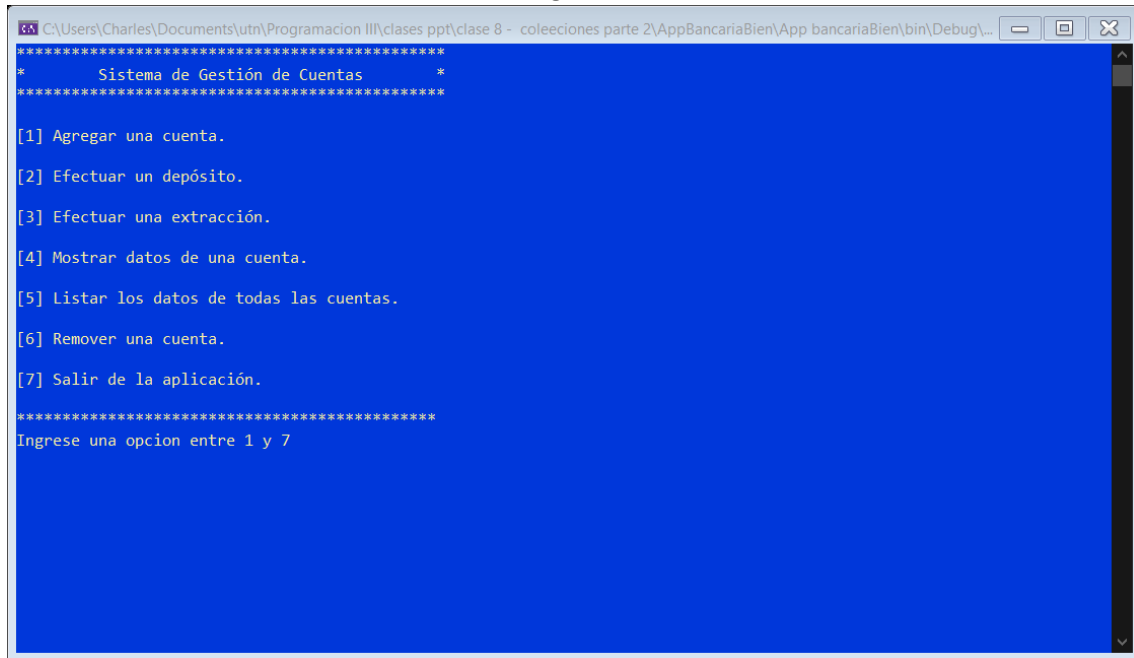


Ejercicio – App bancaria 2.0

Para la clase Cuenta desarrollada anteriormente, vamos a realizar una app bancaria para gestionarlas.

La misma deberá mostrar un menú como el siguiente:



```
C:\Users\Charles\Documents\utn\Programacion III\clases ppt\clase 8 - colecciones parte 2\AppBancariaBien\App bancariaBien\bin\Debug\...
*****
*           Sistema de Gestión de Cuentas           *
*****

[1] Agregar una cuenta.
[2] Efectuar un depósito.
[3] Efectuar una extracción.
[4] Mostrar datos de una cuenta.
[5] Listar los datos de todas las cuentas.
[6] Remover una cuenta.
[7] Salir de la aplicación.

*****
Ingrese una opcion entre 1 y 7
```

Al ingresar cada una de las opciones el programa deberá pedir los datos necesarios para realizar cada una de las acciones:

1. Pedir CBU, nombre del cliente y saldo inicial para crear una cuenta y agregarla a un arraylist.
2. Pedir el CBU para buscar la cuenta en un arraylist, luego pedir el monto para realizar el depósito. En caso de que no se pueda realizar el depósito se deberá avisar al usuario.
3. Pedir el CBU para buscar la cuenta en un arraylist. Luego pedir el monto para realizar la extracción. En caso de que no se pueda realizar la extracción se le deberá avisar al usuario.
4. Pedir el CBU para buscar la cuenta en un arraylist. Luego se deberán mostrar los datos del CBU.
5. Listar los datos de todas las cuentas guardadas en el arraylist.
6. Pedir un CBU para buscar la cuenta. Acto seguido eliminarla del arraylist.
7. Mostrar un mensaje de despedida y luego cerrar el programa.

Para realizar este programa se deberán implementar las siguientes clases:

- Cuenta: Deberá tener todos los datos de la cuenta bancaria y métodos que ya vimos
- ControladorCuentas: Debera tener como atributo el arraylist o List de cuentas y controle la búsqueda dentro del arraylist como toda la lógica del programa (debe tener un método para cada una de las opciones del menú).
- Interfaz: Debe controlar la interacción del usuario y el programa. Debe mostrar los mensajes y permitir la entrada de datos.
- Program o Principal: En esta clase estará el main que permite que llamara a las otras clases según necesite. Por ejemplo, acá se llamara al método que muestre el menú y

devuelva las entradas y según lo que ingrese el usuario se llamara a algún método de la clase controladora.

Link de la solución:

<https://drive.google.com/file/d/1pDIFZJ7VidsuT5rXGdwPPBzUIUmdgg93/view?usp=sharing>

Parte 2

Seleccionar cualquiera de las clases realizadas. Realizar un programa que pueda gestionar varios elementos de la misma tal cual se hizo para la clase cuenta. Tener en cuenta la separación en clases para hacerlo más sencillo.

Parte 3

Realizar a través del uso de HashTables un programa que funcione como un diccionario de términos técnicos que funcione de la siguiente manera:

Al iniciar el programa se deberá mostrar un menú con tres opciones:

1. Buscar una palabra: Deberá permitir al usuario ingresar una palabra. Si existe en el hashtable deberá devolver la definición de la misma.
2. Agregar una definición: Deberá permitir al usuario ingresar una palabra. Si existe en el hashtable deberá devolver un error. De no existir deberá permitir el ingreso de una definición para el diccionario.
3. Modificar una definición: Deberá permitir al usuario ingresar una palabra. Si existe en el hashtable deberá permitir al usuario ingresar una nueva definición para actualizarla. De lo contrario deberá mostrar un error.
4. Eliminar una definición: Deberá permitir al usuario ingresar una palabra. Si existe en el hashtable deberá eliminar la definición y la palabra del mismo. De lo contrario deberá mostrar un error.
5. Salir: Deberá cerrar el programa.

Las opciones se deben mostrar en loop hasta que el usuario elija la última opción que cierra el programa.