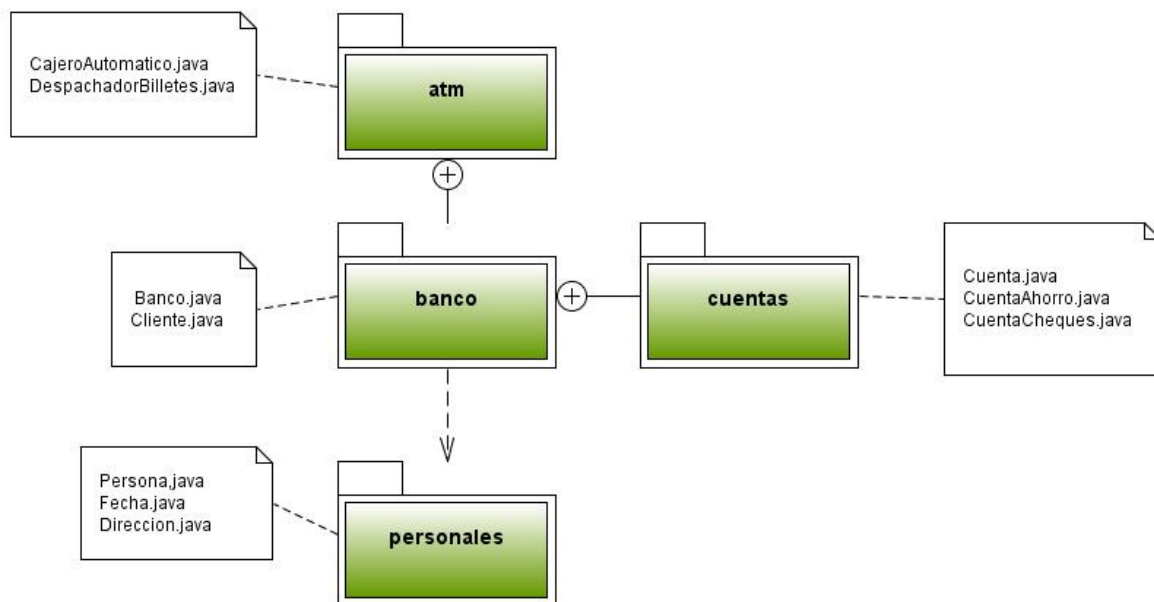


PROYECTO 2 DE PROGRAMACIÓN OO

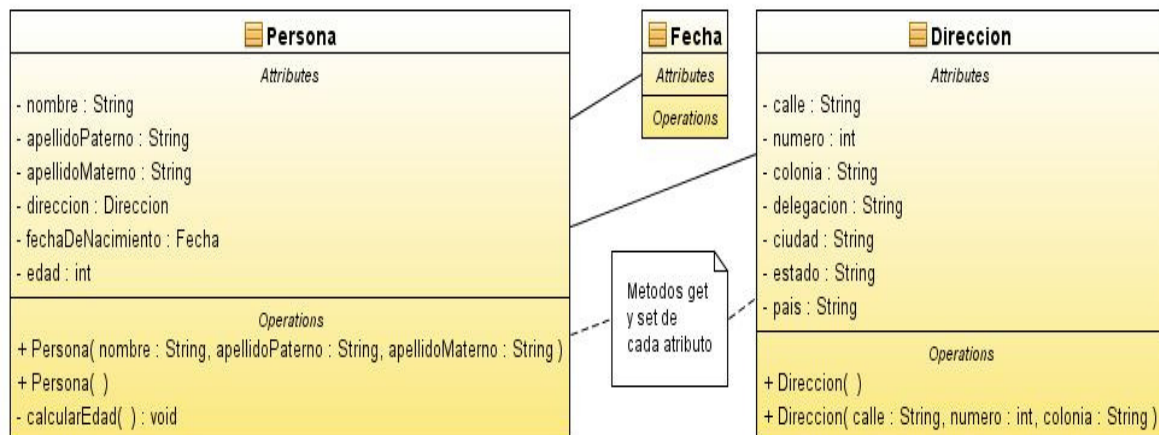
Tema: Clases, herencia, polimorfismo, interfaz gráfica

Desarrolla las siguientes aplicaciones:

1. Crear un directorio con el nombre de proyectoATM.
2. Dentro de proyectoATM crear dos directorios uno con el nombre de classes y otro con el nombre de src. Dentro de src se colocarán todos los archivos con extensión java que se generen. En el directorio classes se colocarán los archivos con extensión class que genere el compilador
3. Dentro del directorio proyectoATM\src los siguientes paquetes:

