

Unidad

2

DIPLOMATURA EN PROGRAMACION JAVA

Ejercicios

Tecnológica Nacional - Derechos Reservados

Capítulo 4

Clases

Capítulo 4

Ejercicio 1

En este ejercicio se crearán dos subclases de Cuenta, CajaDeAhorro (caja de ahorros) y CuentaCorriente (cuenta corriente). Se sobrescribirá el método retira() para las cuentas corrientes y caja de ahorros. Además se utilizará la palabra clave super para invocar el constructor de la superclase. Notar que se han modificado los paquetes y se han movido clases a los nuevos paquetes.

Creación de las subclases de Cuenta

Se agregarán al paquete operacionesBancarias las subclases CajaDeAhorro y CuentaCorriente

Realizar los siguientes pasos:

1. Abrir espacio de trabajo en el directorio Capítulo 4\Ejercicio 1. Los errores que figuran luego de importar los archivos se deben a las modificaciones que deberán realizarse.

Modificar la clase Cuenta

2. La clase Cuenta ha cambiado: el atributo balance tiene ahora visibilidad protegida. Cambiar el modo de acceso del atributo balance a protegido.

La subclase CajaDeAhorros

3. La clase CajaDeAhorro debe heredar de la clase Cuenta
4. Deberá incluir un atributo interes de tipo double
5. Deberá incluir un constructor público que reciba dos parámetros: balance y porcentajeInteres. Este parámetro deberá pasarse a la superclase llamando a super(balance).

La subclase CuentaCorriente

6. La clase CuentaCorriente deberá heredar de la clase Cuenta.
7. Deberá incluir un atributo proteccionSobregiro de tipo double.
8. Deberá tener un constructor público que reciba un parámetro: balance. Este se tendrá que pasar a la superclase llamando a super(balance).
9. Deberá tener un segundo constructor público que reciba dos parámetros: balance y proteccion. Este constructor deberá pasar el parámetro a la superclase llamando a super(balance) e inicializar el atributo proteccionSobregiro.
10. La clase CuentaCorriente deberá sobrescribir el método retira(). Deberá realizar la siguiente verificación: si el balance en curso es adecuado para cubrir la cantidad a retirar, entonces se procede de la forma usual. Sino si existe un descubierto autorizado (el valor que almacena proteccionSobregiro), se intenta cubrir la

diferencia (balance – cantidad) con el valor que almacena proteccionSobregiro. Si la cantidad que se necesita retirar en retira() se cubre con el giro en descubierto (balance – cantidad + proteccionSobregiro es mayor a 0), se efectúa el retiro, se descuenta de proteccionSobregiro el valor utilizado para cubrir el retiro y la operación se realiza normalmente. Caso contrario, toda la transacción deberá fallar y todos los valores almacenados previos al intento de realizarla deberán permanecer inalterados.

Verificación del ejercicio

11. Compilar y ejecutar el programa PruebaOperacionesBancarias.
12. Verificar la utilización del método retira() y que la salida es:

```
Creando al cliente Juan Perez
Creando al cliente Pedro García.
Creando al cliente Oscar Toma.
Creando a la cliente Maria Soley.
```

```
Recuperando al cliente Juan Perez y su caja de ahorro.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: false
```

```
Recuperando al cliente Pedro García y su cuenta corriente.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: false
```

```
Recuperando al cliente Oscar Toma y su cuenta corriente.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: true
```

```
Recuperando a la cliente Maria Soley y su caja de ahorro unida a su
esposo Oscar.
Deposita 150.00: true
Retira 750.00: true
```

Ejercicio 2

En este ejercicio se determinará la instancia de cada tipo de cuenta Para generar un reporte que simplifique la programación. Esto es, dado un determinado cliente (cliente) puede haber distintos tipos de cuentas.

Realizar los siguientes pasos:

1. Abrir el proyecto del directorio Capítulo 4\Ejercicio 2. Los errores que puedan figurar se deben a las modificaciones que deberán realizarse.

Completar la clase ReporteCliente

Este programa crea un conjunto de clientes y cuentas y genera un reporte de clientes y el balance en sus cuentas.

En la clase ReporteCliente se encontrarán bloques de comentarios que comienzan con `/**` y terminan con `*/`. Estos indican el lugar donde deberá colocar el código que modificará el programa.

2. Utilizar el operador `instanceof` para verificar el tipo de cuenta ("Caja de ahorros" o "Cuenta Corriente") y según el que sea inicializar el String `tipoDeCuenta` a "Caja de ahorros" o "Cuenta Corriente".
3. Imprimir en pantalla el tipo de cuenta y su balance.
4. Compilar y ejecutar el programa
5. Verificar que la salida sea similar a la siguiente:

```
Creando al cliente Juan Perez
Creando al cliente Pedro García.
Creando al cliente Oscar Toma.
Creando a la cliente Maria Soley.
```

```
Recuperando al cliente Juan Perez y su caja de ahorro.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: false
```

```
Recuperando al cliente Pedro García y su cuenta corriente.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: false
```

```
Recuperando al cliente Oscar Toma y su cuenta corriente.
Retira 150.00: true
Deposita 22.50: true
Retira 47.62: true
Retira 400.00: true
```

```
Recuperando a la cliente Maria Soley y su caja de ahorro unida a su
esposo Oscar.
Deposita 150.00: true
Retira 750.00: true
```

```
REPORTE DE CLIENTES
=====
```

Diplomatura en Programación Java

Cliente: Pérez, Juan

Caja de Ahorro: el balance actual es \$222,50

Cuenta Corriente: el balance actual es \$200,00

Cliente: García, Pedro

Cuenta Corriente: el balance actual es \$222,50

Caja de Ahorro: el balance actual es \$380,00

Cliente: Toma, Oscar

Cuenta Corriente: el balance actual es \$0,00

Caja de Ahorro: el balance actual es \$100,00

Cliente: Soley, Maria

Caja de Ahorro: el balance actual es \$100,00

Universidad Tecnológica Nacional - Derechos Reservados