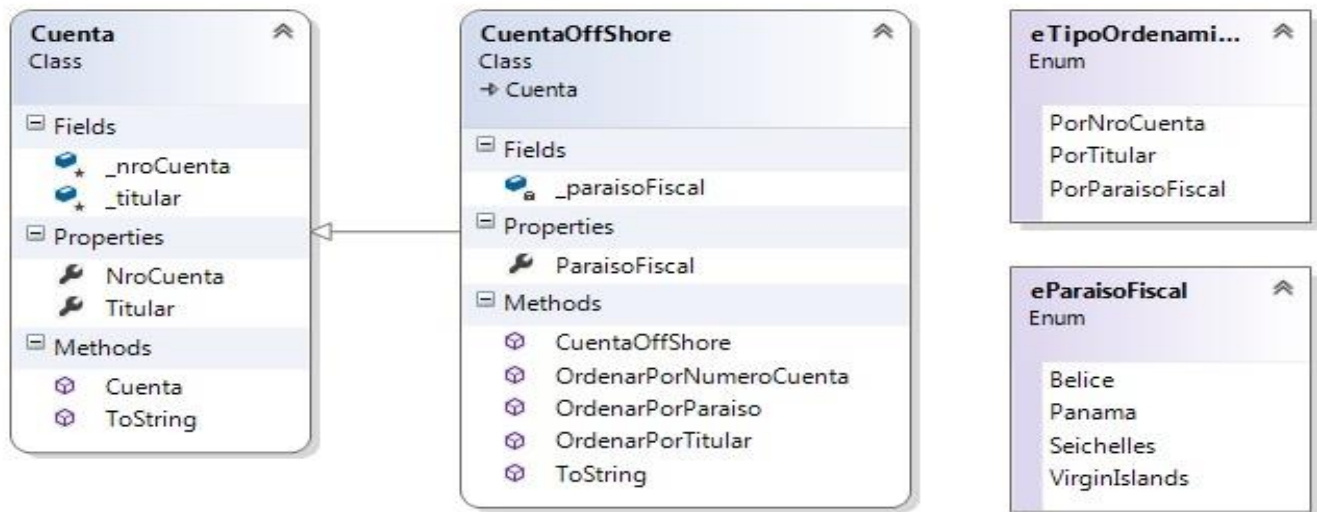


## Primer Parcial de Laboratorio de Computación III

La aplicación deberá poder hacer alta, baja y modificación de la entidad CuentaOffShore.

- 1- En un proyecto de tipo **biblioteca de clase** se debe:
  - a. Desarrollar las clases **Cuenta** y **CuentaOffshore** de acuerdo al siguiente diagrama de clases:



Respetar los constructores y métodos mostrados en el diagrama de clases, no se pueden modificar, agregar u obviar a ninguno de los miembros de las clases.

- 2- En un proyecto de tipo **WindowsForm** (iniciará por un módulo que contenga el método **Main**, con su correspondiente control de errores) se crearán tres formularios:

### FrmCuenta

Tiene dos cuadros de texto, uno para poder ingresar el nro de cuenta y el otro el titular de la misma.

El manejador asignado al evento click del botón **Aceptar** debe ser **virtual** para poder ser sobrescrito en los formularios que hereden de este.

En dicho manejador se asignará el valor **DialogResult.Ok** a la propiedad **DialogResult** del formulario.

El manejador asignado al evento click del botón **Cancelar** debe asignar el valor **DialogResult.Cancel** a la propiedad **DialogResult** del formulario.

Los TextBox de este formulario deben ser **protegido**.

The screenshot shows a Windows Form titled "frmCuenta". It contains two text boxes: "Nro Cuenta:" and "Titular:". At the bottom, there are two buttons: "Aceptar" and "Cancelar".

### **FrmCuentaOffshore**

Hereda de FrmCuenta y posee un ComboBox (**protegido**) cargado con paraísos fiscales definidos en el enumerado **eParaísoFiscal**, este control tendrá seleccionado por defecto el valor “Belice” y no se permitirá la escritura dentro del mismo.

Además este formulario tiene un atributo **privado** de tipo **CuentaOffshore** que tendrá una propiedad de **sólo lectura**, en el manejador del evento click del botón *Aceptar* se instanciará el atributo CuentaOffshore del formulario.

### **FrmPrincipal**

3- Este formulario tendrá como atributo **privado** una lista genérica de tipo **CuentaOffshore**, un menú con 4 botones y un ComboBox y un control ListBox.

