Clases estáticas y sus miembros (Guía de programación de C#)

Las clases estáticas y sus miembros se utilizan para crear datos y funciones a las que se puede tener acceso sin crear una instancia de la clase. Los miembros de clases estáticas pueden utilizarse para separar datos y comportamientos independientes de cualquier identidad de objeto: los datos y las funciones no cambian, sin importar lo que le ocurra al objeto. Las clases estáticas pueden utilizarse cuando no hay datos ni comportamiento de la clase que dependa de la identidad del objeto.

Clases estáticas

Una clase puede declararse como static, lo que indica que contiene sólo miembros estáticos. No es posible crear instancias de una clase estática mediante la palabra clave new. Common Language Runtime (CLR) de .NET Framework carga automáticamente las clases estáticas cuando se carga el programa o espacio de nombres que contiene la clase.

Utilice una clase estática para contener métodos que no estén asociados a un objeto concreto. Por ejemplo, es un requisito común crear un conjunto de métodos que no actúen sobre datos de instancia y que no estén asociados a un objeto concreto del código. Puede utilizar una clase estática para contener esos métodos.

Las características principales de una clase estática son:

- Sólo contienen miembros estáticos.
- No se puede crear instancias de ellas.
- Son de tipo sealed.
- No pueden contener Constructores de instancias (Guía de programación de C#).

Crear una clase estática es, por consiguiente, muy similar a crear una clase que contiene sólo miembros estáticos y un constructor privado. Un constructor privado evita que se creen instancias de la clase.

La ventaja de utilizar una clase estática es que el compilador puede comprobar que no se agregue accidentalmente ningún miembro de instancia. El compilador garantizará que no se puedan crear instancias de esta clase.

Las clases estáticas son de tipo sealed y, por consiguiente, no pueden heredarse. Las clases estáticas no pueden contener un constructor, aunque sigue siendo posible declarar un constructor estático para asignar los valores iniciales o configurar algún estado estático.

Cuándo utilizar clases estáticas

Suponga que tiene una clase CompanyInfo que contiene los siguientes métodos para obtener información sobre el nombre y la dirección de la compañía.

C#
```
class CompanyInfo
{
    public string GetCompanyName() { return "CompanyName"; }
    public string GetCompanyAddress() { return "CompanyAddress"; }
    //...
}
```

Estos métodos no necesitan adjuntarse a una instancia concreta de la clase. Por consiguiente, en lugar de crear instancias innecesarias de esta clase, puede declararla como una clase estática; por ejemplo:

C#
```
static class CompanyInfo
```

```
{
    public static string GetCompanyName() { return "CompanyName"; }
    public static string GetCompanyAddress() { return "CompanyAddress"; }
    //...
}
```

Utilice una clase estática como unidad de organización para métodos no asociados a objetos concretos. Además, una clase estática puede hacer que la implementación sea más sencilla y rápida, porque no es necesario crear un objeto para llamar a sus métodos. Es conveniente organizar los métodos dentro de la clase de forma significativa, como los métodos de la clase Math del espacio de nombres System.

Miembros estáticos

Se puede llamar a métodos, campos, propiedades o eventos estáticos de una clase aunque no se haya creado ninguna instancia de la misma. Si se crea alguna instancia de la clase, no puede utilizarse para tener acceso al miembro estático. Sólo existe una copia de los campos y eventos estáticos, y los métodos y propiedades estáticos sólo pueden tener acceso a campos y eventos estáticos. Los miembros estáticos suelen utilizarse para representar datos o cálculos que no cambian según el estado del objeto; por ejemplo, una biblioteca matemática puede contener métodos estáticos para calcular el seno y el coseno.

Los miembros de clase estáticos se declaran utilizando la palabra clave **static** delante del tipo de valor devuelto del miembro, por ejemplo:

C#
```
public class Automobile
{
    public static int NumberOfWheels = 4;
    public static int SizeOfGasTank
    {
        get
        {
            return 15;
        }
    }
    public static void Drive() { }
    public static event EventType RunOutOfGas;

    //other non-static fields and properties...
}
```

Los miembros estáticos se inicializan antes de que se tenga acceso al miembro estático por primera vez y antes de llamar al constructor estático, si se va a llamar a alguno. Para tener acceso a un miembro de clase estático, utilice el nombre de la clase en lugar de un nombre de variable para especificar la ubicación del miembro. Por ejemplo:

C#
```
Automobile.Drive();
int i = Automobile.NumberOfWheels;
```

static (Referencia de C#)

Utilice el modificador **static** para declarar un miembro estático, que pertenece al propio tipo en vez de a un objeto específico. El modificador **static** puede utilizarse con clases, campos, métodos, propiedades operadores y eventos, pero no puede utilizarse con indizadores, destructores o tipos que no sean clases. Por ejemplo, la siguiente clase se declara como **static** y solo contiene métodos **static**:

```
    static class CompanyEmployee
{
   public static string GetCompanyName(string name) { ... }
   public static string GetCompanyAddress(string address) { ... }
}
```

Comentarios

Una declaración de constante o tipo constituye, implícitamente, un miembro estático.

No se puede hacer referencia a un miembro estático por medio de una instancia. En vez de ello, se debe hacer referencia por medio del nombre de tipo. Por ejemplo, considere la siguiente clase:

```
public class MyBaseC
{
   public struct MyStruct
   {
      public static int x = 100;
   }
}
```

Para referirse al miembro estático x, use el nombre completo (a menos que sea accesible desde el mismo ámbito):

```
MyBaseC.MyStruct.x
```

Mientras que una instancia de una clase contiene una copia independiente de todos los campos de instancia de la clase, sólo existe una copia de cada campo estático.

No es posible utilizar this para hacer referencia a descriptores de acceso de propiedades o métodos static.

Si la palabra clave **static** se aplica a una clase, todos los miembros de la clase deben ser estáticos.

Las clases, incluidas las clases estáticas, pueden tener constructores estáticos. Se llama a los constructores estáticos en algún momento comprendido entre el inicio del programa y la creación de instancias de la clase.