

Sealed

- sealed, sınıflara ve üyelere (metot ve property) uygulanabilen bir anahtar sözcüktür.
- Anlamı; kapalı, mühürlü şeklinde çevrilebilir.
- Sınıflar için kalıtımı, üyeler için ise override edilmeyi önler.
- Sınıflar üzerine sealed işlemi

```
public class A
{
}

public class B : A
{
}

public class C : B
{
}
```

C sınıfı sealed yaptığımızda
C artık türetilemez.

```
public class A
{
}

public class B : A
{
}

sealed public class C : B
{
}

public class D : C
{
}
```

The screenshot shows the Visual Studio IDE with the following code. Class C is marked as sealed. Class D attempts to inherit from C. The error list at the bottom shows a red 'X' icon and the message: 'D': cannot derive from sealed type 'C'.

Error List

1 Error | 0 Warnings

Description

1 'D': cannot derive from sealed type 'C'

Sealed

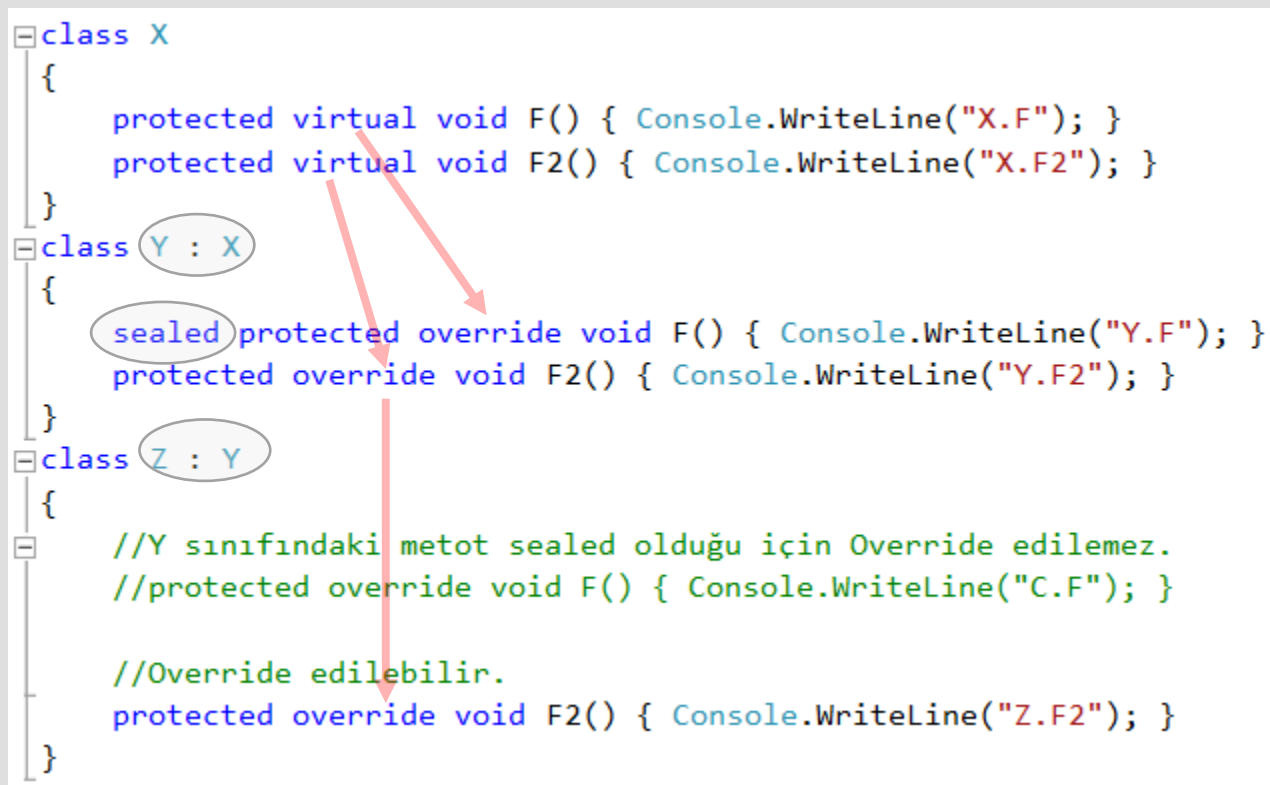
- Sealed metot ve property'e de uygulanabilir.

```
class X
{
    protected virtual void F() { Console.WriteLine("X.F"); }
    protected virtual void F2() { Console.WriteLine("X.F2"); }
}

class Y : X
{
    sealed protected override void F() { Console.WriteLine("Y.F"); }
    protected override void F2() { Console.WriteLine("Y.F2"); }
}

class Z : Y
{
    //Y sınıfındaki metot sealed olduğu için Override edilemez.
    //protected override void F() { Console.WriteLine("C.F"); }

    //Override edilebilir.
    protected override void F2() { Console.WriteLine("Z.F2"); }
}
```



Sealed

- .Net Framework içerisinde örnek vermek gerekirse; string tipi de sealed olarak işaretlenmiştir. Yani string tipinden kalıtım yapamayız.
- string üzerindeyken sağ tuş Goto Definition diyelim

```
Assembly mscorlib.dll, v4.0.0.0

using System.Collections;
using System.Collections.Generic;
using System.Globalization;
using System.Reflection;
using System.Runtime;
using System.Runtime.ConstrainedExecution;
using System.Runtime.InteropServices;
using System.Security;
using System.Text;

namespace System
{
    ...public sealed class String : IComparable, ICloneable, IConvertible, IComparable<string>, IEnumerable<char>, IEnumerable, IEquatable<string>
    {
        ...public static readonly string Empty;

        ...public String(char* value);
        ...public String(char[] value);
        ...public String(sbyte* value);
        ...public String(char c, int count);
        ...public String(char* value, int startIndex, int length);
        ...public String(char[] value, int startIndex, int length);
        ...public String(sbyte* value, int startIndex, int length);
        ...public String(sbyte* value, int startIndex, int length, Encoding enc);
    }
}
```