Reto 3 Java

Lenguaje Java Intentos

1 de 10

Puntaje requerido 90% Enunciado

# Reto 3: Estadísticas para nuestra tienda

En este reto continuaremos trabajando con la aplicación de inventario para la tienda de la Sergio Arboleda desarrollada en el Reto 2. A continuación, se presenta nuevamente el contexto del problema.

Una de las cafeterías de la Sergio Arboleda vende frutas, dulces y algunas bebidas. Con el propósito de mejorar el control sobre las ventas y el inventario de la tienda. La universidad decide construir una aplicación en Java que le permita almacenar la información de los productos y calcular algunas estadísticas sobre el inventario. La Tabla 1 presenta los productos disponibles a la fecha en el almacén.

# código nombre precio inventario

1. Manzanas 5000.0 25
2. Limones 2300.0 15
3. Peras 2700.0 33
4. Arandanos 9300.0 5
5. Tomates 2100.0 42
6. Fresas 4100.0 3
7. Helado 4500.0 41
8. Galletas 500.0 8
9. Chocolates 3500.0 80
10. Jamon 15000.0 10

# Tabla 1: Productos disponibles en la tienda

Nota: observe que los nombres de los productos no llevan tildes. Esto se hace a propósito y tiene que ver con el proceso de calificación automática. Evite colocar tildes en los nombres de los productos en su código.

Debido a algunos cambios en el modelo de negocio, la gerencia solicita modificar el informe generado con la información de los productos. En este caso, se requiere actualizar la función generarInforme desarrollada en el reto anterior para que muestre:

* El nombre del producto con el precio mayor
* El nombre del producto con el precio menor
* El promedio de los precios de los productos
* Producto precio mayor: Jamon
* Producto precio menor: Galletas  Promedio de precios: 4900.0

Por ejemplo, al generar el informe con los datos disponibles en la Tabla 1 obtendríamos:

Para evitar errores a la hora de agregar, actualizar y borrar productos, se debe implementar un método llamado verificarExistencia que permitirá comprobar si el producto existe o no en la base de datos. Este método debe implementarse en la clase BaseDatosProductos.

Cada uno de los casos de prueba estará compuesto por dos líneas.

La primera línea estará formada por una cadena de texto que identifica la operación a

realizar. En este caso, las operaciones válidas son:

ACTUALIZAR

BORRAR

**Entrada** AGREGAR

La segunda línea estará formada por 4 valores (código, nombre, precio, inventario) que representan el producto sobre el cual se quiere realizar la operación.

En el caso de la operación ACTUALIZAR la segunda línea debe contener el código y los nuevos valores del producto

En el caso de la operación BORRAR se deben especificar todos los atributos del producto a eliminar

La salida estará representada por una única línea formada por tres valores separados por un espacio:

Nombre del producto con el precio mayor

Nombre del producto con el precio menor

**Salida** Promedio de precios

Estos 3 valores deben imprimirse después de realizar las operaciones solicitadas en la primera línea

Los valores numéricos deben imprimirse con un número decimal

En caso de solicitar ACTUALIZAR o BORRAR un producto que no existe (es decir, que el código del producto no se encuentra en la base de datos), se debe imprimir 'ERROR'.En caso de solicitar AGREGAR un producto cuyo código ya existe en la base de datos se debe imprimir 'ERROR'

**Casos de prueba:**

# Entrada Salida Esperada

AGREGAR

Jamon Melon 4460.9

11 Melon 70 13

BORRAR

Arandanos Galletas 3777.8

10 Jamon 15000 10

ACTUALIZAR Helado Galletas 10950.0

7 Helado 65000 11

BORRAR

ERROR

14 Maiz 45000 12

Dificultad

Instrucciones

# Instrucciones para la calificación automática

Antes de enviar la solución del reto, por favor tenga en cuenta los siguientes aspectos:

* No coloque tildes a los nombres de los productos.
* Cada caso de prueba se especifica en una única línea.
* Cada línea debe contener los valores de los parámetros separados por un espacio.
* Debe existir en el código, una clase principal llamada "Reto3".
* En la clase "Reto3" debe existir un método llamado "run".
* El código no debe contener variable Scanner, recuerde trabajarla como variable global en la clase Reto0 (no instanciarla en el método principal "run").
* Es importante no utilizar ningún mensaje a la hora de capturar las entradas.
* Únicamente debe imprimir los resultados finales. No imprima valores con cálculos intermedios.
* Evite declarar atributos de clase.

