TEST DEL MODULO JAVA

Sistema de Gestión Financiera en Banco Unión

Introducción

Se te asigna la tarea de desarrollar un sistema en Java para una institución financiera que permita gestionar cuentas corrientes y emitir cheques de manera eficiente. Este sistema debe incluir operaciones administrativas y financieras clave, aplicando los principios más avanzados de programación orientada a objetos, patrones de diseño, y acceso a bases de datos. Además, el examen evaluará tus habilidades para resolver problemas algorítmicos, trabajar con estructuras de datos modernas y optimizar procesos concurrentes mediante herramientas y técnicas actuales del lenguaje Java.

El objetivo principal es crear un sistema funcional que cumpla con los requisitos especificados, asegurando que sea capaz de gestionar datos financieros de manera segura, clara y organizada, cumpliendo con los estándares necesarios en el mundo real.

Contexto del Sistema

En este banco ficticio, se necesita un programa capaz de realizar las siguientes tareas:

1. **Gestor de Cuentas Corrientes:**
    - Este módulo debe permitir que los clientes puedan realizar transacciones esenciales, como depósitos y retiros, y también habilitar la emisión de cheques.
    - Existen dos tipos de cuentas:
        - **Cuentas Personales:** Diseñadas para clientes individuales, con un saldo máximo permitido de $10,000,000.
        - **Cuentas Empresariales:** Orientadas a clientes corporativos, sin límite de saldo.
2. **Gestor de Cheques:**
    - Este módulo se encarga de gestionar la información y procesamiento de cheques. Los cheques deben incluir datos esenciales como el beneficiario, monto, firma digital y un nivel de prioridad que determine el orden de procesamiento.
    - Debe permitir priorizar cheques urgentes para una mejor gestión operativa.

Datos Necesarios para los Clientes

Cada cliente debe estar registrado con los siguientes detalles fundamentales:

- **Identificación:** Un número único que permita identificar al cliente, como su cédula o NIT.
- **Nombre completo:** Nombres y apellidos.
- **Dirección:** Ubicación completa para notificaciones o envíos postales.
- **Teléfono y correo:** Datos de contacto actualizados para comunicación directa.
- **Estado del cliente:** Puede ser Activo (puede realizar transacciones) o Inactivo (todas sus operaciones están restringidas).

Reglas del Sistema

1. **Cuentas Corrientes:**
    - Cada cliente activo puede tener una o más cuentas asociadas.
    - **Límites para cuentas personales:** Estas cuentas no deben exceder un saldo máximo de $10,000,000. Si se intenta superar este monto, el sistema debe rechazar la transacción e informar al usuario.
    - **Cuentas empresariales:** Estas cuentas no tienen restricciones en el saldo.
    - Cuando un cliente cambia su estado a inactivo, todas sus cuentas deben ser bloqueadas automáticamente para impedir nuevas transacciones.
2. **Cheques:**
    - Un cheque solo puede emitirse si el saldo de la cuenta es suficiente para cubrir el monto.
    - La información necesaria para un cheque incluye:
        - Identificación del cliente emisor.
        - Nombre completo del beneficiario.
        - Monto del cheque expresado tanto en número como en letras.
        - Una firma digital generada por el sistema para garantizar su autenticidad.
        - Nivel de prioridad: **Alta**, **Media** o **Baja**, que determina el orden de procesamiento.
    - La emisión de cheques debe incluir una simulación de impresión por pantalla con un formato profesional. A continuación, se presenta un ejemplo:

```text
---------------------------------
BANCO UNIÓN S.A.
Cheque No: 123456
Fecha: 2023-06-20

PAGUESE A: Ana López
LA SUMA DE: Ochocientos mil pesos M/Cte

Valor: $800,000

FIRMA DIGITAL: ABCD1234
---------------------------------
```

Nota: Este es solo un ejemplo. El formato final puede variar según la implementación del estudiante, pero debe incluir todos los datos esenciales.

Tareas que Debe Realizar el Sistema

1. **Operaciones CRUD**

- **Cheques:**
    - Emitir nuevos cheques para un cliente verificando los límites y el saldo disponible.
    - Listar los cheques emitidos previamente de un cliente

2. **Validaciones:**
    - Antes de cualquier operación, verificar que el cliente esté activo y tenga cuentas válidas.
    - Comprobar que el saldo de la cuenta cubra el monto del cheque antes de emitirlo.
3. **Reportes:**
    - Generar un listado de cheques emitidos, mostrando:
        - Cliente emisor.
        - Beneficiario.
        - Monto.
        - Prioridad.
    - Guardar los reportes en archivos de texto con un formato claro y ordenado. El formato es yyyymmdd-hhmmss-procesamiento-cheques.txt
4. **Procesamiento de Cheques Pendientes:**

    El sistema debe procesar cheques pendientes de manera eficiente, priorizando aquellos marcados como de alta urgencia. Durante este proceso, el usuario podrá continuar realizando otras operaciones en el sistema sin interrupciones. Los cheques serán ordenados según su prioridad y luego procesados uno por uno, validando información clave como el saldo disponible en la cuenta asociada y la autenticidad de los datos del cheque. Una vez completado el procesamiento, los resultados deben guardarse en un archivo de texto, detallando información como el ID del cheque, el estado final (procesado o rechazado), y las razones en caso de rechazo.

