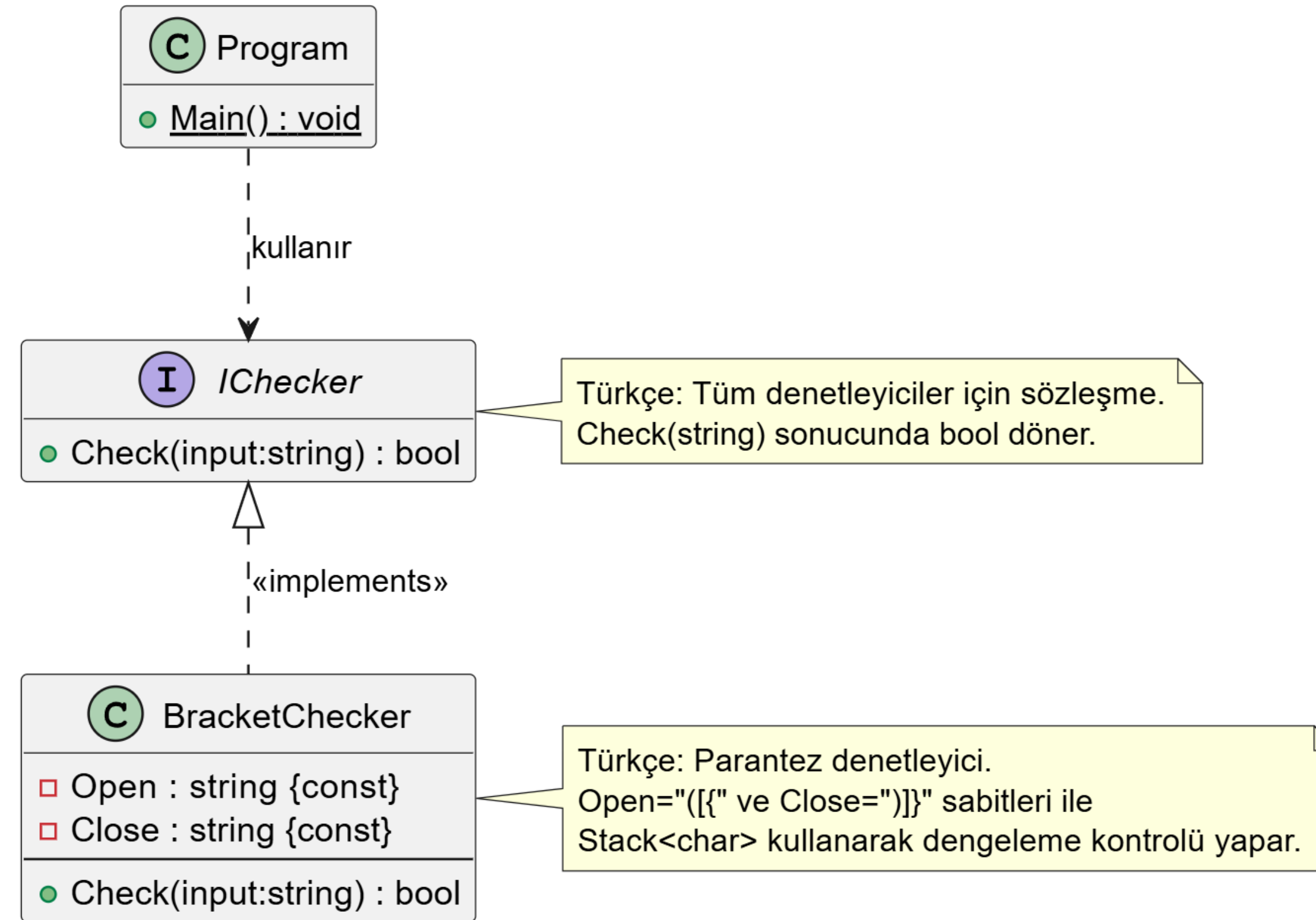
Yığın Stack<T>

Stack<T> UML Diagram



Yığın Stack<T>

BracketChecker - Class Diagram (C#)



