TIENDA ORIENTADA A OBJETOS 3º PARTE

El objetivo de esta práctica es completar la re-implementación de la tienda con aspectos relacionados con las excepciones y los ficheros. Programando los siguientes requisitos:

- Crear la clase **TiendaException** que hereda de **RuntimeExcepcion** (excepciones no comprobadas) con su método constructor.
- Dentro de vuestra clase **Producto** realizar las siguientes modificaciones:
  - En el método constructor comprobar que los parámetros de entrada cumplen con los requisitos básicos, como strings no vacíos, valores numéricos >=0, char categoría con valor correcto, además de que el precio de venta no puede ser menor al coste unitario. En caso de no cumplir con algún requisito lanzar una **TiendaException** con un mensaje adecuado según el error cometido.
  - En los métodos modificadores sustituir el mensaje con sout por lanzar una excepción **TiendaException** con mismo mensaje que tenía el sout.
  - Eliminar cualquier sout que tuviera el código de clase.
- Dentro de vuestra clase **Tienda** realizar las siguientes modificaciones y ampliaciones:
  - En los métodos de añadir producto, borrar, vender, subir y bajar precio sustituir los sout de error por lanzar una excepción **TiendaExcepcion**.
  - Añadir un método para guardar los datos de productos que estén en ese momento en el **HashMap** en un fichero de texto, donde cada línea guarda la información (solo sus atributos sin meter el orden) de un producto usando separadores de formato que vosotros elijáis. No usar como separador `\ ^ $ . | ? * + ( ) [ {` que son caracteres especiales del método **split**. Capturar **IOException** y lanzar **TiendaException** en su lugar.
  - Añadir un método para cargar datos de un fichero de texto con el formato que hayáis elegido en el apartado anterior, de forma que se añada al **HashMap**. Ambos métodos anteriores tienen como parámetro el nombre del fichero donde se guardan/cargan los datos. No puede haber 2 productos con el mismo nombre en el fichero.
  - En el método anterior tenéis que capturar las excepciones relacionadas con ficheros **FileNotFoundException y IOException y lanzar una TiendaException** con un mensaje adecuado**.**
  - **Tambien hay que capturar NumberFormatException y ArrayIndexOutOfBoundException y lanzar una TiendaException** con un mensaje adecuado**.**
- Dentro de vuestra clase **principal**, además de añadir los opciones para guardar y cargar al menú, tenéis que capturar las **InputMismatchExcepcion y TiendaExcepcion** mostrando mediante un sout del error producido
- También en la clase principal después de cada llamada a un método de la clase tienda mostrar un sout con un mensaje tipo "**acción X ejecutada con éxito**".