    Los archivos de procesamiento deben llevar el formato yyyymmdd-hhmmss-procesamiento-cheques.txt. La hora está en formato militar.

    Con el fin de colapsar el sistema, este listsado se empezará a generar 10 segundos después de lanzarlo y el usuario podrá seguir con su trabajo en el sistema

    Un cheque se considera *pendiente* si cumple con las siguientes condiciones:

    - ***Estado no finalizado:*** El cheque no ha sido marcado como "procesado" o "rechazado".
    - ***Sin cobro asociado:*** No hay evidencia de que el monto haya sido debitado de la cuenta emisora.

Ejemplo de Formato del Archivo de Procesamiento:

```text
---------------------------------
Resultados del Procesamiento de Cheques Pendientes
Fecha de procesamiento: 2023-12-12
---------------------------------

Cheque ID: 123456
Cliente: Juan Pérez
Beneficiario: María Gómez
Monto: $500,000
Estado: Procesado
Razón: -

Cheque ID: 123457
Cliente: Ana López
Beneficiario: Pedro Ruiz
Monto: $1,200,000
Estado: Rechazado
Razón: Saldo insuficiente

Cheque ID: 123458
Cliente: José Martínez
Beneficiario: Carla Torres
Monto: $350,000
Estado: Procesado
Razón: -

Cheque ID: 123459
Cliente: Laura Gómez
Beneficiario: Javier López
Monto: $2,000,000
Estado: Rechazado
Razón: Cuenta bloqueada
```

---

6. **Identificar clientes sin actividad transaccional:**

El sistema debe generar un listado de clientes que, teniendo cuentas activas, no han realizado ningún tipo de transacción ni emitido cheques en el ultimpo mes. Este análisis debe considerar únicamente aquellos clientes con cuentas activas y generar un reporte que incluya su identificación y su estado actual en el sistema. Este listado permite entender patrones de inactividad dentro de la base de clientes, ayudando a focalizar estrategias futuras. Por ejemplo, identificar clientes con cuentas activas que no hayan realizado transacciones recientes o listar aquellos que, a pesar de tener cuentas activas, no han emitido ningún cheque. Este tipo de análisis permite generar reportes más completos y útiles para la institución financiera.

7. **Reporte combinado de clientes con transacciones recientes y cheques pendientes:**

El sistema debe diseñar un proceso para generar un reporte que combine dos listados importantes: clientes que realizaron transacciones recientes y clientes con cheques pendientes. Este reporte debe identificar a los clientes que aparecen en ambas categorías y detallar información como su nombre, saldo actual, número de cheques pendientes, y el estado de sus cuentas. La finalidad es priorizar la gestión de estos clientes, asegurando que el reporte sea claro, ordenado y eficiente para el análisis y la toma de decisiones. Además, este proceso puede ser reutilizado para futuras combinaciones de datos que sean requeridas por la institución.

Requerimientos

- La interfaz será por consola a través de menús que permitan acceder a las operaciones requeridas
- En el desarrollo se deben tener las buenas prácticas de software, principios SOLID, patrones de diseño. Para identificar el uso de los patrones de diseños, agregue en el código el comentario: `// REQ: PATRONES DE DISEÑO` de esta manera será fácil encontrarlos para su calificación
- Donde considere aplique funciones y expresiones Lambda y Stream API. Escriba en el código el comentario `// REQ: FUNCION LAMBDA` o `// REQ: EXPRESION LAMBDA` o `// REQ: STREAM API`
- Donde considere necesario y si alguna tarea lo requiere use hilos.
- Si considera hacer inserciones manuales de datos en la base de datos agregue en el proyecto un archivo de script sql con las operaciones que hizo.

Insumos Proporcionados

1. **Diagrama Relacional:** Un esquema visual que muestra las relaciones entre clientes, cuentas y cheques.

2. **Script SQL:** Instrucciones detalladas para crear las tablas necesarias en la base de datos y cargar datos iniciales.

Nota: Los insumos lo puede conseguier en la carpeta compartida del siguiente enlace:

https://drive.google.com/drive/folders/1FB8bwhV8ZLPOUnqor3D7fcCAK807t2bL?usp=sharing