

# İçindekiler

1. Giriş .....	2
2. C#'ta Kullanılan Erişim Belirleyiciler ve Örnekler .....	2
2.1. Public (Genel) .....	2
2.2. Private (Özel).....	3
2.3. Protected (Korumalı) .....	3
2.4. Internal (Dahili).....	4
3. Nesne Yönelimli Programlamada (OOP) Erişim Belirleyicilerin Rolü .....	5
3.1. Kapsülleme (Encapsulation) .....	5
3.2. Kalıtımda (Inheritance) Kullanımı .....	5
4. Sonuç .....	6
5. Kaynaklar.....	6

# C#'ta Eriřim Belirleyiciler ve Deęiřkenler

---

## 1. Giriř

Nesne Yönelimli Programlama (OOP) paradigması, kodun daha güvenli, modüler ve sürdürülebilir olmasını sağlamak için erişim belirleyiciler (Access Modifiers) kullanır. Eriřim belirleyiciler, bir sınıfın içindeki deęiřkenlerin ve metodların hangi kapsamda erişilebilir olacağını kontrol eder. C# dili, dört temel erişim belirleyicisi sunar:

- **public**
- **private**
- **protected**
- **internal**

Bu erişim belirleyiciler, kapsülleme (encapsulation) ilkesini destekleyerek veri gizliliğini sağlar ve yazılımın daha güvenli olmasına katkıda bulunur.

---

## 2. C#'ta Kullanılan Eriřim Belirleyiciler ve Örnekler

Ařaęıda erişim belirleyicilerinin açıklamaları ve gerçek hayattan benzetmeler ile birlikte C# kod örnekleri yer almaktadır.

### 2.1. Public (Genel)

- Tanım:

public erişim belirleyicisi, bir deęiřkenin veya metodun her yerden erişilebilir olmasını sağlar.

- Gerçek Hayat Benzetmesi:

Bir restoran menüsü herkese açıktır ve tüm müşteriler tarafından görülebilir.

- Kod Örneęi:

```
public class Restoran
{
    public string Menu = "Çorba, Kebap, Tatlı"; // Herkes erişebilir
}

class Program
{
    static void Main()
    {
        Restoran r = new Restoran();
        Console.WriteLine(r.Menu); // Eriřim başarılı
    }
}
```

---

## 2.2. Private (Özel)

- Tanım:

private erişim belirleyicisi, bir değişkenin veya metodun sadece tanımlandığı sınıf içinde erişilebilir olmasını sağlar.

- Gerçek Hayat Benzetmesi:

Bir banka müşterisinin hesap şifresi sadece banka sisteminde tutulur, müşteri dışında kimse erişemez.

- Kod Örneği:

```
class BankaHesabi
{
    private string sifre = "123456"; // Sadece bu sınıfta erişilebilir

    public void SifreGoster()
    {
        Console.WriteLine("Şifre: " + sifre); // Erişim mümkün
    }
}

class Program
{
    static void Main()
    {
        BankaHesabi hesap = new BankaHesabi();

        // Console.WriteLine(hesap.sifre); // HATA: private olduğu için erişilemez!

        hesap.SifreGoster(); // Doğru kullanım
    }
}
```

---

## 2.3. Protected (Korumalı)

- Tanım:

protected erişim belirleyicisi, bir değişkenin/metodun sadece aynı sınıfta ve o sınıftan türetilen alt

sınıflarda erişilebilir olmasını sağlar.

- Gerçek Hayat Benzetmesi:

Bir ailenin soyadı çocuklarına miras olarak geçer, ancak dışarıdan biri bu soyadı değiştiremez.

- Kod Örneği:

```
class Aile
{
    protected string soyadi = "Yılmaz"; // Miras alınabilir ama dışarıdan erişilemez
}

class Cocuk : Aile
{
    public void SoyadiniGoster()
    {
        Console.WriteLine("Soyadım: " + soyadi);
    }
}

class Program
{
    static void Main()
    {
        Cocuk cocuk = new Cocuk();
        cocuk.SoyadiniGoster(); // Erişim mümkün
        // Console.WriteLine(cocuk.soyadi); // HATA: protected değişken dışarıdan erişilemez
    }
}
```

---

## 2.4. Internal (Dahili)

- Tanım:

internal erişim belirleyicisi, bir değişken veya metodun yalnızca aynı projedeki diğer sınıflardan erişilebilir olmasını sağlar.

- Gerçek Hayat Benzetmesi:

Bir şirketin yıllık bütçe raporu, şirket çalışanları tarafından görülebilir ama dış dünyaya kapalıdır.

- Kod Örneği:

```
internal class SirketBilgileri
{
    internal string GelirRaporu = "2025 Yılı: 5 Milyon TL"; // Sadece aynı projeden erişilebilir
}

class Program
{
    static void Main()
    {
        SirketBilgileri sirket = new SirketBilgileri();
        Console.WriteLine(sirket.GelirRaporu); // Erişim başarılı (Aynı projede olduğumuz için)
    }
}
```

---

### 3. Nesne Yönelimli Programlamada (OOP) Erişim Belirleyicilerin Rolü

Erişim belirleyiciler, OOP'nin temel prensiplerinden biri olan kapsülleme (encapsulation) ilkesini destekler.

#### 3.1. Kapsülleme (Encapsulation)

- Veri gizliliğini sağlar.
- Yetkisiz erişimleri engeller.
- Kodun modüler ve sürdürülebilir olmasına yardımcı olur.

Örnek: Bir banka sisteminde, müşteri şifrelerinin private olarak saklanması gerekir. Böylece dışarıdan erişim engellenir.

---

#### 3.2. Kalıtımda (Inheritance) Kullanımı

- protected belirleyicisi, türetilmiş sınıfların temel sınıftaki verilere erişmesine izin verir.
  - private belirleyicisi ise, alt sınıfların bile erişemeyeceği şekilde verileri saklar.
-

## 4. Sonuç

Erişim belirleyiciler, veri güvenliğini sağlamak, kodun modülerliğini artırmak ve nesne yönelimli programlamayı desteklemek için kritik bir rol oynar.

- Genel erişim için **public**,
- Sınıf içi veri saklama için **private**,
- Miras yoluyla veri aktarımı için **protected**,
- Sadece proje içinde erişim için **internal** kullanılır.

Bu mekanizmalar, yazılım geliştirme süreçlerinde verimliliği, güvenliği ve kod organizasyonunu artırır.

---

## 5. Kaynaklar

1. Microsoft Docs – C# Access Modifiers
2. C# Programming Guide – Encapsulation in C#
3. Stack Overflow – Understanding Access Modifiers