El ComboBox será cargado con los valores del enumerado **eTipoDeOrdenamiento**.

Al seleccionar un nuevo criterio, los datos de la lista se ordenarán con un delegado de tipo **Comparison** y se refrescarán los datos del ListBox.

#### 4- **Botón de alta:**

Mostrará una instancia del formulario **FrmCuentaOffshore** y agregará el objeto de tipo CuentaOffshore de dicho formulario a la lista de **FrmPrincipal**. Los objetos que se agreguen serán mostrados en el ListBox.

#### 5- Manejadores:

Los manejadores de los eventos click de los botones de modificación y baja, sólo serán agregados al momento de seleccionar un elemento del **ListBox** y removidos al hacer click en cualquiera de los dos botones. Así mismo el manejador del evento **selectedIndexChanged** será quitado al seleccionar un elemento del **ListBox** y será agregado al clicar alguno de los dos botones. **Nota:** No se deben producir excepciones al manipular dichos manejadores.

#### 6- Botón de baja:

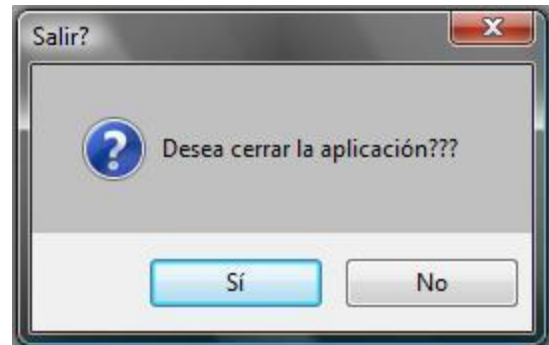
El manejador asociado al evento click se llamará “**ManejadorCentral**” e instanciará un objeto de tipo **FrmCuentaOffshore** pasándole en el constructor del formulario el objeto de tipo **CuentaOffshore** seleccionado del **ListBox**. Los datos de este objeto serán mostrados en los controles del formulario instanciado y si el usuario pulsa **Aceptar** el objeto será removido de la lista genérica de CuentasOffshore de **FrmPrincipal**. Luego se refrescará el **ListBox**.

#### 7- Botón de modificación:

El evento click del botón también tendrá asociado al manejador “**ManejadorCentral**”. Este manejador deberá saber que control lo invocó. Si el botón es el de **Modificación**, tomará al objeto del formulario **FrmOffshore** y lo asignará a la lista genérica de tipo **CuentaOffshore**, luego refrescará el **ListBox**.

#### 8- Botón salir:

Al hacer click en el botón **Salir** o en la “**X**” del formulario principal, se mostrará un mensaje **IDENTICO** al de la imagen (**NO REPETIR LINEAS DE CODIGO**), y si el usuario selecciona “**Sí**”, se cerrará la aplicación



#### 9- Control de Usuario:

Realizar un control propio en una **biblioteca de clases**, que herede de **TextBox** y que a través de una propiedad de tipo enumerado nos permita elegir si ese control va a permitir:

- SoloNumeros:** los números del “0” al “9” y la tecla de borrado (**BACKSPACE**).
- SoloLetras:** las letras de la “a” a la “z” tanto en mayúscula como en minúscula y la tecla de borrado (**BACKSPACE**).

Cambiar los cuadros de texto de Nro de cuenta y titular (en **FrmCuentaOffshore**), por el control de usuario y configurarlos adecuadamente.

10- Se creará un nuevo formulario llamado **FrmMostrar** que posea solamente un **ListBox**. La instancia de **FrmMostar** deberá tener la propiedad **FormBorderStyle=“NONE”**. Este formulario tendrá un sólo método **privado** llamado “**ActualizarListados**” que recibirá una lista genérica de tipo “**CuentaOffshore**” como parámetro. Al invocar a este método se actualizará el **ListBox** (**LstMostrar**) con los titulares de las cuentas de la lista genérica de tipo **CuentaOffshore**) en color **rojo**.

- Crear un tipo de **delegado** para poder invocar al método “**ActualizarListados**” en un **CodeFile**.
- La variable se instanciará en **FrmPrincipal**, en la opción de menú ‘Mostrar Listado’, apuntando al método de la instancia de **FrmMostrar**.
- Invocar al delegado al agregar, modificar o eliminar un elemento de la lista de Cuentas.

Tiempo máximo para resolver el examen: 120 minutos. Transcurrido ese tiempo las máquinas serán bloqueadas.