



RETO 2 - PROGRAMACIÓN BÁSICA

VARIANTE 2

El banco Q a pesar de su resistencia al cambio digital se ha visto obligado a implementar un sistema de software para sus cuentas bancarias y así no quedarse atrás respecto a sus competidores.

El banco dispuso como prioridad implementar el módulo correspondiente a las **cuentas de ahorro y corriente**.

Usted ha sido contratado como Java Expert Developer, porque ha logrado demostrar habilidades de desarrollo en este lenguaje de programación y se le ha concedido implementar las clases correspondientes a cuenta bancaria, y cuenta de ahorro y corriente que heredan de la anterior.

Note que **las cuentas bancarias tienen un número de cuenta, propietario, saldo y se les puede depositar o retirar dinero.**

Por cada retiro el banco cobra el 5% del valor a retirar (comisión).

Las cuentas corrientes cobran cuota de manejo con un valor convenido al abrir la cuenta.

La cuenta de ahorro paga al usuario una tasa de interés efectivo anual (IEA) del 1% (0.01) para para un saldo menor a \$1'000.000, 1.75% (0.0175) para un saldo mayor o igual a \$1'000.000 y menor a \$2'000.000, y 2.3% (0.023) para un saldo mayor o igual a \$2'000.000 pesos, el banco reconoce intereses diarios sobre saldos diarios.

Teniendo claras las condiciones para conocer IEA podemos calcular el saldo más el interés diario así:

$$\text{Saldo} = \text{Saldo} \cdot (1 + \text{IEA} / 365)$$

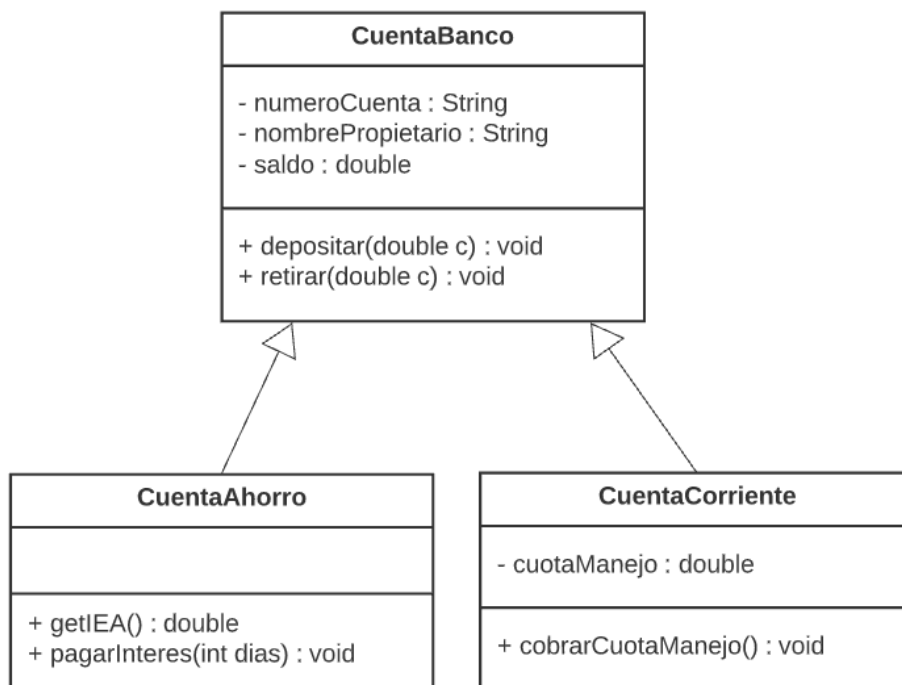
Nota: La fórmula viene del concepto de [interés efectivo o compuesto](#), no es necesario que la entienda a cabalidad.

Para facilitar la implementación de la clases, el equipo de Ingeniería de software le hace entrega del diagrama de clases, recuerde que los métodos relacionados al constructor, getters y setters son obviados en el diagrama de clases, pero deberán ser incluidos en el código (Estos métodos deberán ser





creados con el estándar camel case: Por ejemplo, si el atributo se llama saldo, sus métodos correspondientes a get y set serían getSaldo y setSaldo).



Además del diagrama, el equipo de Ingeniería entrega esta documentación para comprender mejor los elementos del diagrama:

Clase CuentaBanco

Atributos

NOMBRE	TIPO DE DATO	CONCEPTO	INICIALIZACIÓN
numeroCuenta	String	Es el número que identifica la cuenta bancaria.	En el método constructor.
nombrePropietario	String	Es el nombre del propietario.	En el método constructor.
saldo	double	Es el valor de dinero que tiene el usuario en la cuenta.	En el método constructor.





