



Una tienda de artículos de ropa y accesorios decide implementar una aplicación para llevar un control de sus artículos en stock. En dicha aplicación almacenará cierta información. De todos los artículos almacenará su referencia, breve descripción, stock y precio. Los artículos pueden ser accesorios o prendas de ropa. De las prendas de ropa se desea guardar además su talla, que en el caso de las prendas de la parte de arriba (tops) podrán tener el valor XS, S, M, L o XL y las de la parte de abajo (bottoms) podrán tener el valor 32, 34, 36, 38, 40, 42 o 44.

La referencia de los artículos estará formada por 10 caracteres alfanuméricos de forma que los tres primeros sean tres dígitos que indiquen el tipo de artículo: accesorio (286), top (412), falda (612) o pantalón (614), seguidos de un guión, después un número de 3 dígitos que será asignado de forma automática y correlativa por el sistema (es decir, 001 para el primer artículo, 002 para el siguiente, ...), seguido de otro guión y, por último 2 números que se corresponden con la talla, que en el caso de las tallas no numéricas serán: XS (01), S (02), M (03), L (04) y XL(05). Si un artículo no tiene talla los dos últimos dígitos de su referencia serán 00.

Así, por ejemplo, si el primer artículo añadido es una falda de la talla 38 su referencia sería 612-001-38 y si después añadimos una camisa de la talla M, su referencia sería 412-002-03.

Cuando se da de alta un artículo nuevo se debe indicar su descripción, su precio, el número de unidades disponible (stock) y, en el caso de aquellos que tengan talla, su talla. El sistema generará la referencia correspondiente. Además se implementarán los métodos vender que restará en una unidad el stock de un artículo, siempre que el stock sea mayor que 0, y reponer que incrementará el stock de un artículo en la cantidad indicada como parámetro.

Aunque esta tienda no admite devoluciones sí permite el cambio de talla de un artículo por lo que aquellos artículos que tengan talla implementarán una interfaz llamada iCambiaable que tendrá un único método devolver. Este método incrementará en una unidad el stock del artículo.

Para poder realizar cambios de talla las prendas de ropa implementarán un método estático (de clase) llamado cambioTalla que recibirá una referencia y una nueva talla y devolverá la nueva referencia con la parte correspondiente a la talla cambiada (de recibir una talla que no se corresponde con el tipo de artículo devolverá null)

Dispondremos de un programa llamado Tienda que tendrá la clase main y que almacenará en un array un máximo de 50 artículos. Este programa mostrará un menú con las siguientes opciones:

1. Nuevo artículo: preguntará de qué artículo se trata ofreciendo como opciones (accesorio, top, pantalón o falda) y pedirá los datos necesarios en función del tipo de artículo, para después añadirlo al array. Si no hubiese espacio en el array lo indicará con un mensaje antes de pedir ningún dato.
2. Listar artículo: mostrará un listado con todos los artículos, mostrando uno por línea con toda su información (referencia, descripción, stock, precio y talla si la tuviera)
3. Vender: preguntará por la referencia del artículo. Si el artículo no existe, lo indica mediante un mensaje. Si existe pero no hay stock, lo indica con otro mensaje. Si existe y hay stock, llama al método vender del artículo, indica un mensaje avisando de que la venta se ha hecho correctamente y muestra la información del artículo con el stock actualizado.

4. Reponer: pregunta por la referencia del artículo. Si existe, pregunta por la cantidad que se va a reponer y aumenta el stock en esa cantidad. Si no existe, lo indica con un mensaje.
5. Cambiar talla: preguntará por la referencia del artículo que se va a devolver. Si el artículo existe y es un artículo de ropa (con talla) entonces pide la nueva talla que se quiere llevar. Se llamará al método `cambioTalla` para obtener la nueva referencia y se comprobará si hay artículos y stock que se correspondan con esa referencia. Si lo hay, se llama a los métodos `devolver` (del artículo que se va a devolver) y `vender` (del artículo que se quieren llevar).
6. Salir: terminará la ejecución del programa. Mientras no se llame a esta opción se volverá a mostrar el menú y el programa seguirá en ejecución.

Se pide:

- Diseñar el diagrama de clases para este sistema, agrupando elementos (atributos y métodos) comunes evitando duplicar elementos. Para ello se podrá añadir clases intermedias o abstractas si se considera oportuno. Se deberá entregar la representación de dicho diagrama por escrito.
- Implementar las clases descritas ajustándose al esquema diseñado en el apartado anterior. Se implementarán los métodos que se consideren necesarios y se valorará el correcto uso de los modificadores.

Para aprobar el examen será imprescindible que éste no tenga errores de compilación.



NOMBRE: _____

DIAGRAMA DE CLASES

Criterios de calificación			
Apartado	Criterio	Max	Puntos
Compilación	Si el programa no compila la nota máxima del examen será de 2 puntos		
Diseño del diagrama de clases	Elección adecuada de las clases añadiendo aquellas necesarias para evitar duplicidad	1,25	
Implementación del diagrama de clases	Correspondencia del diagrama con su implementación	0,25	
	Correcto uso de los modificadores	0,25	
	Uso correcto de clases abstractas e interfaces	0,5	
	Implementación de los atributos y métodos necesarios para cumplir con las especificaciones descritas	0,5	
	Implementación y uso de atributos y métodos estáticos	0,5	
	Correcto funcionamiento de los métodos de las clases siguiendo las especificaciones del enunciado.	1,5	
Programa principal	Organización del programa principal usando los métodos necesarios para tener un código claro y legible	0,5	
	Se añade artículo correctamente	1,5	
	Se muestran los artículos	0,5	
	Se realiza una venta correctamente	0,5	
	Se reponen los artículos correctamente	0,5	
	Se hace una devolución correctamente	1,5	
	Funcionamiento del menú	0,25	
Total		10	