

Garbage Collector (Çöp Toplayıcı), C# ve .NET platformlarında bellek yönetimini otomatik olarak gerçekleştiren bir mekanizmadır. GC, kullanılmayan nesneleri bellekten temizleyerek bellek sızıntılarını (memory leak) önler ve uygulamanın performansını optimize eder.

Garbage Collector Nasıl Çalışır?

Garbage Collector, .NET'in Managed Heap (**Yönetilen Yığın**) alanında çalışan nesneleri yönetir. Çalışma mantığı şu adımlarla ilerler:

1. Nesnelerin Tahsis Edilmesi

- Yeni bir nesne oluşturulduğunda, Managed Heap üzerinde bellekte bir alan tahsis edilir.

2. Kullanılmayan Nesnelerin Belirlenmesi

- GC, hangi nesnelerin hala erişilebilir olduğunu belirler.
- Referansı olmayan (erişilemeyen) nesneler işaretlenir.

3. Belleğin Temizlenmesi

- İşaretlenen gereksiz nesneler bellekten kaldırılır.

4. Belleğin Sıkıştırılması

- Hafızadaki boşluklar düzenlenerek parçalanmayı önler.

GC'nin Bellek Yönetimindeki Rolü

- Elle bellek yönetimi gereksinimini ortadan kaldırır.
- Memory Leak (bellek sızıntısı) ve Dangling Pointer (boş referans) gibi hataları önler.
- Uygulamanın performansını artırır, ancak gerektiğinde çalıştığı için anlık gecikmelere neden olabilir.

Generic Sınıflar

Generic sınıflar, farklı veri türleriyle çalışabilen esnek yapılar oluşturmamızı sağlar. T gibi tür parametreleri kullanılarak tanımlanır ve hangi türle çalışacağını nesne oluşturulurken belirleriz.

Generic Yapıların Avantajları

Generic sınıflar farklı türlerle çalışabilen esnek kodlar yazmamızı sağlar. Bunun başlıca avantajları şunlardır:

- Kod Tekrarını Azaltır** : Aynı işlemi farklı türler için tekrar tekrar yazmak yerine tek bir generic yapı kullanılabilir.

2. **Tür Güvenliği Sağlar** :Derleme zamanı (compile-time) hatalarını önleyerek hataların erken tespit edilmesini sağlar.
3. **Performansı Artırır** : Boxing & Unboxing işlemlerini engelleyerek bellek yönetimini iyileştirir.

Boxing Nedir?

- Değer tipinin (value type) referans tipine (object) dönüştürülmesi işlemidir.
- Ekstra bellek tahsisi ve işlem maliyeti oluşturur.

Unboxing Nedir?

- Referans tipinin tekrar değer tipine dönüştürülmesi işlemidir.
- Ekstra işlem yükü ve bellek kullanımı gerektirir.

Boxing & Unboxing Performans Etkisi

- Ekstra bellek kullanımı ve CPU yükü oluşturur.
- Garbage Collector'ı (GC) daha sık çalıştırabilir, bu da uygulamanın yavaşlamasına neden olabilir.
- Generic kullanımı, Boxing & Unboxing'i önleyerek daha verimli bellek yönetimi sağlar.