Concepto de evaluación:

*Leé completo y con cuidado el enunciado. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es evaluar la correcta aplicación de los conceptos y técnicas vistos hasta el momento:*

* *Correcta implementación de constructores.*
* *Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, publicando setters y getters sólo cuando corresponda.*
* *Manejo de clases, colecciones, y arreglos uni y bidimensionales.*
* *Posicionamiento directo e indirecto.*
* *Excepciones*
* *Herencia, Polimorfismo, Interfaces*
* *Pilas, Colas y Listas Ordenadas*
* *Importación y Exportación de proyectos Java desde Eclipse.*

Enunciado

Una entidad financiera nos contrata para desarrollar un programa que les permita a sus clientes registrar sus operaciones correspondientes a la compra de criptomonedas.

Del cliente se sabe su edad, nombre y las transacciones que realiza, que se guardan en orden inverso al que fueron generadas.

Esta entidad financiera trabaja con diferentes tipos de criptomonedas, Bitcoin y Ethereum.

Cada moneda cuenta con un coeficiente (de tipo double) que no puede ser menor a 1

Utilizan las Transacciones para poder comprar y guardar el tipo de criptomoneda que se compró y cantidad (en su correspondiente moneda), de tipo *double*.

Para que un cliente pueda realizar una transacción primero tiene que registrarse en esta entidad financiera, que tendrá un número límite de clientes posibles a registrar. Es decir que puede suceder que al intentar registrar un cliente, esto no se pueda realizar por falta de disponibilidad.

Para poder hacer una compra, el cliente deberá indicar el tipo de criptomoneda a comprar y la cantidad de dólares con la que dispone. Por lo tanto, al momento de efectuar la compra, automáticamente se generará una transacción en el tipo de moneda indicado, y la cantidad corresponderá al valor en dólares convertido a la moneda correspondiente.

Tanto la entidad financiera como el cliente deben ser “detallables”.

La entidad también nos solicita que se generen distintos tipos en relación a las transacciones existentes.

**Para convertir de dólar a Bitcoin, la fórmula es:**

valor\_en\_dolares / coeficiente - (valor\_en\_dolares / coeficiente \* 0.1)

**Para convertir de dólar a Ethereum, la fórmula es:**

valor\_en\_dolares / coeficiente \* 0.1234

**Para aprobar el examen se deberá realizar lo siguiente:**

1. Monedas: Realiza el código necesario para que sea escalable
2. Transacción: agregar los atributos correspondientes, cuyos valores seran recibidos por parámetro en el constructor.
   1. Completar *public String getNombreMoneda()*, tal que devuelva el nombre de la moneda correspondiente a la transacción
   2. Completar *public void convertir(double val)* tal que reciba el valor de dólares y convierta al valor de la criptomoneda de dicha transacción.
3. Cliente: agregar los atributos correspondientes
   1. Completa el constructor de la clase Cliente, asignando los parámetros recibidos a los respectivos atributos.
   2. Completar *public double cantMonedas(String moneda)* tal que muestre la cantidad que tenga del tipo de moneda recibido por parámetro. Las estructuras utilizadas en este método deben quedar como estaban al principio una vez que se finaliza este proceso.
   3. Completar *public void agregarTransaccion(Moneda moneda, double valor)* tal que agregue una Transaccion nueva
4. Entidad Financiera: agregar los atributos correspondientes
   1. Constructor: completar el método, teniendo en cuenta que no se debe permitir que el nombre de la entidad sea vacío o nulo.
   2. Completar *private int verificarDisponibilidad()* tal que devuelva el valor de la primera posición disponible para registrar un nuevo cliente. En caso negativo, devolverá -1.
   3. Completar *public void agregarCliente(String nombre, int edad)*, tal que agregue el cliente en la entidad en la primera posición disponible (solamente si hay disponibilidad).
   4. Completar *public void agregarTransaccion(int nroCliente*, *Moneda moneda, double valor)* tal que le asigne una nueva transacción al cliente correspondiente a ese número.
   5. Completar *public void cantMonedasPorCliente(String moneda)* tal que imprima por pantalla la cantidad de monedas para cada cliente del tipo de moneda recibido por parámetro.
   6. Completar *public void mostrarDetalleCliente(int nroCliente)* tal que muestre el detalle del cliente correspondiente al id recibido por parámetro.
5. mostrarDetalle: Implementar el método *mostrarDetalle()* definido en la interfaz Detallable.
   1. Cliente: debe mostrar el detalle de todas las transacciones, incluyendo moneda y cantidad.
   2. EntidadFinanciera debe mostrar el detalle de cada cliente, pero los mismos deben estar ordenados de menor a mayor.
6. Test
   1. Completa el método *crearEntidad*

Donde se le pasa por parámetro el nombre de la entidad y el límite del número de clientes y se genera una nueva EntidadFinanciera

**Debe poder manejar errores**

* 1. Completa la funcion *mostrarInfoCliente*

Que le pida al usuario que ingrese por teclado el nro de cliente del cual desee mostrar el detalle. Debe pedirlo hasta que ingrese un número válido.

**Debe poder manejar errores**

**Un Ejemplo de la salida del programa es el siguiente:**

|  |
| --- |
| **La cantidad de clientes no puede ser menor o igual a cero Se ha agregado el cliente Jose Perez en la posicion: 1 Se ha agregado el cliente Alejandra Gonzalez en la posicion: 2 Se ha agregado el cliente Carlos Garcia en la posicion: 3 ---------------------- Mostrando el detalle de los clientes de: EntidadVirtual Alejandra Gonzalez - Edad:30 Transaccion: 1- Moneda: Bitcoin , Cantidad: 0.1377 Transaccion: 2- Moneda: Bitcoin , Cantidad: 1.27125 Transaccion: 3- Moneda: Bitcoin , Cantidad: 1.395 Transaccion: 4- Moneda: Ethereum , Cantidad: 0.29616 Carlos Garcia - Edad:27 Transaccion: 1- Moneda: Ethereum , Cantidad: 0.003796923076923077 Transaccion: 2- Moneda: Ethereum , Cantidad: 0.030375384615384617 Transaccion: 3- Moneda: Bitcoin , Cantidad: 4.005 Jose Perez - Edad:55 Transaccion: 1- Moneda: Bitcoin , Cantidad: 0.1674675 Transaccion: 2- Moneda: Ethereum , Cantidad: 0.17465846153846154 Transaccion: 3- Moneda: Ethereum , Cantidad: 0.04556307692307692 Transaccion: 4- Moneda: Bitcoin , Cantidad: 0.315 ---------------------- Mostrando la cantidad de monedas Bitcoin por cliente Cliente: 1 - Jose Perez. Promedio del tipo Bitcoin: 0.24123375 Cliente: 2 - Alejandra Gonzalez. Promedio del tipo Bitcoin: 0.93465 Cliente: 3 - Carlos Garcia. Promedio del tipo Bitcoin: 4.005 ---------------------- Ingrese la edad de un cliente: 30 Alejandra Gonzalez - 30 Transaccion: 1- Moneda: Bitcoin , Cantidad: 0.1377 Transaccion: 2- Moneda: Bitcoin , Cantidad: 1.27125 Transaccion: 3- Moneda: Bitcoin , Cantidad: 1.395 Transaccion: 4- Moneda: Ethereum , Cantidad: 0.29616 ----------------------** |