



Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Programación Avanzada
Proyecto 03 – A - Herencia
Programación Orientada a Objetos

El objetivo del siguiente proyecto es realizar una aplicación que se implemente en programación orientada a objetos, haciendo uso del lenguaje Java, los conceptos de manejo de archivos, clases y objetos, cadenas de caracteres, colecciones, entre otros. Se hará énfasis en Herencia, Polimorfismo y Clases Abstractas.

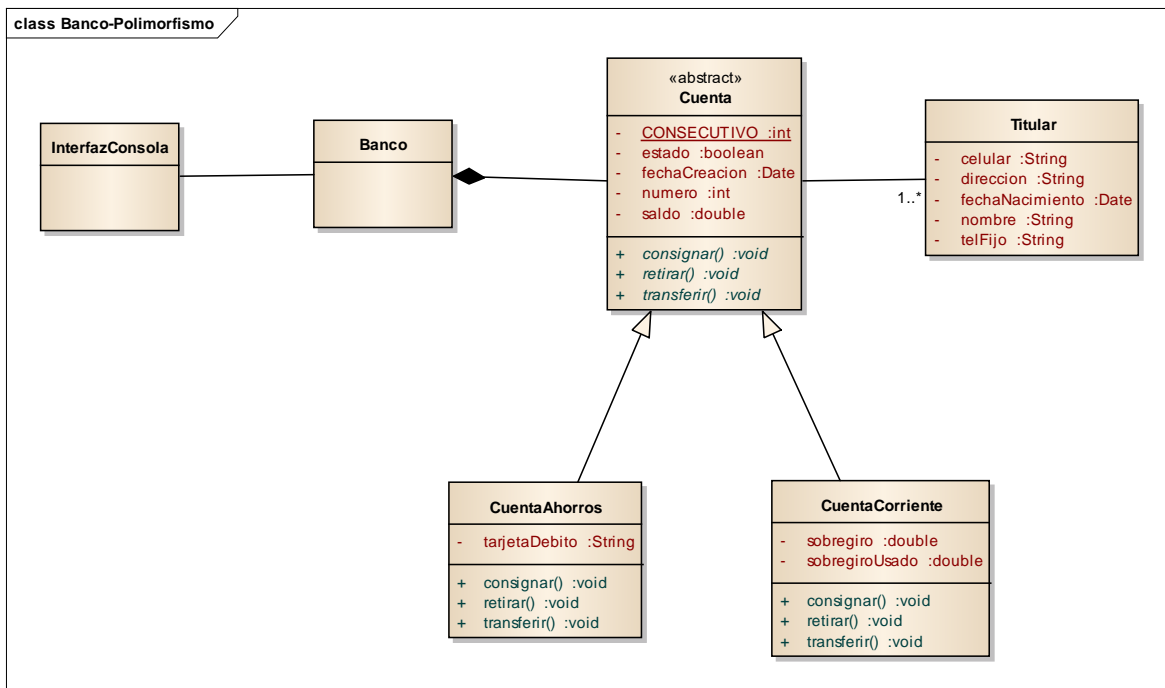
El banco estudiantil JaviBank desea implementar un sistema para el manejo de sus clientes (titulares) y cuentas.

El banco se considera como un conjunto de cuentas. Una cuenta tiene como atributos: saldo, estado, fecha de creación (dada como una cadena aaaa-mm-dd), número de Cuenta y titulares. Con el fin de evitar duplicidad debe idear un mecanismo que haga que el número de cuenta se maneje a través de un atributo de clase (static) llamado "CONSECUTIVO" que se actualiza cada vez que se crea una cuenta.

Una Cuenta puede tener varios Titulares. De cada Titular nos interesa conocer su nombre, fecha de nacimiento (dada como una cadena aaaa-mm-dd), teléfono fijo, teléfono celular y dirección.

Ahora la Cuenta tiene dos subclases: CuentaAhorros y CuentaCorriente. Tienen todos los atributos de Cuenta. Sin embargo, CuentaAhorros tiene un atributo que es la TarjetaDebito (un número) y CuentaCorriente tiene un sobregiro (que es el autorizado por el Banco) y sobregiro usado que es lo que puede utilizar y su límite máximo es el sobregiro autorizado.

En resumen, para este proyecto, el Banco tiene una colección de Cuentas y cada Cuenta tiene una colección de Titulares.