Métodos

NOMBRE	TIPO DE RETORNO	PARÁMETROS	CONCEPTO
depositar	void	double c: cantidad de dinero a depositar.	Suma c al saldo.
retirar	void	double c: cantidad de dinero a retirar.	Resta c al saldo si $c + 0.05 * c$ es menor o igual al saldo.

Clase CuentaAhorro

Métodos

NOMBRE	TIPO DE RETORNO	PARÁMETROS	CONCEPTO
getIEA	double		Retorna el valor de la tasa de IEA correspondiente según las condiciones expuestas anteriormente.
pagarInteres	void	int días: la cantidad de días que han pasado con el saldo constante.	Suma al saldo el interés diario por cada día transcurrido.





Clase CuentaCorriente

Atributos

NOMBRE	TIPO DE DATO	CONCEPTO	INICIALIZACIÓN
cuotaManejo	double	Es la cantidad acordada que debe de pagar el usuario cada mes por el uso de la cuenta.	En el método constructor.

Método

NOMBRE	TIPO DE RETORNO	PARÁMETROS	CONCEPTO
cobrarCuotaManejo	void		Resta al saldo el valor de la cuota de manejo.

PRECISIONES

1. No hay métodos estáticos.
2. El método constructor debe inicializar todos los atributos de dicha clase y de la clase padre si la tiene.
3. Deben existir getters y setters de todos los atributos de cada clase, estos deben ser escritos en la forma estándar, por ejemplo, los métodos getter y setter para la variable saldo serían getSaldo y setSaldo.
4. El retiro no se realiza si el valor a retirar más la comisión es mayor al saldo.
5. El número de días sobre los que se va a calcular el interés debe de ser mayor o igual a 1.
6. El cobro de la cuota de manejo puede resultar en saldo negativo.
7. No haga aproximaciones en ninguna de las operaciones.





TAREAS

- En los archivos preconstruidos en la plataforma Moodle, implementar las clases especificadas en el diagrama de clases, teniendo en cuenta las precisiones dadas por el equipo de Ingeniería de software.
- Los nombres de los métodos y atributos **DEBEN** ser nombrados tal y como aparecen en el diagrama de clases.
- Usted **NO** debe solicitar datos por teclado, ni programar un método main, tampoco use `Java Source Package`, usted está solamente encargado de la construcción de las clases.

NOTA ACLARATORIA

Usted podrá desarrollar la clase requerida en un IDE como NetBeans, y al final copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:

Modo incorrecto:





Modo correcto:



EJEMPLO

El calificador automático hará las veces de usuario, y será quien evalúe la experiencia de usuario que tuvo al usar la aplicación del Banco Q que usted programó:

1. El calificador crea una cuenta corriente con \$50.000 pesos como saldo inicial con una cuota de manejo de \$7.000 pesos y deposita \$600.000 pesos.

Note el orden de los parámetros del método constructor: Primero se ingresa el número de cuenta, luego el nombre del propietario, saldo y cuota de manejo (Si su constructor NO cumple con esta especificación, su calificación se verá afectada).

```
CuentaCorriente c = new CuentaCorriente("1234", "John",  
50000, 7000);  
c.depositar(600000);
```

NOMBRE	CONTENIDO
numeroCuenta	1234
nombrePropietario	John
saldo	650000
cuotaManejo	7000





2. El calificador paga la cuota de manejo de la cuenta corriente.

```
c.cobrarCuotaManejo();
```

NOMBRE	CONTENIDO
numeroCuenta	1234
nombrePropietario	John
saldo	643000
cuotaManejo	7000

Note que el saldo fue el resultado de restar la cuota de manejo (7 mil) al saldo anterior (650 mil)

3. El calificador retira \$150000 pesos.

```
c.retirar(150000);
```

NOMBRE	CONTENIDO
numeroCuenta	1234
nombrePropietario	John
saldo	485500
cuotaManejo	7000

Note que el saldo es el resultado de restar al saldo el valor a retirar más el 5% de dicho valor cómo se muestra a continuación:

$$\text{saldo} - (\text{valorRetiro} + \text{valorRetiro} * 0.05)$$

aplicado al ejemplo sería:

$$643000 - (150000 + 150000 * 0.05) = 485500$$

