Programación Visual 3

Miguel Angel Barba Contreras

PVT14

20/11/19



**Delegados y Eventos**

Una fuente de confusión es la sobrecarga del término "delegado". Algunas veces se usa para referirse a un tipo de delegado, y en otras ocasiones puede usarse para referirse a una instancia de un tipo de delegado. Usaré "tipo de delegado" e "instancia de delegado" para distinguir entre ellos y "delegar" cuando hable sobre el tema completo en un sentido general.

## **Tipos de delegado**

Los tipos de delegados se declaran con la palabra delegate clave. Pueden aparecer solos o anidados dentro de una clase.

Tenga en cuenta que la palabra delegate clave no siempre significa que se está declarando un tipo de delegado. Se utiliza la misma palabra clave al crear instancias del tipo delegado utilizando métodos anónimos.

Los tipos declarados aquí derivan de System.MulticastDelegate, que a su vez derivan de System.Delegate. En la práctica, solo verá los tipos de delegado derivados MulticastDelegate. La diferencia entre Delegatey MulticastDelegate es en gran parte histórica; en versiones beta de .NET 1.0 la diferencia fue significativa (y molesta): Microsoft consideró fusionar los dos tipos, pero decidió que era demasiado tarde en el ciclo de lanzamiento para hacer un cambio tan importante. Puedes fingir que son solo un tipo.

### **Crear instancias delegadas**

Los puntos clave de datos en cualquier instancia de delegado en particular son el método al que se refiere el delegado y una referencia para llamar al método en (el destino). Para los métodos estáticos, no se requiere un objetivo. El CLR en sí mismo admite otras formas de delegado ligeramente diferentes, donde el primer argumento pasado a un método estático se mantiene dentro del delegado, o el objetivo de un método de instancia se proporciona como un argumento cuando se llama al método.

Entonces, ahora que conocemos los dos datos necesarios para crear una instancia (junto con el tipo en sí, por supuesto), ¿cómo le decimos al compilador cuáles son? Usamos lo que la especificación de C # llama una expresión de creación de delegado que tiene la forma new delegate-type (expresión). La expresión debe ser otro delegado del mismo tipo (o un tipo de delegado compatible en C # 2.0) o un grupo de métodos: el nombre de un método y, opcionalmente, un destino, especificado como si estuviera llamando al método, pero sin los argumentos o soportes.

**Delegados combinados**

Los delegados se pueden combinar de modo que cuando llame al delegado, se llame a una lista completa de métodos, potencialmente con diferentes objetivos. Cuando dije antes que un delegado contenía un objetivo y un método, eso fue una ligera simplificación. Eso es lo que contiene una instancia de delegado que representa un método. En aras de la claridad, me referiré a tales instancias de delegado como delegados simples. La alternativa es una instancia de delegado que es efectivamente una lista de delegados simples, todos del mismo tipo (es decir, que tienen la misma firma). Llamaré a estos delegados combinados. Los delegados combinados pueden combinarse entre sí, creando efectivamente una gran lista de delegados simples de la manera obvia.

Es importante comprender que las instancias delegadas siempre son inmutables. Cualquier cosa que los combine (o quite uno del otro) crea una nueva instancia de delegado para representar la nueva lista de objetivos / métodos para llamar. Esto es como las cadenas: si llama, String.PadLeft por ejemplo, en realidad no cambia la cadena a la que la llama, solo devuelve una nueva cadena con el relleno adecuado.

## **Eventos**

**Los eventos no son instancias delegadas.**

Los eventos son pares de métodos, adecuadamente decorados en IL para unirlos y hacer que los idiomas sepan que los métodos representan eventos. Los métodos corresponden a operaciones de agregar y quitar, cada una de las cuales toma un parámetro de instancia de delegado del mismo tipo (el tipo del evento). Lo que haga con esas operaciones depende de usted, pero el uso típico es agregar o eliminar el delegado de una lista de controladores para el evento.

Los eventos en sí pueden ser declarados de dos maneras. El primero es con métodos explícitos de agregar y quitar, declarados de manera muy similar a las propiedades, pero con la palabra  event clave.