

Laboratorio L06 y L07. Herencia

Objetivos

- Trabajar conceptos de POO en Java: herencia, polimorfismo, clases abstractas, interfaces.

Herramientas a utilizar

- *Eclipse* y *Javadoc*

Entrega

Diagrama UML con las clases del laboratorio, fichero **.zip** con todo lo solicitado en el laboratorio: exportación del proyecto eclipse (código y documentación). Recuerda la nomenclatura de los diferentes elementos:

- Nombre del proyecto Java: *apellidoNombre_L06*
- Nombre del archivo a entregar: *apellidoNombre.zip*
- La entrega debe ser individual y a través de eGela
- La entrega de los laboratorios 6 y 7 se hará en conjunto.

L06: descripción de las principales clases de la aplicación

El objetivo final del proyecto es desarrollar una aplicación para gestionar los productos del inventario de una tienda online. El desarrollo de la aplicación se basará en una descripción general de su funcionamiento.

Los **productos** (que estarán en el paquete `packProduct`) tienen las siguientes características: *codigo* (número entero), *nombre* (string), *peso* (número real), *cantidad* (número entero), *precio* (número real) e *IVA* a aplicar (número real entre 0.0 y 1.0). La **constructora** de productos tendrá un parámetro por cada atributo, excepto para el IVA. Además, debe ser posible **calcular el precio** de los productos con IVA y obtener un string con todas las **características** de los productos. Por otro lado, hay que poder decidir si hay que **retirar** o no un producto (porque está caducado, descatalogado...). Este método incluirá un String como parámetro (*motivo de la retirada del producto*). Método **isCheaper** que devolverá si el objeto que se utiliza para invocar el método es o no más barato que el producto que se le pasa como parámetro (IVA excluido).

Los productos lácteos, bebidas y frutas son productos **perecederos** que requieren *fecha de caducidad* (string). Entre los productos no perecederos se encuentran electrodomésticos y productos de limpieza. Los productos de limpieza tendrán también su *categoría* (String). Además, las frutas conservarán el *origen*; los lácteos, el *lote*; y las bebidas, la *graduación*.

Una vez establecido el IVA no se podrá modificar, pero su valor dependerá del tipo de producto. Con carácter general, el *IVA será de 0,21*, excepto para productos lácteos y frutas (**0,04**), y para bebidas, de **0,10**.

Generalmente, la causa de retirada de los productos perecederos será que su *fecha de envío* sea igual o anterior a la recogida como parámetro¹, excepto en los productos lácteos, en los que se calculará si se trata del propio lote. Los electrodomésticos se retirarán cuando su *nombre* coincida con el String dado como motivo y en los productos de limpieza cuando el parámetro dado coincida con la categoría.

Además, tendremos una clase llamada `Stock` (del paquete `packStock`) que tendrá una lista de productos (`productList`), de la clase `ArrayList`. Esta clase tendrá los siguientes métodos:

1. El constructor, que inicializará la lista de productos
2. `addProduct`: dado un producto, si no está en la lista de productos, lo añadirá. El método devolverá `true` si el producto se añade y `false` si ya estaba en ella.
3. `obtainPos`: dado un código de producto, devolverá la posición que ocupa dicho producto en la lista que lo contiene o -1 si no está. Este será un método de ayuda.
4. `removeProduct`: dado un código de producto, este método elimina el producto que tiene dicho código. Si no hay ningún producto con el código dado, escribirá un mensaje de error en la salida estándar.
5. `productQuantity`: devuelve la cantidad de productos diferentes que hay en stock.
6. `quantityType`: dado el tipo de un producto, devuelve la cantidad de productos diferentes de ese tipo que hay en el Stock
7. `toString`: devuelve un String con la lista de todos los productos en stock.
8. `itemsToRemove`: dado un criterio de retirada (String), indica si un producto debe retirarse o no.

¹ El método `compareTo` de String devuelve un número entero que indica si la fecha es menor (negativa), igual (0) o mayor (positiva) que la fecha dada.