

**1.Public:** public erişim belirleyicisi ile işaretlenen bir ögeye herhangi bir yerden erişilebilir. Yani, bu öge sınıf dışından ve hatta diğer projelerden bile erişilebilir.

**Gerçek Hayat Örneği:** Bir okul tarafından yayımlanan bir duyuru. Herkes duyuruyu görebilir ve ondan faydalanabilir.

**OOP Rolü:** public, genel olarak erişilmesi gereken öğeleri belirtmek için kullanılır. Kullanıcıların programın çeşitli alanlarına serbestçe erişebilmesini sağlar.

**2. Private:** private erişim belirleyicisi ile işaretlenen bir öge yalnızca kendi sınıfında erişilebilir. Diğer sınıflardan veya dışarıdan erişim olamaz.

**Gerçek Hayat Örneği:** Bir kişinin günlüğü. Sadece o kişi günlüğünü görebilir, başkalarına erişimi yoktur.

**OOP Rolü:** private, sınıfın iç işleyişini gizler ve dışarıdan müdahaleyi engeller. Bu, sınıfın dışarıya gereksiz veya tehlikeli verilerini sızdırmaktan korunmasına yardımcı olur. Kapsülleme ilkesinin bir parçasıdır.

**3. Protected:** protected erişim belirleyicisi ile işaretlenen bir öge, yalnızca sınıf içinde ve sınıftan türeyen alt sınıflarda (inheritance) erişilebilir.

**Gerçek Hayat Örneği:** Bir ailenin içindeki belirli özellikler (örneğin, evdeki özel bir eşya), yalnızca aile üyeleri tarafından erişilebilir, ancak başka bir aileye katılan yeni üyeler (alt sınıf) de erişebilir.

**OOP Rolü:** protected, alt sınıfların, üst sınıflardan miras aldığı bazı öğelere erişmesini sağlar, ancak dışarıdan erişimi engeller.

**4.Internal:** internal erişim belirleyicisi ile işaretlenen bir öge, yalnızca aynı derlemede bulunan kodlardan erişilebilir. Yani, farklı projelerde veya farklı derlemelerde bu ögeye erişilemez.

**Gerçek Hayat Örneği:** Bir şirketin iç yazılım sisteminde kullanılan araçlar. Bu araçlar yalnızca şirketteki çalışanlar tarafından kullanılabilir, dışarıdaki kişiler erişemez.

**OOP Rolü:** internal, yalnızca bir uygulamanın içindeki bileşenler arasında erişimi sağlar ve dışarıdan gelen erişimi sınırlar.

## Erişim Belirleyicilerin Nesne Yönelimli Programlamadaki (OOP) Rolü

Eriřim belirleyiciler, nesne ynelimli programlamadaki **kapsllemenin** temelini oluřturur. Sınıfın iřsel iřleyiřine sadece belirli seviyelerde eriřim saęlanmasını garanti eder ve yazılımın gvenlięini, esneklięini ve bakımını iyileřtirir. Kapslleme, sınıfların iř iřleyiřini dıřarıdan saklamak ve yalnızca gerekli olan bilgileri dıřarıya sunmak anlamına gelir.

**Eriřim belirleyicilerin nesne ynelimli programlamadaki rolleri řunlardır:**

1. **Veri Gizlilięi ve Gvenlięi:** Sınıfların iřindeki kritik bilgiler (private ve protected ile) dıřarıya sızdırılmaz, bylece yanlış kullanımlar veya dıř mdahaleler engellenir.
2. **Modlerlik ve Bakım Kolaylıęı:** Sınıflar arası etkileřim, sadece belirli sınıfların ve metodların dıřarıya sunulmasıyla sınırlıdır. Bu, yazılımın daha modler olmasını saęlar ve bakım srecini kolaylařtırır.
3. **Esneklik ve Miras Kullanımı:** protected eriřim belirleyicisi, alt sınıfların st sınıfın zelliklerine eriřebilmesine olanak tanır, bylece miras yoluyla esneklik saęlanır.
4. **Yazılımın Stabilitesi:** internal gibi belirleyiciler, sadece belirli bir modlde kullanılmak zere tasarlanmış ęelerin dıřarıdan eriřilememesini saęlar. Bu, uygulamanın dıřarıdan istenmeyen deęiřikliklere karřı daha dayanıklı hale gelmesini saęlar.