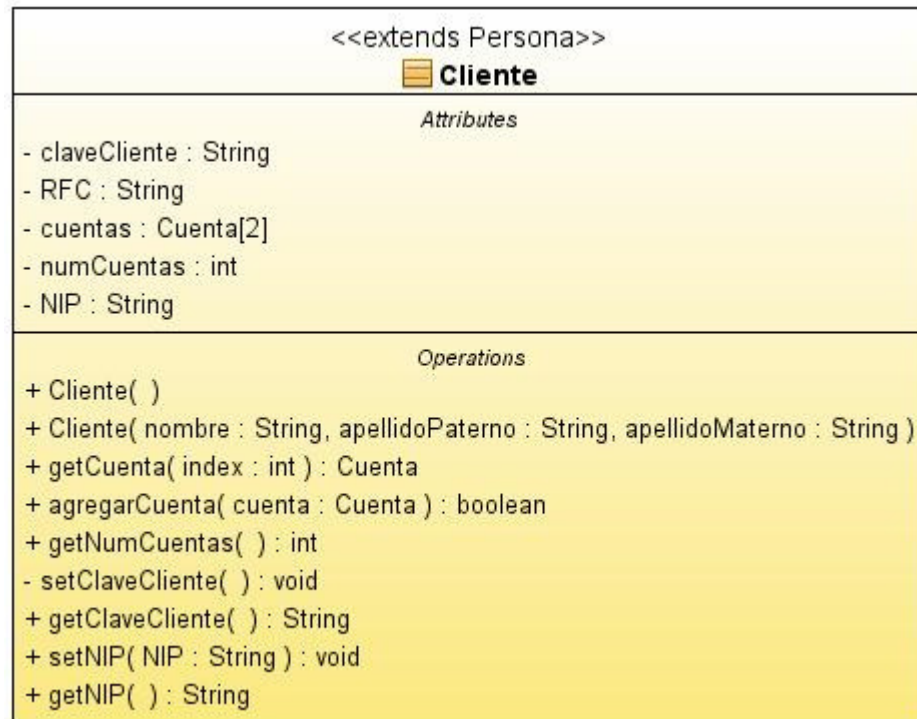


4. Coloca en el paquete personales las clases de la Practica 2 y añade la clase Direccion definida en el siguiente diagrama UML:

