

Tarea presencial. Unidad 7.

Estructuras de datos externas. Ficheros

Indicaciones de entrega

Una vez realizada la tarea, el envío se realizará a través de la plataforma Moodle, en la tarea "Tarea unidad 7. Estructuras de datos externas. Ficheros". Deberás entregar únicamente el archivo SuperEcoMercado.java

Evaluación

Cómo se valora y puntúa la tarea

Método actualizar los productos del supermercado	→ 3.5 puntos
Método para procesar la compra	→ 2.5 puntos
Multiplicador de productos en caja	→ 1 punto
Acumulación de productos en la compra	→ 1 punto
Método para mostrar el ticket	→ 1 punto
Control de errores	→ 1 punto

Factores que influyen en la puntuación de cada ejercicio

- Principal: El código debe compilar y ejecutarse correctamente
- Errores de base sobre conceptos de temas anteriores: nombres de identificadores incorrectos, elección de tipo inadecuado, etc.
- Elección de la estructura de control de flujo más apropiada para cada solución: if-else, switch-case, for, while...
- Calidad, legibilidad y limpieza del código.
- Control y manejo de excepciones que proporcionen una buena experiencia de usuario.
- La implementación de métodos que modularicen los programas.

Resultados de aprendizaje que se trabajan en la tarea

RA5: Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

RA6: Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

Enunciado

Un supermercado de productos ecológicos necesita hacer un programa para vender su mercancía. Crea una única clase principal llamada SuperEcoMercado.java donde se incluya todo el código necesario.

Ejemplo de productos que se indican en la carga inicial. Los productos se venden por unidades de bote, brick, etc.

AVENA	GARBANZO	TOMATE	JENGIBRE	QUINOA	GUISANTE
2,21	2,39	1,59	3,13	4,5	1,6

Apartado 1. Carga de productos y precios

La carga y actualización de los productos y los precios se hará mediante una carga masiva por fichero.

- La carga se realiza cuando se introduce en la caja la palabra "UPDATE" seguida del nombre del fichero. Ejemplo:
UPDATE carga_inicial.csv
- En caso de que un producto lleve como precio negativo o cero, significará que ese producto se retira (o no se añade) del supermercado.
- En caso de que sea un valor positivo, indicará el producto y el precio actual.
- Los nombres de los productos siempre se enviarán y guardarán en mayúsculas.
- Los productos que no aparezcan en el fichero de carga masiva quedarán en el supermercado sin ninguna modificación.
- Una carga de fichero NO podrá realizarse si hay una compra en curso.
- Se aportan los ejemplos de fichero carga_inicial.csv y actualizacion_prod.csv
- El fichero tendrá el siguiente formato: Producto;Precio. La primera línea servirá a modo de leyenda y no se debe procesar.

```
TOMATE;1.65
JENGIBRE;0
QUINOA;4
```

Apartado 2. Compra en caja

Cuando se pasan los productos por el escáner de la caja, se leerá el producto y se contabilizará una unidad.

La persona que está en la caja también podrá previamente introducir el número de artículos para solo leer el producto una vez por el escáner, en tal caso sumará el número de unidades indicado por teclado.

Por ejemplo se introduciría un 3 y luego se leería el producto "GUISANTE" por el escáner si se quieren comprar 3 botes de guisantes.

La compra se termina cuando la persona que está en la caja introduce un 0.

Los productos podrán escanearse por separado, luego en el ticket se mostrarán los precios agrupados por producto.

Apartado 3. Cierre de la caja

Al recibir la instrucción EXIT, se cerrará el programa (interrumpiendo la compra actual si la hubiere), dejando el estado actual de los productos en un fichero con nombre "cierre_YYYYMMdd.csv" para poder cargarlo en una futura apertura de la caja.

NOTAS

- Utiliza un Map<String, float> para almacenar los nombres y precios de los productos.
- Se aconseja leer el fichero de carga apoyándose en la clase BufferedReader.
- Se aconseja usar el método nextLine() de la clase Scanner para la introducción de los inputs.
- La caja se queda siempre a la espera de un input. Los inputs permitidos son (No hay que comprobar que los inputs son correctos):
 - Nombre de un producto
 - Número de cantidad de producto
 - "0" (número cero) para terminar una compra
 - Instrucción de actualización de productos. "UPDATE fichero.csv"
 - Instrucción de salida del programa "EXIT"
- El propio software del escáner lee el código de barras del producto y lo transforma al nombre correspondiente que tenemos en la tabla.
- Suponemos que el usuario no va a intentar comprar un producto que no existe.
- Puedes usar los métodos esNumeroEntero y formateaFloat.

Ejemplo ejecución apartado 1

```
INPUT: UPDATE carga_inicial.csv
PRODUCTOS ACTUALIZADOS CON ÉXITO
INPUT:
```

Ejemplo ejecución apartado 2

```
INPUT: TOMATE
INPUT: 2
INPUT: QUINOA
INPUT: AVENA
INPUT: 3
INPUT: TOMATE
INPUT: 0
```

Producto	Precio	Cantidad	Subtotal
TOMATE	1,59	4	6,36
QUINOA	4,50	2	9,00
AVENA	2,21	1	2,21

Total: 17,57 €			
INPUT:			