José María Isay Villarruel Castellanos

La gente comúnmente se le dificulta el saber las diferencias sobre un evento y un delegado.

Muchas veces podemos pensar que un delegado es una pequeña interface con un solo método, pero más específicamente es una asignación del método el cual instanciaste, con esto puedes llamar al método con la misma signatura dada. Los delegados mantienen una referencia de un método, y otra referencia la cual apunta al objeto del método para poderlo llamar. Al momento de declarar un delegado es necesario declarar el tipo al cual va pertenecer ese delegado

public delegate string FirstDelegate ( int x ); En esta primera declaracion se recibe como dato un entero para regresar el dato como un tipo carácter.

public delegate void SecondDelegate ( char a, char b ); En la segunda declaración estamos recibiendo dos datos de tipo carácter y el delegado no retornara nada por que es de tipo vacio.

Para poder usar nuestros delegados es necesario el instanciarlos.

FirstDelegate d1 = new FirstDelegate ( InstanceMethod );

FirstDelegate d2 = new FirstDelegate ( this.InstanceMethod );

Las anteriores dos lineas son unas expresiones que son equivalentes cuando instanciamos el método que esta dentro de la clase.

FirstDelegate d3 = new FirstDelegate ( anotherInstance.InstanceMethod );

En la anterior línea se creo una instancia del delegado la cual hace referencia sobre el mismo método, pero con diferente apuntador.

FirstDelegate d4 = new FirstDelegate ( instanceOfOtherClasss.OtherInstanceMethod );

En la anterior instancia del delegado llamamos o instanciamos un método de una diferente clase con un especifico apuntador el cual llama a ese método.

FirstDelegate d5 = new FirstDelegate ( StaticMethod );

La anterior instancia se usa cuando quieres usar un método estático el cual proviene de una clase base.

FirstDelegate d6 = new FirstDelegate ( OtherClass.OtherStaticMethod );

La anterior instancia se usa cuando quieres usar un método estático pero el cual proviene de una clase distinta.

Las instancias de los delegados se llaman como si fueran los propios métodos. para poder llamar al delegado mencionado solamente hacemos referencia de el

String result = d1(10);

Los delegados pueden combinarse ya que un solo delegado se refiere a un delegado simple pero cuando combinamos varios delegados, se convierten en una lista de delegados, esto puede ser útil al momento de trabajar con mucha información.

Los eventos proporcionan un medio apropiado para que los objetos puedan señalizar cambios de estado que pueden resultar útiles para los usuarios o el programador y los eventos se declaran mediante los delegados por eso la conexión entre ellos.

Hay una manera simple de declarar una variable delegada y un evento al mismo tiempo o juntos, esto se hace de la siguiente manera;

public evento EventHandler MyEvent;

¿Por qué se hace esto?, ¿cuál es el punto de tener a ambos juntos los delegados y los eventos? simplemente por encapsulación.