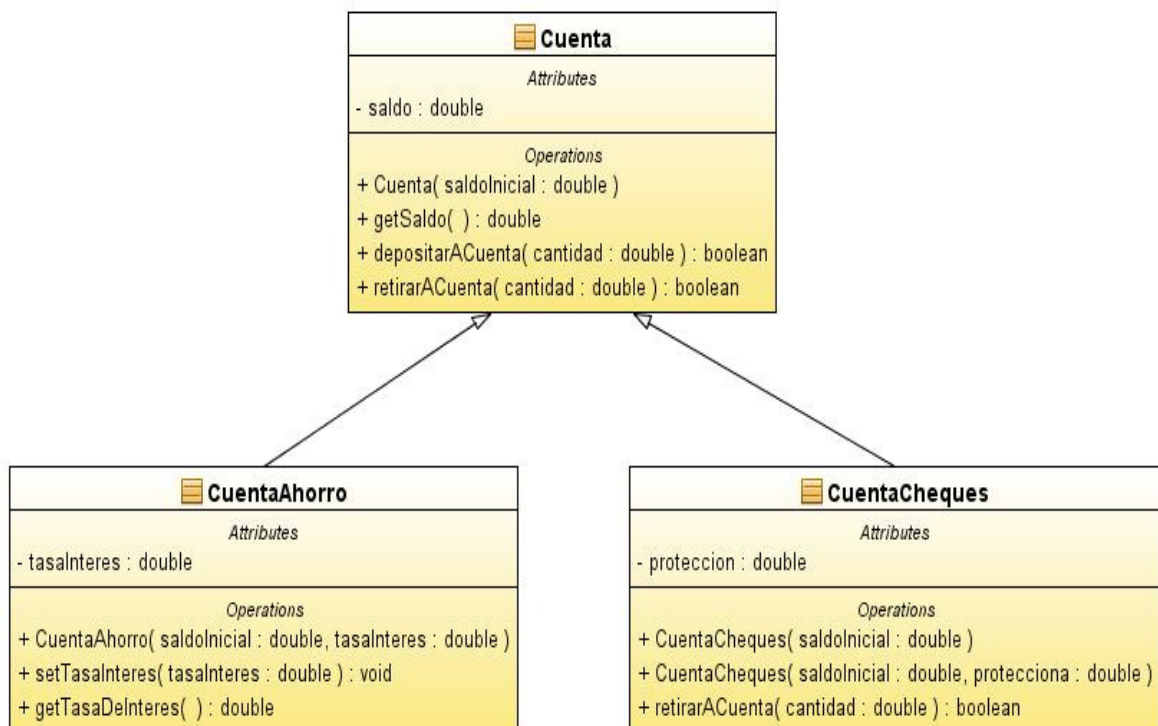




5. Crear en banco la clase Cliente con la siguiente especificación:



6. Crear las siguientes clases en el paquete cuentas:





7. Dentro del paquete banco, crear la clase Banco que maneja el nombre del banco y un arreglo de Clientes (Crear el diagrama UML de esta clase)
8. Dentro del paquete atm crear las siguientes clases:

a) Clase DespachadorBilletes

Clase que proporciona el efectivo en al CajeroAutomatico. Revisar la clase DespachadorDeJugos proyecto anterior. (Crear el diagrama UML de esta clase)

b) Clase CajeroAutomatico

Clase que recibe un objeto de tipo Banco y que para su funcionamiento pregunta el nombre del banco y el NIP del cliente. Posee un arreglo de despachadores de billetes con las siguientes nominaciones: 500, 200, 100, 50, 20. (Crear el diagrama UML de esta clase)

Su aplicación debe llevar a cabo lo siguiente:

1. Ingresar NIP
2. Cambio de NIP
3. Consulta de Saldo
4. Abono en Cuenta
5. Retiro en Cuenta
6. Pago de Servicios
7. Salir

9. Su aplicación debe contemplar las siguientes funcionalidades generales:

1. Administracion de Clientes
2. ATM
3. Salir

Para la administración de clientes se debe presentar el siguiente menú:

1. Alta Banco
2. Alta Cliente
3. Alta Cuenta
4. Salir

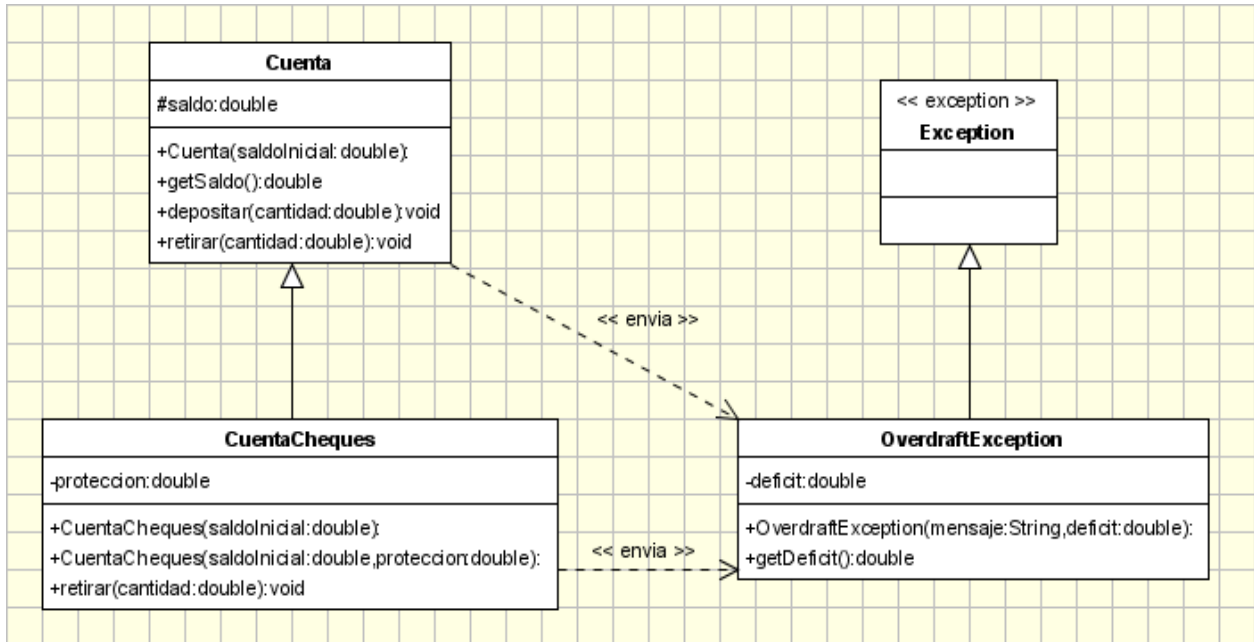
- Para la segunda opción, preguntar nombre, apellidos, dirección.
- Para la tercera opción, preguntar o presentar un menú con las opciones de cuenta de ahorro o de cheques y en cada caso pedir los datos necesarios para crear dichas cuentas

