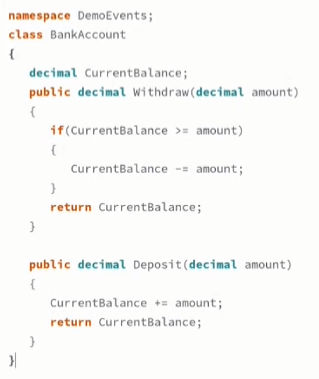
***Eventos***. Mecanismos que permiten a los objetos notificar a otros objetos cuando algo ocurre.

Por ejemplo, cuando se interactúa con un botón.

El objeto que envía o genera el evento se llama Editor o Publicador.

El que recibe o maneja el evento se denomina Suscriptor.

BankAccount debe conocer la dirección de memoria del método que va a invocar. Hay un tipo de dato que *permite guardar direcciones de métodos*. Esto es un ***delegado***. Son métodos cuyo *retorno es void*; el parámetro debe ser definido.



¿Cómo hacer para que la clase BankAccount avise a otra cuando ya no hay saldo disponible?



Eso significa que ya te dieron un valor, ya tiene su dirección de memoria, por es no es nulo.

Operadora o = new Operadora();

o.Operacion();

public class Operadora

{

BankAccount banco = new BankAccount();

public void Operacion()

{

NotificacionImportante("Aquí el parámetro le llega desde la misma clase");

banco.ErrorHandlerDelegate = NotificacionImportante;

NotificacionImportante("Aquí el parámetro le llega desde la misma clase");

banco.DepositarSaldo(1500);

banco.RetirarSaldo(800);

NotificacionImportante("PRE EXTRACCIÓN CLAVE: Aquí el parámetro le llega desde la misma clase");

banco.RetirarSaldo(900);

NotificacionImportante("POST EXTRACCIÓN CLAVE: Aquí el parámetro le llega desde la misma clase");

//ErrorHandlerDelegate es una propiedad, no un método. Solo puede almacenar direcciones de métodos que cumplan

//las especificaciones del método. Por eso es void y recibe parámetro tipo string.

//Console.WriteLine($"El saldo actual es de ${banco.currentBalance}");

}

void NotificacionImportante(string mensaje)

{

Console.WriteLine(mensaje);

}

}

public delegate void ErrorHandler(string info);

class BankAccount

{

public decimal currentBalance { get; set; }

public ErrorHandler ErrorHandlerDelegate { get; set; }

public decimal RetirarSaldo(decimal saldo)

{

if (currentBalance >= saldo)

{

currentBalance-= saldo;

if(currentBalance == 0)

{

ErrorHandlerDelegate?.Invoke("NOTIFICACION DESDE DELEGADO: El saldo actual es cero");

//ESTO Y LO QUE SIGUE SON LO MISMO:

/\*if(ErrorHandlerDelegate != null)

{

ErrorHandlerDelegate("NOTIFICACION DESDE DELEGADO: El saldo actual es cero");

}\*/

}

}

else

{

if (ErrorHandlerDelegate != null)

{

ErrorHandlerDelegate("NOTIFICACION DESDE DELEGADO: El saldo es insuficiente para realizar la extracción");

}

}

return currentBalance;

}

public decimal DepositarSaldo (decimal saldo)

{

currentBalance += saldo;

return currentBalance;

}

