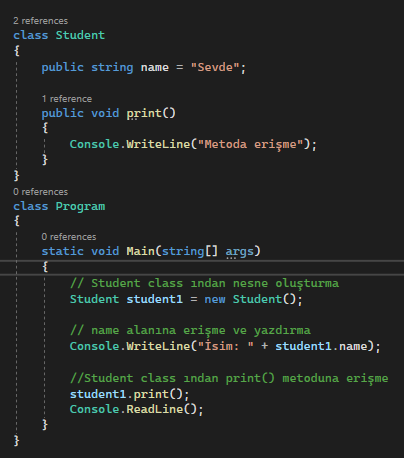
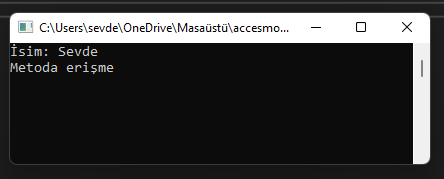
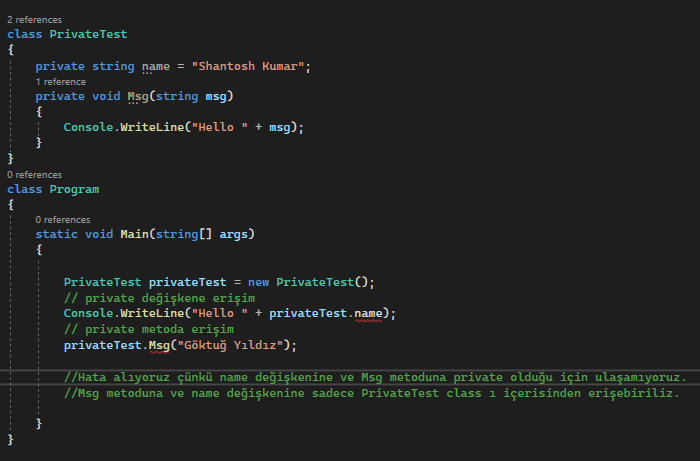
**ACCESS MODIFIERS**

C# programlama dilinde oluşturulan uygulamaların güvenliğini artırmak için, kullanılan sınıfların (class) erişilebilirliğinin kısıtlanması gerekmektedir. Bu anlamda, erişim belirleyiciler (access modifiers) koda dışardan yapılmak istenen müdahalenin sınırlarını belirlemek amacıyla kullanılan anahtar ifadelerdir.

* [**public**](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/public)**:** Uygulayan nesnelere projenin her yerinden erişilebilir. Bu nedenle, herhangi bir erişilebilirlik kısıtlaması yoktur.

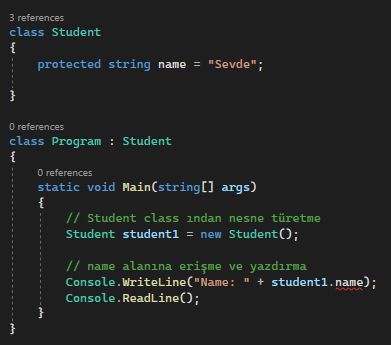


* [**private**](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/private)**:** Uygulayan nesnelere yalnızca bir sınıf veya yapı içinde erişilebilir. Sonuç olarak, oluşturuldukları sınıfın dışında onlara erişemiyoruz.

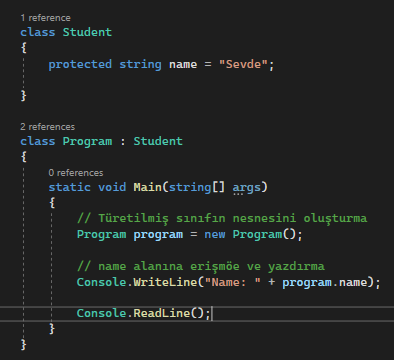


* [**protected**](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/protected)**:** Yalnızca aynı sınıftan ve o sınıftan türetilen tüm sınıflarda erişilebilir olduğunu ima eder.

Aşağıdaki örnekte alan korumalı olduğu için main den erişemiyoruz.



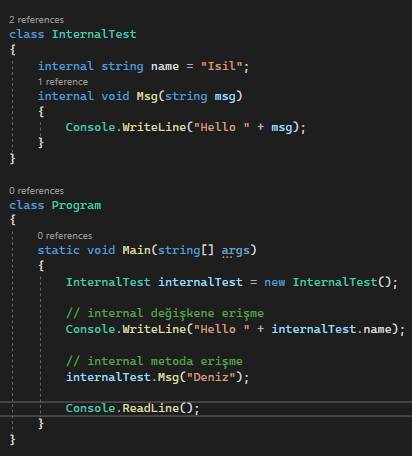
* Alana türetilmiş sınıftan erişme:

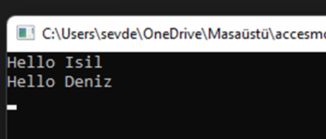


**NOT:** Alan, türetilmiş sınıflara erişebildiği için protected alana erişebiliyoruz.

* [**internal**](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/internal)**:** Yalnızca aynı derleme içinde erişilebilir.
* Kendi projesi içerisinde public, farklı bir projeden/dışarıdan çağırılmak istenildiğinde ise private özelliklerini taşır.
* Birlikte çalışmak ve mantıksal bir işlevsellik birimi oluşturmak için inşa edilmişlerdir.
* Bu nedenle, bir derlemeyi çalıştırdığımızda, derleme içindeki tüm sınıflar ve arayüzler birlikte çalışır.

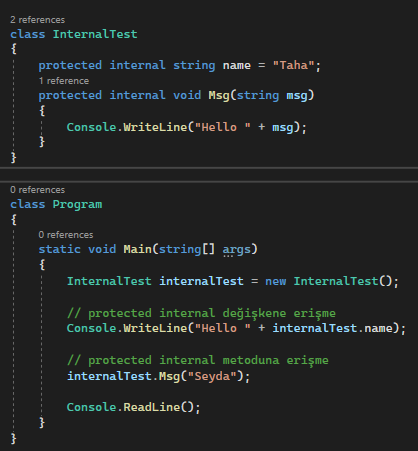
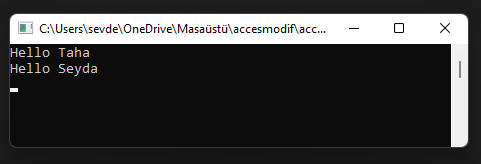
Aşağıdaki örnekte alan ve metod internal olduğu için Program sınıfı içerisinden erişebiliyoruz.





* [**protected internal**](https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/protected-internal) **:** Aynı protected gibi kendi bulunduğu class üzerinde ve bu class ı miras alan classlar üzerinden çağrılabilir. Artı olarak aynı proje (assembly/dll) üzerinde olmasalar dahi, tanımlandığı class’tan türetilmiş diğer class’ların içinden de çağırılabilirler.

*Özel bir durum olmadıkça varsayılan olarak değişkenler ve methodlar private, classlar ise internal dır.*

**

Bu erişimlerin özelliklerini kısaca özetleyecek olursak;