Kuyruk Queue<T>

Soyut Veri Türü

Enqueue

Dequeue

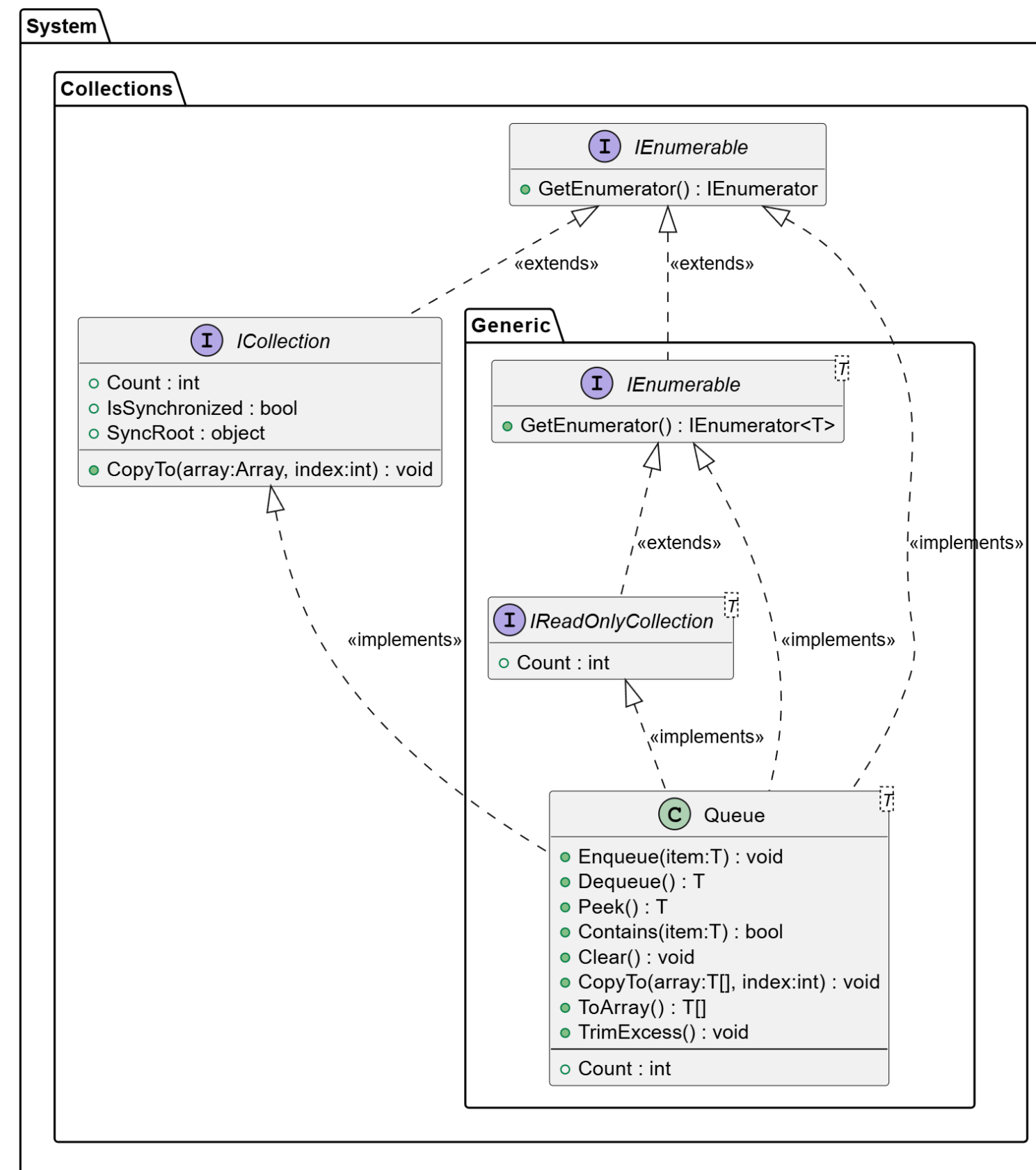
Peek()

Count



İlk giren, ilk çıkar
First-In First-Out (FIFO)

