**TAREA BANCO**

Se trata de manejar una relación muchos a muchos de forma que un cliente puede ser titular de muchas cuentas y una cuenta puede tener muchos titulares.

el profesor te entrega las clases Cliente, Cuenta y MovimientoException. Muy simplificadas respecto a ejercicios anteriores, por ejemplo, el número de cuenta es un string cualquiera y lo mismo el dni. No hay control de formato de ambos.

Debes escribir una clase BaseDeDatos y una clase InterfaceConsola. La clase BaseDeDatos se encarga de manipular los datos (inserciones, borrados, …) de forma que sus métodos harán el trabajo que permite separar el interface del usuario del tratamiento de los datos de tal forma que ahora haremos una interface de consola pero más adelante podemos hacer un interfaz swing. Además para almacenar la relación de clientes y cuentas que va creando la aplicación vamos a incorporar a nuestra clase dos mapas

* un hashmap clientes
* un hashmap cuentas

*public class BaseDeDatos {*

*Map<DNI,Cliente> clientes= new HashMap<>();*

*Map<String,Cuenta> cuentas= new HashMap<>();*

*métodos de clase BaseDeDatos*

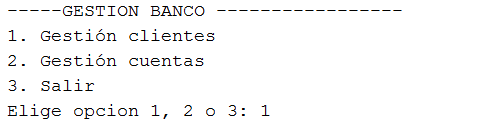
*……….*

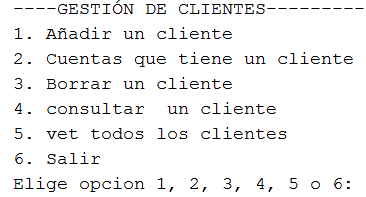
*}*

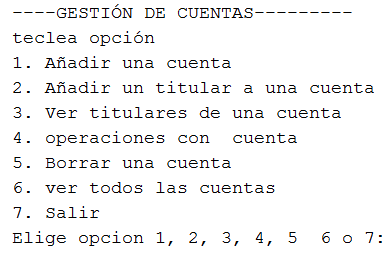
Observa que para reflejar la relación muchos a muchos, ampliamos la clase Cuenta y la clase Cliente:

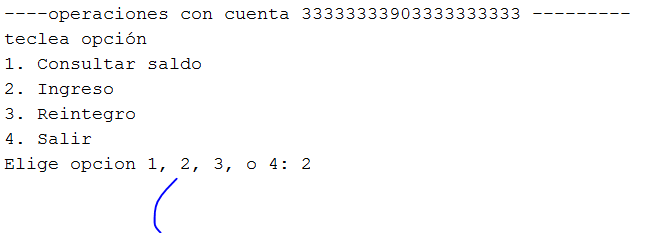
* Cada cliente tiene dentro un set de strings que representa a los ccc de las cuentas bancarias de las que es titular
* Cada cuenta tiene dentro un set de strings que representa a los nif de los clientes

Una posible estructura de menús podría ser:

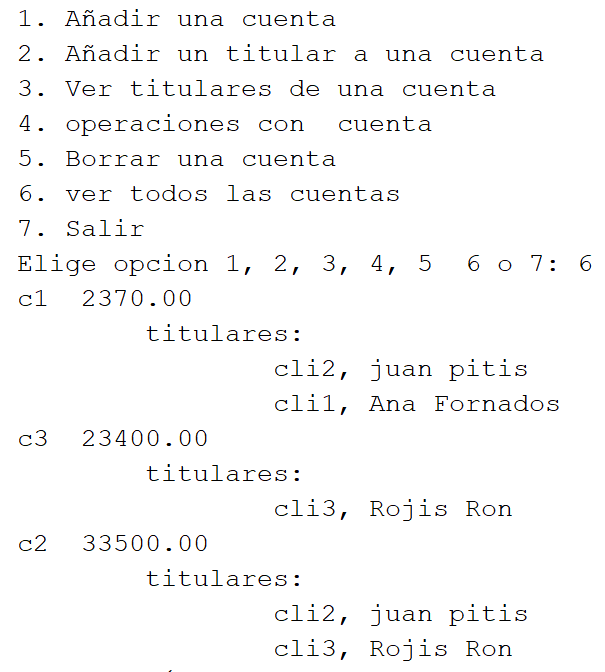


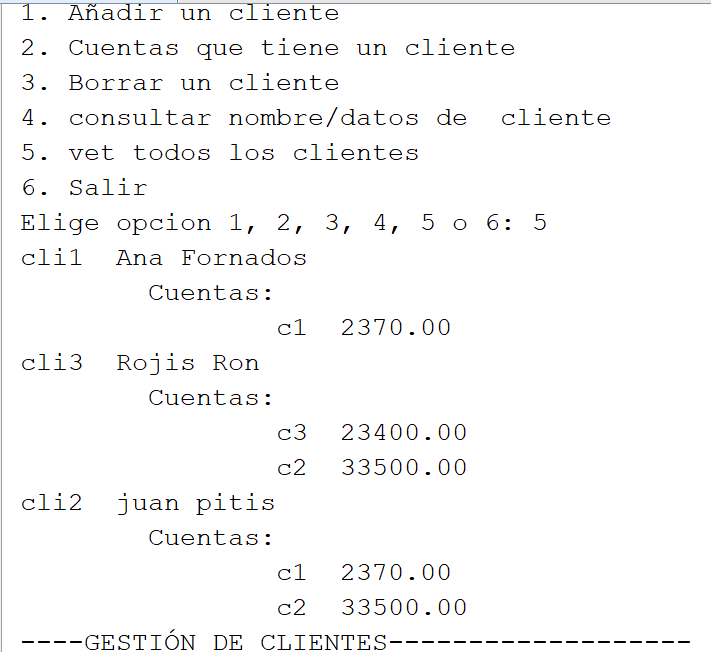






En el ejemplo usamos tres cuentas c1, c2 y c3 y tres clientes cli1, cli2 y cli3. Si listamos por cuentas y por clientes se aprecia la relación muchos a muchos.





Algunas cosas que deben ocurrir:

* “ver todas las cuentas” debe listar las cuentas ordenadas ascendentemente por saldo.
* la opción “ver todos los clientes” debe listar los clientes ordenados por nombre ascendentemente
* cada vez que se realiza una inserción o un borrado de una cuenta tenemos que actualizar la relación muchos a muchos
* sólo se puede borrar una cuenta si su saldo es 0
* sólo se puede borrar un cliente si no tiene cuentas.