NOTA: Manejar a los clientes a través de clases del API de Collection



Manejo de Excepciones

1. De la aplicación de Sistema de Banco, añade una clase OverdraftException para manejar la insuficiencia de fondos al retirar en la cuenta a través de las excepciones.



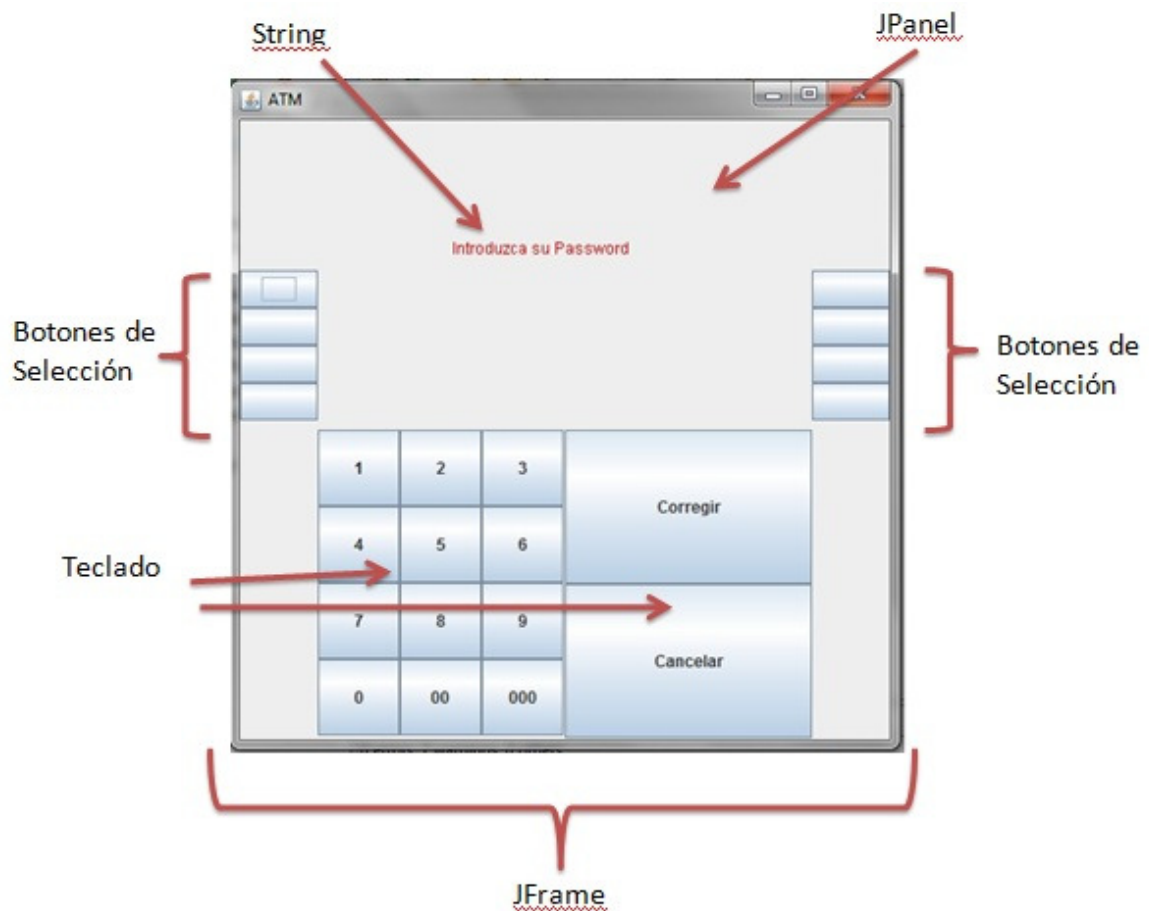
- a) Crear una clase publica, llamada **OverdraftException** en el paquete banco. Esta clase hereda de la clase **Exception**
- b) Añade un atributo privado, llamado **déficit** que es de tipo doble. Añade un método de acceso llamado **getDeficit**.
- c) Añade un constructor que toma dos argumentos: un mensaje y un déficit. Pasa el mensaje al constructor del padre. Con el parámetro **déficit** inicializa el atributo **déficit**.
- d) Modifica el método **retirar** de la clase **Cuenta** de tal manera que no regrese nada y envíe la excepción **OverdraftException**, especificando un mensaje "Insuficientes fondos" y el déficit (la cantidad requerida sustraída del actual saldo).
- e) Modifica el método **retirar** de la clase **CuentaCheques** de tal manera que no regrese nada y envíe la excepción **OverdraftException**. Existen dos casos para manejar esta excepción:
 - Primero, hay un déficit sin protección. Utiliza el mensaje "No hay protección" y manda el déficit
 - Segundo, la protección es insuficiente para cubrir el déficit. Utiliza el mensaje "Insuficientes fondos en la protección" y manda el déficit.
- f) Añada en el Sistema de Banco y Cajero Automático los mensajes de error para el usuario en forma de Excepciones. Utilice la parte inferior de sus pantallas para presentar los mensajes de error