El sistema debe ofrecer una interfaz de texto (consola) llamada InterfazConsola que provea un menú de opciones y tenga una única instancia de JaviBank (de la clase Banco). Dicho banco debe proveer los servicios que se describen a continuación:

1. Crear cuentas y titulares: para esto, el sistema solicita al empleado bancario el archivo de texto "cuentas.txt" con el siguiente formato:

```

#Cuenta
#Saldo-----Tipo-----Tarjeta/Sobregiro
25000-----ahorros-----123456
#Titulares
#nombre-----fecha Nacimiento----TelFijo-----Celular---Direccion
Angela Carrillo      *1974-05-09      *222      *320      *Cra 20
Hernando Hurtado     *1970-06-29      *333      *321      *Calle 5
0
#Cuenta
#Saldo-----Tipo-----Tarjeta/Sobregiro
1000000-----corriente-----75000
#Titulares
#nombre-----fecha Nacimiento----TelFijo-----Celular---Direccion
Sneyder Sanchez      *1998-04-20      *111      *341      *Cra 50
Julián Méndez        *2000-04-25      *111      *348      *Cra 50
Sara De la Rosa      *2010-12-13      *111      *312      *Cra 50
0
#Cuenta
#Saldo-----Tipo-----Tarjeta/Sobregiro
258300-----ahorros-----222333
  
```

```
#Titulares
#nombre-----fecha Nacimiento----TelFijo-----Celular---Direccion
Estefania De la Rosa *2009-01-02      *400          *310          *C11e 58
0
#FIN
```

Se considera que su estado debe comenzar como "ACTIVA". La fecha de creación de la cuenta es la de hoy. Una vez se crea la cuenta, esta queda al final de la colección. Note que se adiciona información sobre el tipo de cuenta (ahorros o corriente) y en caso de ser corriente se asigna como su sobregiro. En caso de ser de ahorros se asigna como el número de la tarjeta débito. Si la cuenta es corriente, se supone que el sobregiro Usado es \$0 para empezar.

2. Consignar en una cuenta: dado el número y el monto a consignar. Si la cuenta es de ahorros, se aumenta el saldo con el monto. Si el saldo es \$50000 y el monto a consignar es \$2500, el nuevo saldo es \$52500.
Si la cuenta es corriente, primero se debe disminuir el sobregiro Usado con el monto a consignar. Si el monto es mayor al sobregiro Usado, se debe dejar en \$0 el sobregiro Usado y se aumenta el saldo con la diferencia. Si se tiene un sobregiroUsado en \$5.000 y el monto a consignar es \$7.000, el sobregiro Usado debe quedar en \$0 y el nuevo saldo es de \$2.000.
3. Retirar de la cuenta: dado el número y el monto a retirar. Si la cuenta es de ahorros, el monto máximo a retirar es el saldo. Si esto sucede, se disminuye el saldo con el monto. Si el saldo es \$50000 y el monto a retirar es \$2500, el nuevo saldo es \$47500.
Si la cuenta es corriente, el monto máximo a retirar es el saldo+sobregiro-sobregiroUsado. Si el monto a retirar es inferior al saldo, sólo se disminuye el saldo con el monto. Si es superior, primero se debe dejar en \$0 el saldo y luego se aumenta el sobregiro Usado con la diferencia entre el monto a retirar y el saldo. Si se tiene un saldo de \$7.000 y el monto a retirar es \$9.000, el sobregiro Usado debe quedar en \$2000 y el nuevo saldo es de \$0.
4. Transferir de una cuenta a otra: dado el número de la cuenta origen, el número de la cuenta destino y el monto. Se hace un retiro de la cuenta origen y una consignación a la cuenta destino.
5. Dado el nombre de un Titular, mostrar cuántas cuentas de ahorro tiene y cuántas cuentas corrientes, con su saldo y al final mostrar el saldo total.
6. Dado un año, mostrar todos los titulares que nacieron después de este año.
7. Generar tanto por consola como en un archivo texto llamado "sobregiradas.txt" donde se muestre toda la información de las cuentas corrientes que tienen un sobregiro Usado superior a \$0.
8. Serializar: crear un archivo JaviBank.dat donde se guarde el objeto Banco (con sus cuentas y titulares).
9. Deserializar: cargar a memoria principal de un archivo JaviBank.dat donde se guarde el objeto Banco (con sus cuentas y titulares).

El Proyecto debe ser realizado por máximo dos personas, será entregado el jueves de la semana 17 (27 de mayo de 2021 hasta las 9:00am). Revise que su programa realice cada una de las funcionalidades que se le solicitan. El proyecto Entrega 3 debe estar sin errores. Los programas que no compilen tendrán como nota 0.0. Se le podrá solicitar a uno o los dos integrantes explicación ampliada del programa y/o modificación de alguna funcionalidad adicional. La nota será asignada para las dos personas.

1. Sólo se permite el uso de List con su implementación ArrayList para el manejo de colecciones; no se puede utilizar arreglos o matrices ni contenedores que manejen memoria estática.
2. Es obligatorio el uso de archivos
3. No se debe "quemar código" para simular la entrada/salida de datos
4. No se permite hacer Entrada/Salida de Datos en clases diferentes a InterfazConsola.
5. Es obligatorio el uso de Herencia.

Se tendrán dos notas:

- 1.- Nota del programa donde se revisará la codificación y la presentación del código. Se deben generar los comentarios a través del programa.
- 2.- Nota de Sustentación. Si la sustentación es exitosa el proyecto se calificará sobre el 100% en caso contrario se calificará sobre el 30%.