

Cuentas bancarias

Contexto

Una cuenta bancaria es un contrato financiero con una entidad bancaria en virtud del cual se registran el balance y los subsiguientes movimientos de dinero del cliente.
[Wikipedia](#)

Nosotros vamos a ser menos estrictos, ya que por el momento no nos interesa computar los movimientos de dinero del cliente sino los totales instantáneos (es decir, cuánto dinero tiene el cliente a cada momento luego de realizar cada transacción).

Tipos de cuentas

Vamos a establecer, para el contexto de este ejercicio, tres tipos de cuentas: la **Cuenta Sueldo**, la **Caja de Ahorros** y la **Cuenta Corriente**. Cada una cumple con ciertas reglas de negocio, las cuales se detallan a continuación.

Cuenta Sueldo

Es el tipo de cuenta más simple, ya que se rige por la premisa de que en tanto y en cuanto se tenga tanto o más dinero en cuenta del que se quiere extraer, la operación se debe efectuar correctamente.

Caja de Ahorros

Similar a la anterior, pero se pide que luego de la quinta extracción de dinero se cobre un costo adicional por extracción de \$ 6

Cuenta Corriente

La más compleja de las cuentas, ésta permite establecer una cantidad de dinero a girar en descubierto. Es por ello que cada vez que se desee extraer dinero, no sólo se considera el que se posee, sino el límite adicional que el banco estará brindando. Por supuesto esto no es gratis, ya que el banco nos cobrará un 5% como comisión sobre todo el monto en descubierto consumido en la operación. Por ejemplo, si tuviéramos \$ 100 en la cuenta, y quisiéramos retirar \$ 200 (con un descubierto de \$ 150), podremos hacerlo. Pasaremos a deberle al banco \$ 105 en total: los \$ 100 que nos cubrió, más el 5% adicional sobre el descubierto otorgado.

Consigna

Se pide implementar una solución que permita utilizar los tres tipos de cuentas. Para ello deberás emplear los conceptos de *herencia* y *encapsulamiento*. Aquí un ejemplo de cómo debería funcionar:

```
CuentaSueldo cuenta = new CuentaSueldo();
cuenta.depositar(4000.0);

Double saldo = cuenta.getSaldo();
// saldo debe valer 4000.0

cuenta.extraer(500.0);
saldo = cuenta.getSaldo();
// saldo debe valer 3500.0

cuenta.extraer(4000.0);
// la operación no debe poder realizarse
// y debe lanzar una excepción (RuntimeException) – tema a desarrollar 12Jun17
```

Y así con cada tipo de cuenta. Considerar los recargos establecidos por la consigna.

Sugerencias

Una implementación directa puede llevar a la repetición indiscriminada de código. Esto, por supuesto, reduce la mantenibilidad y aumenta la probabilidad de deslizar errores en tu código. Es por ello que deberás buscar una estrategia para no repetir código (la herencia suele ayudar en ese sentido). De hecho, si encuentras que repetiste líneas de código, incluso una línea solamente, es probable que estés plantando una bomba de “inmantenibilidad” a futuro. Intenta revisar y reducir la copia de código al máximo.

A tener en cuenta:

La entrega será grupal (dos integrantes). Se pide que el proyecto del grupo esté compartido en GitHub (<https://github.com/>).

El integrante N° 1 deberá crear el repositorio y el integrante N° 2 deberá trabajar sobre una copia del mismo (fork).

Deberá hacer los test's aplicando lo visto en clase y comprobando la totalidad de la funcionalidad requerida.

A la hora de dividir el trabajo, cada integrante debería realizar aproximadamente el 50% de las funcionalidades.

Una vez completado el se deberá enviar un mail al grupo con el tema [ENTREG A-TP-n] y en el cuerpo del mensaje aclarar:

- nombre y apellido.
- Dni.
- url del repositorio.
- Usar eclipse y java 8.

Fecha límite de entrega: 19 de mayo 2019-