Tarea 7. Programación.

Xerach E. Casanova Cabrera – DAW A.

Informe.

Para la realización de esta aplicación he dividido el programa en las siguientes clases:

Principal:

Es la clase que muestra el menú y sus opciones, así como la encargada de recibir toda la información y mostrarla por pantalla. En ella se instancia un objeto de tipo banco, el cual será el encargado de manejar el resto de clases (cuentas bancarias y personas), a través de sus métodos.

Banco:

Es la clase encargada de manejar las distintas cuentas bancarias, así como los clientes, que son objetos contenidos dentro de las distintas clases de cuentas bancarias. Contiene un array de tipo CuentaBancaria en el cual se van guardando las instancias de las distintas clases, haciendo uso del polimorfismo y la ligadura dinámica.

En esta clase podemos encontrar los métodos principales que manejan las cuentas:

- Crear cuentas bancarias
- Listar cuentas bancarias
- Mostrar información de una cuenta
- Ingresar efectivo
- Retirar efectivo
- Mostrar saldo de cuenta bancaria

Pero además tiene otros métodos de utilidad para el funcionamiento del programa como:

- Obtener números de cuenta de un usuario en concreto.
- Obtener la cantidad de cuentas que tiene un usuario asignadas.
- Método para saber si existe o no una cuenta bancaria.
- Obtener los datos de un cliente a partir de su número de cuenta.
- Obtener los datos de un cliente a partir de su DNI.
- Obtener el total de la comisión e interés de una retirada de dinero en una cuenta en negativo.

La combinación de todos estos métodos con los métodos principales, genera toda la interfaz y el funcionamiento interno del programa.

CuentaBancaria

Es la superclase abstracta de la cual heredan todos los demás tipos de cuenta. Contiene la estructura básica de todos los tipos de cuentas bancarias. Atributos generales de todas ellas:

- Titular
- Saldo
- Número de cuenta

Métodos getter y setter y un método que pertenece a la interfaz Imprimible, la cual implementan todos los tipos de cuentas bancarias y sirve para devolver la información de la cuenta bancaria en una variable de tipo String.

CuentaCorriente

Es una clase abstracta, ya que su uso se limita a dar una estructura básica a los dos tipos de cuenta que heredan de ella: CuentaCorrientePersonal y CuentaCorrienteEmpresa. No será necesaria la instanciación de objetos de tipo CuentaCorriente, por eso, aunque no haya generado ningún método abstracto, la clase si lo es.

Contiene todos los atributos heredados de CuentaBancaria y además un atributo propio que es un array que almacena entidades de cobro para cada cuenta bancaria.

Para este atributo tenemos dos métodos que en una futura ampliación del programa podremos usar para añadir o eliminar entidades de cada cuenta bancaria.

CuentaCorrientePersonal

Es una clase que hereda de CuentaCorriente, que a su vez hereda de CuentaBancaria.

La clase tiene dos constructores, uno que obtiene como parámetros los atributos de sus super clases y que le da un valor por defecto a los atributos propios y otro constructor que se puede utilizar en un futuro para poder crear directamente objetos de tipo de cuenta personal insertando los valores de los atributos propios desde el mismo constructor.

Cuenta con todos los atributos heredados, más un atributo llamado comisionAnual, el cual tiene sus métodos getter y setter y además un método para ejecutar la comisión anual sobre el saldo actual de la cuenta.

Además cuenta con la implementación del método perteneciente a la clase Imprimible para guardar los datos de la cuenta en una cadena.

CuentaCorrienteEmpresa

Es una clase que hereda de CuentaCorriente, que a su vez hereda de CuentaBancaria.

La clase tiene dos constructores, uno que obtiene como parámetros los atributos de sus super clases y que le da un valor por defecto a los atributos propios y otro constructor que se puede utilizar en un futuro para poder crear directamente objetos de tipo de cuenta empresa insertando los valores de los atributos propios desde el mismo constructor.

Cuenta con todos los atributos heredados, más sus propios atributos, que son:

- Máximo descubierto en cuenta.
- Tipo de interés
- Comisión

Todos tienen sus métodos getter y setter y además un método que calcula el tipo de interés y comisión si se hace un movimiento en cuenta con saldo en negativo, o que se quede en negativo después de realizar el movimiento. También limita el máximo descubierto a partir del propio atributo.

Además cuenta con la implementación del método perteneciente a la clase Imprimible para guardar los datos de la cuenta en una cadena.

CuentaAhorro

Es una clase que hereda de CuentaBancaria, tiene los atributos de su clase padre y además un atributo para asignar el tipo de interés remunerado anual.

La clase tiene dos constructores, uno que obtiene como parámetros los atributos de sus super clases y que le da un valor por defecto a los atributos propios y otro constructor que se puede utilizar en un futuro para poder crear directamente objetos de tipo de cuenta ahorro insertando los valores de los atributos propios desde el mismo constructor.

Cuenta también con los métodos getter y setter de interés remunerado y un método que servirá en un futuro para aplicar el interés remunerado sobre el saldo de la cuenta.

Además cuenta con la implementación del método perteneciente a la clase Imprimible para guardar los datos de la cuenta en una cadena.

Persona

Es una clase que genera objetos de tipo persona y tiene como atributos el nombre, los apellidos y el DNI del titular.

Cuenta con un constructor que genera copias de objetos de tipo personas, para poder usarlo en la clase Banco como atributo sin perder la norma de la ocultación.

Además cuenta con la implementación del método perteneciente a la clase Imprimible para guardar los datos de la persona en una cadena.

Interfaz Imprimible

Cuenta con un método de tipo String que tendrán que implementar todas las clases que utilicen la interfaz.

Clase ValidarDatos

Cuenta con algunos métodos que servirán para validar los datos que se van introduciendo en el programa, como el DNI o el número de cuenta.