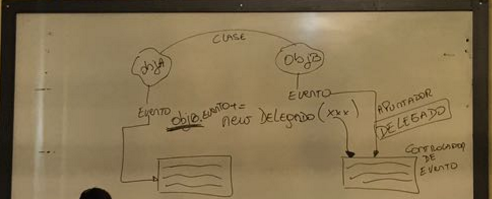
Un evento es una accion que esta definida en una clase pero a diferencia de una operacion no tenie codigo,Eran los objetos creados de dicha clase los que le puede asignar de forma individual codigo al evento(si tenemos 100 objetos de la misma clase,estos pueden hacer 100 acciones diferentes al mismo evento)

(delegados a eventos en mdsc(microsoft))

**Delegado**

Es un puntero a un método de memoria(me va a indicar donde esta)Esto significa que cada vez que cree un delegado voy a tener que indicarle a donde va a tener que apuntar

New delegado(“Nombre controlador”)(Tenemos que hacer el controlador a mano para que funcione)



**Controladores del evento(codigo del cuadrado)**

Es una operacion,que puede recibir parametros,para que llegen datos al controlador el delegado es quien se los da(es la unica forma)

Probocar un evento programaticamente(echo a mano) es esactamente lo mismo que ejecutar una operacion

La provocación del evento es quien va a buscar el delegado para llegar al controlador por lo tanto si el delegado tiene que darle datos al controlador por lo tanto los parametros se los da el evento al delegado y el delegado al controlador

**Como se define un delegado**

1. Podemos usar uno de los del net framework predefinido (Event handler listado de delegados
2. Crear mi propio delegado

Sintaxis para crear delegado:

“TipoAcceso” delegate “TipoDevolucion” nombre(parametros);

**TipoAcceso** si no es public todo lo que tenga que ver con el acceso tiene que estar en mismo archivo(por lo general

**Delegate** define que tipo de elemento estoy definiendo

**Tipo devolución** para los delegados cuando trabajamos con elementos es void

Esta devolución es void ya que aunque el remitente que provoque el evento define quien es el receptor que lo va a manejar el ciclo no se cierra ya que el receptor no sabe quien es el remitente, por lo que no tiene a quien devolverle algo.

Se pueden o no especificar parámetros en el delegado(aunque no haya parámetros van los paréntesis igual)Al final va siempre ; porque es una sentencia declarativa no lleva codigo extra

**Orden:**

1. Lo primero que definimos es que delegados vamos a usar, si los de net framework o uno personalizado(creado por mi) Si es uno personalizado vamos a tener que definir donde lo vamos a declarar al delegado(Se declaran en cualquier archivo por fuera de una declaración de una clase o una interfaz(fuera del public class o public interfaz))
2. Tengo que determinar que clase va a ser la que defina el evento,Como se define el evento en la clase? TipoAcceso event delegado Nombre ;

**TipoAcceso** es fundamental porque va a determinar desde donde se puede ver el evento (de donde se provoca y desde donde se le puede asignar un controlador)

Para que se pueda usar en clases diferentes se hacen públicos

**Event define que estamos frente a un evento**

**delegado** Definimos cual es el delegado que se va a usar para poder apuntar al controlador asociado al evento esto genera(que delegado tengo que usar) ya que estoy definiendo como va a ser la firma del controlador y a su vez estoy indicando como tiene que ser la provocación del evento es decir que datos le tengo que pasar cuando provoque el evento

; porque a diferencia de una operación no hay codigo es una declaración

para probocar un evento recordemos que es como invocar a una operación y que si hay que pasarle parámetros están definidos por el delegado en la definición en el evento

el controlador que voy a usar tiene que coincidir con los datos que espera del delegado(parametros)

los parametros son la firma del delegado

3)Es saber donde se provoca el evento(Se puede hacer dentro de la misma clase creadora de objetos) sino se hace ahí se tiene que hacer donde existan los objetos(porque se tiene que ver)

4) es cuando creo los objetos de la clase y le asignó el controlador para el evento

obj.Evento += new delegado (Nom OperacionEscrita) ;

Net framework maneja 2 parámetros(no son obligatorios)

1. El primer parámetros es de tipo object y se reconoce como el objeto provocador del evento
2. Representa los argumentos del evento son datos extra del contexto donde se provocó el evento(NO TODOS LOS EVENTOS TIENEN ARGUMENTOS)Ejemplo click del mouse, cuando no hay argumentos extra (no hay nada para dar)se usa la clase básica EventArgs(cualquier argumento extra es derivada de EventArgs)

Ejemplo clase

Vamos a trabajar con una universidad La misma va a tener un conjunto de materias que dicta

y un conjunto de alumnos inscriptos en la universidad además los alumnos se pueden inscribir a una materia(la inscripción a la materia es el evento)

Vamos a tener nuestro propio EventArgs

Universidad (Conjunto materias (Clase y Alumnos(Clase) inscriptos materias, inscriptos universidad)

La clase hereda de eventArgs para mantener el esquema establecido por net framework

La propiedad es de solo lectura porque el argumento es para que se consulte no se modifique la ejecución del evento

objet sender va a ser el alumno que provoque el evento

el segundo parametro que reprecenta el argumento del evento va a contener una materia que es a la que se quiere inscribir el alumno

si no se hace publico no se puede ejecutar fuera del ambito de la clase,la idea es asignar un controlador por fuera de la clase del objeto porque sino seria una operacion interna