**EVALUACIÓN 1**

**ASIGNATURA: DSY1102 – PROGRAMACIÓN ORIENTADA A OBJETOS**

**SECCIÓN:**

|  |  |  |
| --- | --- | --- |
| **PUNTAJE MÁXIMO** | **100 puntos** | **NOTA** |
| **PUNTAJE OBTENIDO** |  |
| **FECHA** |  |
| **DURACIÓN** | **120 minutos** |

**INDICACIONES GENERALES**

La nota 4,0 se obtiene logrando un 60% del puntaje total.

Cuando termine la evaluación, comprima la carpeta donde ha dejado los trabajos de la prueba y siga las instrucciones del docente.

**LEER ATENTAMENTE LA PROBLEMÁTICA PLANTEADA ANTES DE LA CREACIÓN DEL PROYECTO.**

**ENUNCIADO**

Una empresa que billeteras virtuales llamada **DUOMATICA** se encuentra en una etapa de modernización de sus procesos, por lo cual, han decidido contratar sus servicios para que desarrolle un sistema que permita almacenar la información de los clientes y sus tarjetas.

Para asignar una tarjeta se requiere información básica del cliente quien será asociado a la misma. Considere que una tarjetatiene **un solo dueño**.

Determine 5 atributos para el **cliente** y 5 atributos para la **tarjeta** (como mínimo), que permitan guardar la información de la tarjeta que utilizará el cliente. Debe aplicar **COLABORACIÓN** y además debe **cumplir con los siguientes requerimientos** del cliente (dueño de la empresa):

* Debe **existir un método** que permita mostrar la información de ambas clases.
* Debe **existir un método** que permita realizar una compra y aplicar “*CashBack”,* siempre cuando sea su primera compra.
* Debe **existir un método** que permita “enviar un correo” (imprimir un mensaje a la consola), el cual, se deberá ejecutar cuando se realice un cargo o compra en la tarjeta. El mensaje debe contener, todos los datos del dueño y de la tarjeta.

Implemente las siguientes reglas de negocio:

* El valor del abono a la tarjeta debe ser superior a los $0.
* El rut, nombre del cliente y el numero de la tarjeta no deben ser vacíos.
* El límite de la devolución o “*CashBack*” es de $10.000.-

**Si no se cumple con estas reglas, debe enviar un mensaje a la consola.**

Además, ambas clases deben tener los siguientes métodos implementados: constructores con y sin parámetros, accesadores y mutadores.

Desde una clase **Principal**, que contiene el **main,** crear un menú **con las siguientes opciones**:

1. Crear 2 **Clientes**.
2. Crear 2 **Tarjetas** y asociarle un cliente a cada una de ellas.
3. Mostrar la información de todas las **Tarjetas**.
4. Realizar una compra a un **cliente** nuevo.
5. Realizar una compra a un **cliente** antiguo.
6. Salir.

**Otras consideraciones en la creación de la solución:**

* Crear las variables necesarias para cumplir con los requisitos.
* Los datos los puede ingresar directo en el código o por el teclado.
* La compra debe ser ingresada por teclado.
* Recordar que se debe programar en **Java** y en Apache NetBeans, aplicando todas las convenciones aprendidas en clases y **encapsulando** los atributos de la clase.

**UTILIZAR SOLO INSTRUCCIONES VISTAS EN CLASES O PPTS DISPONIBLES EN AVA O GITHUB.**

**CREAR DEL PROYECTO CON SU NOMBRE COMPLETO, EN EL DISCO DE RESPALDO. NO UTILIZAR IA U OTRA HERRAMIENTA DE GENERACIÓN AUTOMATICA DE CODIGOS.**