Unidad

6

DIPLOMATURA EN PROGRAMACION .NET

Ejercicios

Capítulo 12

Interfaces Gráficas

Ejercicio 1

Para los ejercicios del capítulo, abrir la solución cuyo nombre es Capitulo12.

Modificar la clase Cliente

1. Rescribir el método ToString para que retorne una cadena formada con el apellido, una coma, un espacio en blanco y el primer nombre. Esto permite que si se pasa a un control el objeto, este pueda utilizar la cadena retornada por el método para presentarla en la salida del mismo.

Modificar la clase CajaDeAhorro

2. Rescribir el método ToString para que retorne la cadena "Caja de Ahorros" Esto permite que si se pasa a un control el objeto, este pueda utilizar la cadena retornada por el método para presentarla en la salida del mismo.

Modificar la clase CuentaCorriente

3. Rescribir el método ToString para que retorne la cadena "Cuenta Corriente". Esto permite que si se pasa a un control el objeto, este pueda utilizar la cadena retornada por el método para presentarla en la salida del mismo.

Modificar la clase ReporteCliente

4. Renombrar el método GenerarReporte con el nombre GuardarReporteEnDisco
5. Generar un nuevo método estático llamado Reporte el cual retornará una cadena con el formato de salida del reporte.
6. Para construir el reporte en una cadena, declarar un objeto del tipo StringBuilder sobre el cual se irá modificando la cadena a medida que se genere el reporte.
7. Sustituir todas las salidas por consola por invocaciones al método Append del StringBuilder. En el caso que se quiera agregar una nueva línea, utilizar Environment.NewLine para que el retorno de carro se interprete correctamente en los controles gráficos.
8. Una vez armado el reporte, utilizar el método ToString para retornar la cadena que contiene el StringBuilder

Modificar el formulario del proyecto

9. Agregar en el constructor cada cliente del banco al ComboBox cbxClientes utilizando en método Add y la instancia del cliente en particular que se agregue

Modificar el manejador de evento Form1_Load del formulario

10. Agregar los manejadores de eventos al maskedTextBox1 para los eventos MaskInputRejected y KeyDown. Utilizar el operador += en C# o la instrucción AddHandler en VB.

Modificar el manejador de evento cbxClientes_SelectedIndexChanged del formulario

11. Limpiar de Items el cbxCuentas para volverlo a llenar según las cuentas de cada cliente

12. Seleccionar el ítem 0 del cbxOperacion para asegurarse que siempre haya una cuenta seleccionada y no tener que validar la selección
13. Obtener el objeto del tipo Cuenta seleccionado en el ComboBox cbxCuentas que va a ser el primero de la lista porque así se lo indicó en la instrucción anterior. Para ello recuperar del control el objeto y convertirlo para asignarlo a una referencia del tipo Cuenta. Usar como ejemplo la conversión realizada en el objeto del tipo Cliente que es la primera instrucción del manejador.
14. Asignar el balance de la cuenta al txtbBalance para mantener la información del balance de la cuenta actualizada

Modificar el manejador de evento btRealizarOperacion_Click del formulario

15. Obtener el objeto del tipo Cliente seleccionado en el ComboBox cbxClientes. Para ello recuperar del control el objeto y convertirlo para asignarlo a una referencia del tipo Cliente.
16. Validar que la referencia al objeto del tipo Cuenta no sea nula para asegurarse que hay una seleccionada en la interfaz gráfica antes de realizar la operación. En caso de ocurrir, usar MessageBox.Show para mostrar el mensaje de error
17. Basándose en el índice del ComboBox cbxOperacion se sabe que el elemento 0 es la operación Retira, sino es Deposita. Usar la referencia al objeto de tipo cuenta para realizar la operación apropiada en base a la cantidad obtenida del maskedTextBox1. Recordar que la operación Retira puede lanzar una excepción del tipo ExcepcionSobregiro y que se debe controlar. En caso de ocurrir, usar MessageBox.Show para mostrar el mensaje de error.
18. Asignar el balance de la cuenta al txtbBalance para mantener la información del balance de la cuenta actualizada.
19. Actualizar el reporte que se muestra luego de cada operación en txtBReporte. Para ello, obtener la cadena que retorna ReporteCliente.Reporte() y asignarla al TextBox que fue configurado para que tenga múltiples líneas.

Modificar el manejador de evento cbxCuentas_SelectedIndexChanged del formulario

20. Recuperar el objeto del tipo Cuenta cada vez que cambia la selección del ComboBox para poder recuperar el balance que corresponde a dicha cuenta
21. Asignar el balance de la cuenta al txtbBalance para mantener la información del balance de la cuenta actualizada.

Modificar el manejador de evento btGuardarReporte_Click del formulario

22. Guardar el reporte en disco invocando al método ReporteCliente.GuardarReporteEnDisco()

Compilar y ejecutar la aplicación

23. Verificar que no haya errores de compilación
24. Ejecutar la aplicación y verificar que se obtiene un comportamiento similar al mostrado en la siguiente figura