

Niveles de Accesibilidad

Todos los tipos y miembros de tipo tienen un nivel de accesibilidad que controla si se pueden usar desde otro código del ensamblado u otros ensamblados. Puede usar los siguientes modificadores de acceso para especificar la accesibilidad de un tipo o miembro cuando lo declare:

[public](#)

Puede obtener acceso al tipo o miembro cualquier otro código del mismo ensamblado o de otro ensamblado que haga referencia a éste.

[private](#)

Solamente el código de la misma clase o estructura puede acceder al tipo o miembro.

[protected](#)

Solamente el código de la misma clase, o de una clase derivada de esa clase, puede acceder al tipo o miembro.

[internal](#)

Puede obtener acceso al tipo o miembro cualquier código del mismo ensamblado, pero no de un ensamblado distinto.

[protected internal](#) Cualquier código del ensamblado en el que se ha declarado, o desde cualquier clase derivada de otro ensamblado, puede acceder al tipo o miembro.

[private protected](#) El código de la misma clase, o de un tipo derivado de esa clase, puede acceder al tipo o miembro solo dentro de su ensamblado que declara.

En los ejemplos siguientes se muestra cómo especificar modificadores de acceso en un tipo y miembro:

```
public class Bicycle
{
    public void Pedal() { }
}
```

No todos los tipos o miembros pueden usar todos los modificadores de acceso en todos los contextos; en algunos casos la accesibilidad de un miembro de tipo está restringida por la accesibilidad de su tipo contenedor. En las secciones siguientes se proporcionan más detalles sobre la accesibilidad.

Accesibilidad de clase y estructura

Las clases y estructuras que se declaran directamente en un espacio de nombres (es decir, que no están anidadas en otras clases o estructuras) pueden ser públicas o internas. Si no se especifica ningún modificador de acceso, el valor predeterminado es internal.

Los miembros de estructura, incluidas las clases y las estructuras anidadas, se pueden declarar como public, internal, o private. Los miembros de clase, incluidas las clases y las estructuras anidadas, pueden ser public, protected internal, protected, internal, private protected o private. El nivel de acceso de los miembros de clase y los miembros de estructura, incluidas las clases y estructuras anidadas, es private de forma predeterminada. Los tipos anidados privados no son accesibles desde fuera del tipo contenedor.

Las clases derivadas no pueden tener mayor accesibilidad que sus tipos base. En otras palabras, no puede tener una clase pública B que derive de una clase interna A. Si se permitiera, convertiría A en público, porque todos los miembros protegidos o internos de A son accesibles desde la clase derivada.

Puede habilitar otros ensamblados concretos para acceder a los tipos internos mediante InternalsVisibleToAttribute.

Accesibilidad de miembros de clase y estructura

Los miembros de clase (incluidas las clases y las estructuras anidadas) se pueden declarar con cualquiera de los seis tipos de acceso. Los miembros de estructura no se pueden declarar como protegidos porque las estructuras no admiten la herencia.

Normalmente, la accesibilidad de un miembro no es mayor que la del tipo que lo contiene. Pero un miembro público de una clase interna podría ser accesible desde fuera del ensamblado si el miembro implementa los métodos de interfaz o invalida los métodos virtuales definidos en una clase base pública.

El tipo de cualquier miembro que sea un campo, propiedad o evento debe ser al menos tan accesible como el propio miembro. Del mismo modo, el tipo devuelto y los tipos de parámetro de cualquier miembro que sea un método, indicador o delegado deben ser al menos tan accesibles como el propio miembro. Por ejemplo, no puede tener un método público M que devuelva una clase C a menos que C también sea público. Del mismo modo, no puede tener una propiedad protegida de tipo A si A se declara como private.

Los operadores definidos por el usuario siempre deben declararse como public.

Los finalizadores no pueden tener modificadores de accesibilidad.

Para establecer el nivel de acceso de un miembro de clase o estructura, agregue la palabra clave adecuada a la declaración de miembro, como se muestra en el ejemplo siguiente.

```
// public class:
public class Tricycle
{
    // protected method:
    protected void Pedal() { }

    // private field:
    private int wheels = 3;

    // protected internal property:
    protected internal int Wheels
    {
        get { return wheels; }
    }
}
```