eTicket

eTicket es un software para la gestión de productos, almacén y tickets de una tienda.
En el archivo zip adjunto a la tarea tienes un programa en Java que está sin terminar de implementar.

Diseña el programa en varias fases:
En cada una de las fases existe un programa que deberás analizar para ver qué tareas tienes que realizar para que funcione.

En las páginas siguientes, se detallan cada una de las fases:

Fase 1: (10 puntos)

Deberás hacer que funcione el programa ETicket1.java.
Para ello deberás implementar la clase Product de manera que el programa funcione correctamente.
Los nombre de los métodos a implementar en la clase los irás sacando del programa ETicket1.java, para ello analiza el programa y podrás obtener los nombres de dichos métodos. Crea una interfaz con esos métodos y haz que la clase producto implemente esa interfaz.
Aquí tienes información de lo que debería hacer la clase Producto:

```
/** Product
* Mantiene la información de un producto
* Información que mantiene:
* code: Código del producto (se debe autogenerar empezando en 1)
* brand: Marca del producto
* model: Modelo del producto
* tradeName: Nombre comercial del producto
* price: Precio del producto
* Nota:
* - Getters de cada uno de los campos
* - Debes implementar el método equals que indica si un producto
es igual a otro
* para ello se deben comparar los códigos de productos y así
determinar si son iguales o no
* - Se debe poder imprimir la información de un producto de la
siguiente forma:
* Ej:
* Código: 7
* Marca: C&A
* Modelo: G123456
* Nombre comercial: Vestido señora
```

```
* Precio: 119,95 €
*/
```

Fase 2: (40 puntos)

Deberás hacer que funcione el programa ETicket2.java.
Para ello deberás implementar la clase Stock de manera que el programa funcione correctamente. (Obviamente has tenido que implementar correctamente la Fase 1).
Los nombre de los métodos a implementar en la clase los irás sacando del programa ETicket2.java, para ello analiza el programa y podrás obtener los nombres de dichos métodos.
Crea una interfaz con esos métodos y haz que la clase stock implemente esa interfaz.
Aquí tienes información de lo que debería hacer la clase Stock:

```
/** StockInterface
* Mantiene información sobre el stock de la tienda
* Para facilitar las cosas se te indica como debe manejar el stock
* haciendo uso de un HashMap
* Por cada producto almacenado se tiene información de la cantidad
* disponible del mismo
* Debe poder:
* - Añadir producto y retirar producto del stock
* - Imprimir el stock de la tienda de la siguiente manera:
* Ej:
* **********************************
* Código: 7
* Marca: C&A
* Modelo: G123456
* Nombre comercial: Vestido señora
* Precio: 119,95 €
* Stock disponible: 10
* **********************************
* Código: 1
* Marca: Levis
* Modelo: A123456
* Nombre comercial: Levis 501
* Precio: 79,95 €
* Stock disponible: 10
*
* Al añadir producto recibirá el producto y la cantidad a añadir
* Al retirar producto recibirá el producto y la cantidad a retirar.
* Si el producto no existe deberá lanzar una execpción
(ProductDoesNotExists)
```

```
* Si no existe suficiente stock para retirar deberá lanzar una
excepción (NotEnoughStock)
* En el caso de que el producto no exista deberá lanzar una
excepción
* indicando de que el producto no existe.
* En el caso de que no existan suficientes existencias deberá lanzar
* una excepción indicando de que no hay suficiente stock.
*/
```

Fase 3: (50 puntos)

Deberás hacer que funcione el programa ETicket3.java.
Para ello deberás implementar las clases TicketLine y Ticket en ese orden de manera que el programa funcione correctamente. (Obviamente has tenido que implementar correctamente las dos fases anteriores).

Los nombre de los métodos a implementar en las clases los irás sacando del programa ETicket3.java, para ello analiza el programa y podrás obtener los nombres de dichos métodos.

Crea una interfaz con los métodos necesarios para la clase TicketLine y haz que la clase TicketLine implemente esa interfaz.
Aquí tienes información de lo que debería hacer la clase TicketLine:

```
/**TicketLine
* Mantiene información de una línea de un ticket
* Una línea contiene información del producto y la cantidad comprada
del mismo
* Debe poder:
* - Devolver el producto de la línea
* - Devolver la cantidad comprada del producto
* - Modificar la cantidad comprada del producto
* - Devolver el precio total de la línea (Cantidad*Precio del
producto)
* - Imprimir la línea del ticket de la siguiente manera:
* ------------------------ 50 --------------------      --- 10 ---      --- 10 ---
* Levis 501                                      ( 1 x      79,95 € ) =      79,95 €
*/
```

Crea una interfaz con los métodos necesarios para la clase Ticket y haz que la clase Ticket implemente esa interfaz.
Aquí tienes información de lo que debería hacer la clase Ticket:

```
/**Ticket
```

```
* Mantiene información de un Ticket de compra
* Debe almacenar el código del ticket que se genera automáticamente
* empezando en 1, la fecha del ticket y las compras realizadas
* Para facilitar las cosas se te indica como debe manejar los
productos
* del ticket haciendo uso de un HashMap (Obviamente debes haber
implementado
* la clase TicketLine antes que esta)
* Debe poder:
* - Obtener la fecha del ticket
* - Obtener el código del ticket
* - Añadir una cantidad de un producto determinado a la compra
* - Quitar una cantidad de un producto determinado de la compra
* Si no existe el producto debe lanzar una excepción
(ProductDoesNotExists)
* Si la cantidad de productos a retirar de un producto es mayor a la
* que existe en el ticket debe lanzar una excepción (NotEnoughStock)
* - Obtener el total de la factura de la compra
* - Imprimir el ticket de la siguiente manera:
* ------------------------ 50 ---------------------       --- 10 ---       --- 10 ---
*                                      Levis 501 ( 1 x      79,95 € ) =      79,95 €
*                                 Camisa strech fit ( 2 x      49,95 € ) =      99,90 €
*                            Zapato caballero piel ( 3 x      99,95 € ) =     299,85 €
*                                           Total 479,70 €
*/
```