Guía N° 4: Listas y Arreglos

Contents

Pasos iniciales: Crear un proyecto en eclipse

Deberá crear un proyecto Java, en eclipse, denominado "died-guia04".

Crear en la carpeta "src" el paquete "died.guia04"

Ejercicio 01

Realizar un sistema de generación recepción de apuestas.

El sistema permite registrar **apuestas** realizadas por un usuario. De cada apuesta se registra el nombre del usuario y un arreglo de 6 valores de tipo byte que son los que el que el usuario decide apostar.

La clase apuestas posee un método: calcularAciertos, que recibe como argumento un arreglo con los números que salieron como resultado del sorteo y retorna la cantidad de aciertos que tuvo el jugador.

El sistema posee una clase **sorteo,** que guarda la semana del año en que se realiza el sorteo, y el año (por ejemplo la apuesta de la primera semana del año se registra como {**numero: 1 – año 2023}**). También registra todas las apuestas en un ArrayList de apuestas.

A su vez la clase sorteo tiene 2 métodos:

Generar sorteo: genera 6 números aleatorios entre 0 y 42 y los almacena en un arreglo de bytes.

Calcular ganadores: este método invoca de cada apuesta el método calcularAciertos y guarda en una lista temporal de ganadores, las apuestas con mas aciertos. Así si hubo 3 apostadores que con 5 aciertos y el resto tuvo 4 o menos aciertos este método retorna una lista con las 3 apuestas que tuvieron 5 aciertos.

Implementar en java las clases:

- Apuesta: puede crear los constructores / getters / setters que considere necesarios
- Sorteo: puede crear los constructores / getters / setters que considere necesarios

- Y crear una clase App con un método main que:
    - cree al menos 4 apuestas
    - realice un sorteo
    - imprima los ganadores y la cantidad de aciertos.

Ejercicio 02

Un sistema de gestión de ventas de un minimercado necesita registrar los pedidos que realizan los clientes y obtener estadísticas a través de ellos.

De los productos se conoce su descripción, código, precio unitario y la medida de las unidades (un enum que permite registrar valores como KILO – UNIDAD – LITRO – METRO – etc).

Cada Pedido esta asociado a un cliente y posee una lista de DetallePedido, donde se consigna el producto y la cantidad solicitada. El detalle pedido tiene un método que retorna el precio de esa línea del pedido. A su vez el pedido tiene un método que retorna el subtotal (el precio de cada línea de pedido) y el total, que es el subtotal mas el 21% de IVA en caso de que el cliente sea consumidor final.

De cada cliente se conoce el nombre, el correo electrónico y el CUIT (long) y si es consumidor final o no (boolean). Además cada cliente registra una lista de los pedidos que realiza.

El cliente tendrá los siguientes métodos:

- iniciar pedido: abre un pedido y permite agregar productos al pedido y retorna true si esto es posible o false si ya existe un pedido abierto
- agregar producto: si un pedido esta abierto, agrega un producto y la cantidad indicada recibida como parámetro al pedido actual y retorna true. Si no hay pedido abierto retorna false
- confirmar pedido: si hay un pedido abierto y tiene al menos un producto agrega el pedido a la lista de pedidos, pone en null el pedido actual y retorna true. Caso contrario retorna false.
- costoPromedioPedido: calcula el costo promedio de todas las compras que realizó el cliente
- productoMasComprado: busca el producto cuya cantidad en unidades es la mayor que se ha comprado (por ejemplo si compro en total 60 litros de agua y 40 kg de pan, el producto mas comprado son 60 litros de agua).

Crear las clases para representar este problema y una clase App con un método main que permita crear un pedido para un cliente y verificar los métodos solicitados.