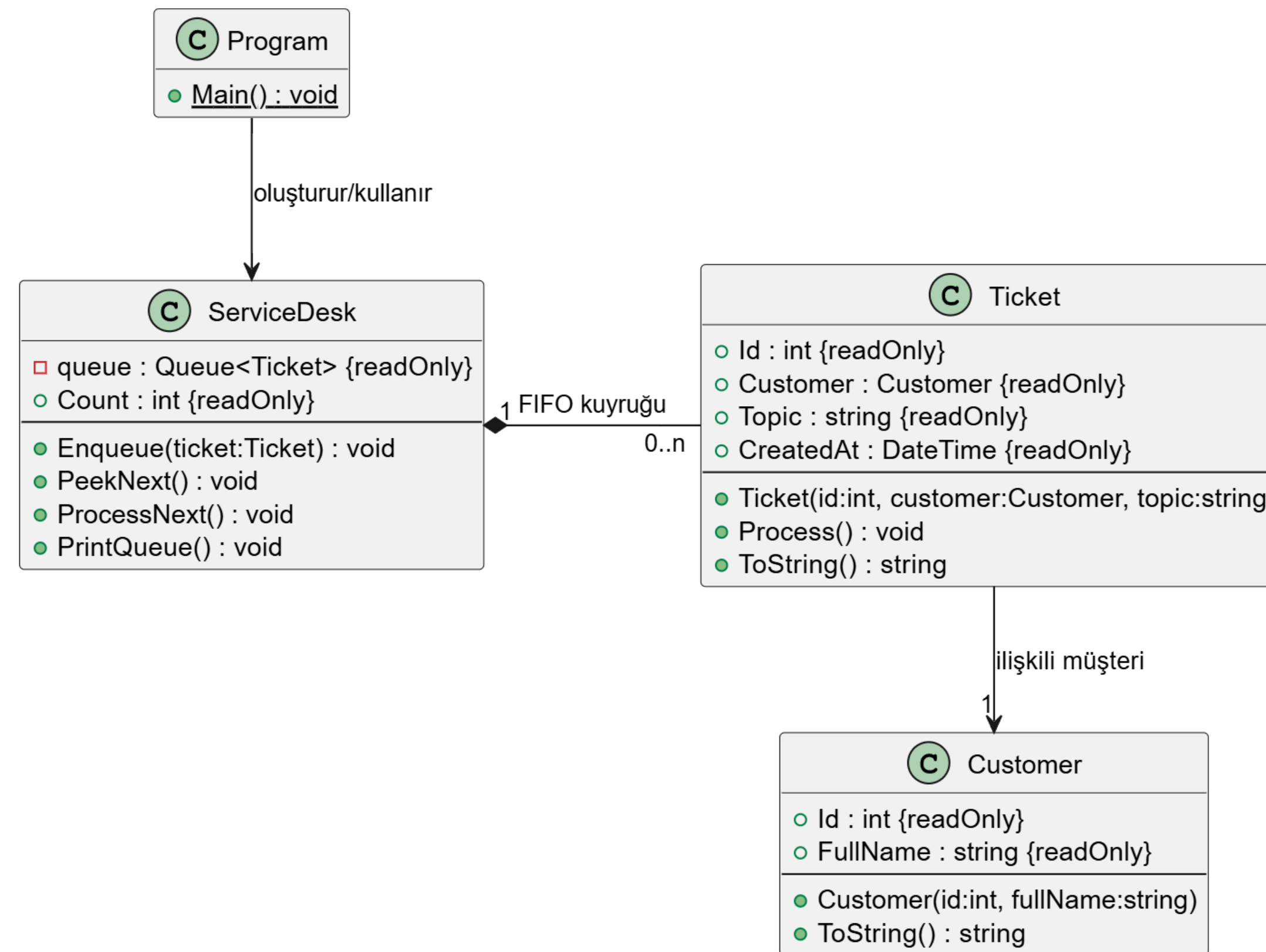
Queue<T> and Related Interfaces — Clean Rework



Kuyruk

Queue<T>

Queue OOP Mini (Customer ile) - Sınıf Diyagramı (C#)



Koleksiyonlar

Diziler

Sabit boyutludur.

İşlevsellik sınırlıdır.

Geliştirme maliyetlidir,
programcının yapması gerekir.

Koleksiyonlar

Esnek ve dinamiktir

Zengin fonksiyonlar sunar

Farklı senaryolar için özel
yapılandırılmıştır.

Teşekkürler

ZAFER CÖMERT
Öğretim Üyesi