Interfaz Gráfica

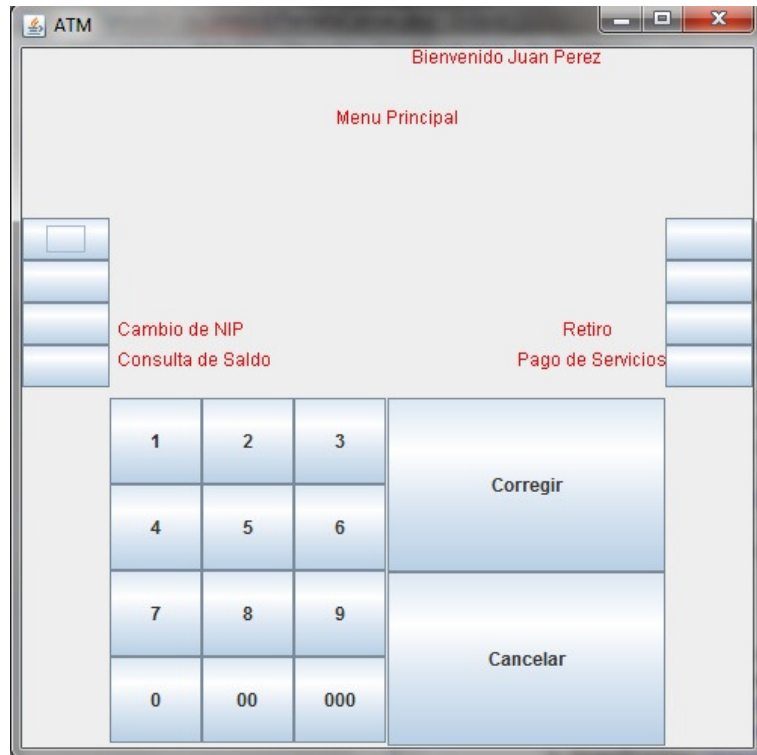
1. Desarrollar la interfaz gráfica que presente como menú principal:

- 1 Sistema Banco: Administración Clientes
- 2 Cajero Automatico

- Para entrar al Sistema de Banco se debe verificar el acceso a través de algún password para evitar la entrada a cualquier persona.
- El Sistema de Banco debe permitir Alta, Bajas y Cambios a Clientes y Cuentas de Clientes
- La interfaz para el Sistema de Banco es libre, debe estar hecha en Swing y debe estar diseñada para aceptar cambios de practicas posteriores
- El Cajero Automático debe tener la siguiente interfaz:



Ejemplos de pantallas en el Sistema del Cajero Automático:



ATM

Bienvenido Juan Perez

Menu Principal

Cambio de NIP
Consulta de Saldo

Retiro
Pago de Servicios

1	2	3	Corregir
4	5	6	
7	8	9	Cancelar
0	00	000	



ATM

Bienvenido Juan Perez

Retiro

\$50.00	\$400.00
\$100.00	\$500.00
\$200.00	\$1000.00
\$300.00	Otra Cantidad

1	2	3	Corregir
4	5	6	
7	8	9	Cancelar
0	00	000	



ATM

Bienvenido Juan Perez

Pago Servicios

Luz

Agua

Telefono

Celular

Predial

Tenencia

1	2	3
4	5	6
7	8	9
0	00	000

Corregir

Cancelar

Siguiendo estos ejemplos, diseñar las pantallas faltantes.