

Este artículo se tradujo de forma manual. Mueva el puntero sobre las frases del artículo para ver el texto original. [Más información.](#)

Traducción  Original

# Restricciones en el uso de los niveles de accesibilidad (Referencia de C#)

Visual Studio 2013

Cuando se especifica un tipo en una declaración, compruebe si el nivel de accesibilidad del tipo es dependiente del nivel de accesibilidad de un miembro o de otro tipo. Por ejemplo, la clase base directa debe ser al menos tan accesible como la clase derivada. Las siguientes declaraciones producirán un error del compilador, ya que la clase base `BaseClass` es menos accesible que `MyClass`:

```
class BaseClass {...}
public class MyClass: BaseClass {...} // Error
```

La siguiente tabla resume las restricciones de uso de los niveles de accesibilidad declarados.

Contexto	Comentarios
Clases	La clase base directa de un tipo de clase debe ser al menos tan accesible como el propio tipo de clase.
Interfaces	Las interfaces base explícitas de un tipo de interfaz deben ser al menos tan accesibles como el propio tipo de interfaz.
Delegados	El tipo de valor devuelto y los tipos de los parámetros de un tipo delegado deben ser al menos tan accesibles como el propio tipo delegado.
Constantes	El tipo de una constante debe ser al menos tan accesible como la propia constante.
Campos	El tipo de un campo debe ser al menos tan accesible como el propio campo.
Métodos	El tipo de valor devuelto y los tipos de parámetros de un método deben ser al menos tan accesibles como el propio método.
Propiedades	El tipo de una propiedad debe ser al menos tan accesible como la misma propiedad.
Eventos	El tipo de un evento debe ser al menos tan accesible como el propio evento.
Indizadores	El tipo y los tipos de parámetros de un indizador deben ser al menos tan accesibles como el propio indizador.
Operadores	El tipo de valor devuelto y los tipos de parámetros de un operador deben ser al menos tan accesibles como el propio operador.
Constructores	Los tipos de parámetros de un constructor deben ser al menos tan accesibles como el propio constructor.

## Ejemplo

El siguiente ejemplo contiene declaraciones erróneas de diferentes tipos. El comentario que sigue a cada declaración indica el error esperado del compilador.

```
// Restrictions on Using Accessibility Levels
// CS0052 expected as well as CS0053, CS0056, and CS0057
// To make the program work, change access level of both class B
// and MyPrivateMethod() to public.

using System;

// A delegate:
delegate int MyDelegate();

class B
{
    // A private method:
    static int MyPrivateMethod()
    {
        return 0;
    }
}

public class A
{
    // Error: The type B is less accessible than the field A.myField.
    public B myField = new B();

    // Error: The type B is less accessible
    // than the constant A.myConst.
    public readonly B myConst = new B();

    public B MyMethod()

```

```
{
    // Error: The type B is less accessible
    // than the method A.MyMethod.
    return new B();
}

// Error: The type B is less accessible than the property A.MyProp
public B MyProp
{
    set
    {
    }
}

MyDelegate d = new MyDelegate(B.MyPrivateMethod);
// Even when B is declared public, you still get the error:
// "The parameter B.MyPrivateMethod is not accessible due to
// protection level."

public static B operator +(A m1, B m2)
{
    // Error: The type B is less accessible
    // than the operator A.operator +(A,B)
    return new B();
}

static void Main()
{
    Console.WriteLine("Compiled successfully");
}
}
```

## Especificación del lenguaje C#

Para obtener más información, consulte la [Especificación del lenguaje C#](#). La especificación del lenguaje es la fuente definitiva de la sintaxis y el uso de C#.

## Vea también

### Referencia

[Palabras clave de C#](#)

[Modificadores de acceso \(Referencia de C#\)](#)

[Dominio de accesibilidad \(Referencia de C#\)](#)

[Niveles de accesibilidad \(Referencia de C#\)](#)

[Modificadores de acceso \(Guía de programación de C#\)](#)

[public \(Referencia de C#\)](#)

[private \(Referencia de C#\)](#)

[protected \(Referencia de C#\)](#)

[internal \(Referencia de C#\)](#)

### Conceptos

[Guía de programación de C#](#)

### Otros recursos

[Referencia de C#](#)

---

## Adiciones de comunidad

---