**Tipos de delegados**

Un tipo de delegado es como una interfaz con un solo método. Este especifica la firma del método y cuando se tiene la instancia puedes llamarlo como si fuera un método. La razón de la existencia delegado es el llamar con una firma particular. Los delegados tienen una referencia a un método y una referencia al objeto de destino al que se debe llamar método. Los tipos de delegados se declaran con la palabra “delegate” y pueden aparecer solos o anidados dentro de una clase

Se debe tener en cuenta que la palabra delégate no siempre significa que se esta declarando un tipo de delegado. se puede utilizar la palabra delegate para crear instancia del tipo de delegado utilizando métodos anónimos.

**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.

Llamar instancias delegadas:

El método al que hace referencia la instancia de delegado se llama en el objeto de destino (si hay uno) y se devuelve el resultado.

**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 se dice que un delegado contenía un objeto y un método, eso fue una ligera simplificación. Eso es lo que contiene una instancia de delegado que representa un método. 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.

La combinación de dos instancias de delegado generalmente se realiza utilizando el operador de suma, como si las instancias de delegado fueran cadenas o números. Restar uno de otro generalmente se hace con el operador de resta. Tenga en cuenta que cuando resta un delegado combinado de otro, la resta funciona en términos de listas.

Las instancias delegadas también se pueden combinar con el Delegate.Combinemétodo estático, y una se puede restar de otra con el Delegate.Removemétodo estático. El compilador de C # convierte los operadores de suma y resta en llamadas a estos métodos. Debido a que son métodos estáticos, funcionan fácilmente con nullreferencias.

**Eventos:**

Los eventos no son instancias delegadas: para entender los eventos lo que uno debe hacer es pensar en ellas como propiedades. S bien los eventos parecen instancias delegadas en terminas de forma en la que expresa operaciones de agregar y quitar no lo son.

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 métodos de agregar y quitar se llaman en C # usando eventName += delegateInstance; y eventName -= delegateInstance; respectivamente, donde eventNamese pueden calificar con una referencia (por ejemplo, myForm.Click) o un nombre de tipo (por ejemplo, MyClass.SomeEvent). Los eventos estáticos son relativamente raros.

**Eventos de campo**

C # proporciona una manera simple de declarar una variable delegada y un evento al mismo tiempo. Esto se llama un evento similar a un campo y se declara de manera muy simple: es lo mismo que la declaración de evento "a largo plazo", pero sin la parte "cuerpo":

public event EventHandler MyEvent;

Esto crea una variable delegada y un evento, ambos con el mismo tipo. El acceso al evento está determinado por la declaración del evento (por lo que el ejemplo anterior crea un evento público, por ejemplo) pero la variable delegada siempre es privada. El cuerpo implícito del evento es el obvio para agregar / eliminar instancias de delegado a la variable delegada, pero los cambios se realizan dentro de un bloqueo.

**Delegar instancias: otros métodos**

Anteriormente vimos cómo una llamada a someDelegate (10) es realmente una abreviatura para someDelegate.Invoke(10). Los tipos de delegados también pueden permitir un comportamiento asincrónico utilizando el BeginInvoke/ EndInvokepar. Estos son opcionales en lo que respecta a la especificación CLI, pero los tipos de delegado de C # siempre los proporcionan. Siguen el mismo modelo para la ejecución asincrónica que el resto de .NET, lo que permite proporcionar un controlador de devolución de llamada, junto con un objeto para almacenar información de estado. Los delegados se ejecutan en subprocesos creados por el grupo de subprocesos del sistema.

Los delegados proporcionan una manera simple de representar una llamada a un método, potencialmente con un objeto de destino, como un dato que se puede pasar. Son la base de los eventos, que son convenciones efectivas para agregar y eliminar el código del controlador que se invocará en los momentos apropiados.