NOMBRE:.....
CURSO Y GRUPO:..... **FECHA: 7/03/2019**

Desarrollo de Aplicaciones Web
Java III – La caja del supermercado

1. Realiza un programa en lenguaje Java que imite la labor llevada a cabo en la caja de un supermercado. Debes realizar el ejercicio cumpliendo uno a uno los siguientes apartados:

a. (1,76 puntos) Existe un carro lleno de artículos. Debes crear un supertipo y varios subtipos para organizar los artículos del carro. Por ejemplo, puedes tener un supertipo llamado "Articulo" y varios subtipos llamados "LimpiezaYHogar", "ProductoFresco", "Lacteos", etc.…. De cada uno de esos tipos tendrás que crear los artículos del carro de la compra. Debes crear un mínimo de 5 artículos y 5 categorías de artículos (los subtipos descritos en este mismo párrafo). Cada artículo tiene, al menos, las siguientes características, que debes implementar en tu programa:
   a. Identificador, algún dato numérico parecido a un código de barras.
   b. Nombre
   c. Cantidad de artículos del mismo tipo
   d. Precio unitario

   Además, para cada subtipo (categoría de artículos) debes crear una propiedad específica que pertenezca sólo a ese tipo, piensa un poco.

b. (0,59 puntos) Todos las categorías de artículos deben implementar, obligatoriamente, un método llamado "public String getDescripcion()". En este método se creará y se devolverá una cadena de texto descriptiva del artículo, siendo algo parecido al conocido método "toString()". Eres libre de llevar a cabo este apartado a través de "Interface" o método abstracto.
c. (0,59 puntos) Los diferentes artículos se deben crear por ti, a través de lenguaje Java. Con ellos debes crear una lista de artículos, de forma que se simule la cola de artículos típica que se pone ante una caja de supermercado. Recapacita sobre el lugar en el que debes crear esta lista de supermercado. Sería una buena idea crear una clase de nombre "Cliente" y que ese cliente tuviese su lista de artículos. También te va a ser necesario crear una clase que simule ser la "Caja". La Caja también tendrá una colección de artículos, que corresponderá con los artículos que ya han pasado del Cliente a la Caja. La colección de artículos de la Caja debe ser implementada con un elemento de tipo HashMap.
d. (0,59 puntos) Debes crear un método que extraiga artículos de la lista de artículos del cliente y que los coloque en el interior del elemento HashMap de la caja. Debe existir un método, en la caja, que lea todos sus artículos y obtenga una cantidad total de dinero a pagar (Cantidad de artículos del mismo tipo multiplicada por el precio unitario).
e. (1,18 puntos) Debes simular que uno o varios artículos puedan eliminarse de la caja, simulando el hecho de que un cliente desee, en el último momento, no llevarse a casa dichos artículos. Para simular esto debes recorrer todos los artículos de la caja, mostrar uno a uno al cliente y preguntar si desea devolverlo.
f. (0,59 puntos) Cuando un artículo es rechazado (eliminado de la caja) debes guardar en dicho artículo un campo de tipo String en el que almacenes la fecha y hora en que esto ha ocurrido. En el formato de la fecha-hora debes Utilizar SimpleDateFormat y un formato "dd-MM-yyyy HH:mm:ss".
g. (1,18 puntos) Debes crear un fichero de propiedades en tu aplicación, que contenga el nombre del supermercado. Al principio del programa debes imprimir en consola el nombre anterior, obtenido a través de la clase Properties.
h. (1,76 puntos) El supermercado tiene un "SupervisorCajas", es un puesto desempeñado por un empleado o empleada y que debe ser notificado, con el modelo de delegación de eventos y listeners, cuando un artículo es devuelto (rechazado finalmente en caja). Debes crear todo lo necesario para llevar a cabo este punto.
i. (1,76 puntos) Ha llegado el momento de pagar los artículos que se desean comprar. La "Caja" debe conocer la cantidad que se debe pagar y debe pedir un dato numérico al cliente, para que indique la cantidad (en euros y en efectivo) con la que desea pagar. Por ejemplo, si la cuenta asciende a 56,04 € el cliente puede querer pagar con 56,50 €. Sin embargo, si la cantidad que indica el cliente no es suficiente para abonar toda la cuenta, se debe disparar una excepción. La excepción se puede llamar "CashInsuficienteException". Debes utilizar las instrucciones "try…catch", "throw" y "throws" necesarias para implementar este apartado. La detección de la excepción y las instrucciones "throw" y "throws" deberían estar en un método de la caja, llamado "pagar()". Existirá un método "main()" que, en un momento determinado, debe llamar a "pagar()" y comprobar si se produce la excepción.