

Enunciado nuevos requerimientos

El banco dueño de la aplicación del simulador bancario inicio recientemente una promoción de puntos y regalos con sus clientes. La idea de la promoción es que los clientes ganen puntos con ciertas operaciones dentro del banco. Estos puntos pueden ser canjeados posteriormente por premios o millas de viajero frecuente en las principales aerolíneas del país.

Se requiere que el simulador bancario permita llevar registro de estos puntos para que los clientes puedan proyectar qué tipo de operaciones necesitan para obtener cierto número de puntos.

Las operaciones que entregan puntos son:

- Consignación en cuenta corriente: 5 puntos
- Consignación en cuenta de ahorros: 2 puntos
- Apertura de CDT: 10 puntos

La modificación dentro de nuestro sistema sería agregar la característica de puntos al simulador y modificar los métodos necesarios para llevar registro de esos puntos.

Para solucionar esto usted debe hacer las siguientes modificaciones:

1. Declaración de atributos: Declare en la clase SimuladorBancario un atributo llamado puntos:
2. Inicialización de atributos: Modifique el método constructor de la clase SimuladorBancario para que el atributo puntos sea inicializado en 0.
3. Declaración e implementación de métodos: Agregue a la clase SimuladorBancario un método llamado darPuntos(). La responsabilidad de este método es retornar el total de puntos obtenido por cliente según lo explicado inicialmente.
4. Implementación de métodos: Modifique los métodos consignarCuentaCorriente(..), consignarCuentaAhorros(..) e invertirCDT(..) de la clase SimuladorBancario para que, además de la funcionalidad que ya tienen, incrementen los puntos del cliente según el enunciado.

Parte 2

1. Agregue a la clase SimuladorBancario un atributo para modelar el tipo de cliente (Ejemplos de tipos de clientes son: "VIP", "Platino" o "Normal").
2. Complete el método constructor de la clase SimuladorBancario para que inicialice el tipo de cliente según un valor recibido por parámetro.
3. Implemente el método llamado cambiarTipoCliente() de la clase SimuladorBancario para que actualice el valor del atributo creado en el punto 1, con un valor recibido por parámetro.
4. Modifique el método retirarValor() de la clase CuentaCorriente para que se le descuente del saldo al cliente, el 1% del valor retirado

5. En la clase CDT, modifique el método `calcularValorPresente()` para que ahora retorne el valor presente utilizando un interés compuesto. En la siguiente formula, la variable n corresponde al número de meses transcurridos, la variable i corresponde al valor del interés y, por último, la variable p corresponde al valor inicial del CDT. $p * (1 + i)^n$