

Materia: Taller De Programación 1
FINAL

Leé el enunciado con cuidado y por lo menos dos veces para entender qué se pide y qué estructuras son necesarias para resolver lo pedido basándote en el diseño UML incompleto que figura más abajo y el proyecto entregado junto con este enunciado. Dentro del proyecto encontrarás clases incompletas, métodos con comentarios y comentarios sueltos de métodos que debes implementar. Estos comentarios dan información importante para la resolución (parámetros, validaciones, etc)

Pensá bien la estrategia de resolución antes de comenzar el desarrollo. **El objetivo de este examen es evaluar la correcta aplicación de los conceptos y el dominio de las técnicas vistos durante la cursada:**

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Correcta aplicación de Herencia y Polimorfismo, incluyendo interfaces.
- Correcta aplicación ArrayList, Vectores, Pilas y/o Colas, siempre que se necesiten, y aprovechando las interfaces e implementaciones provistas.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, utilizando *setters* y *getters* sólo cuando corresponda y con la visibilidad adecuada.
- Uso adecuado de excepciones (lanzamiento y captura) en la creación de objetos y sus validaciones.
- Dominio de la herramienta utilizada durante el cuatrimestre para el desarrollo de proyectos Java (Eclipse), en especial lo referido a la importación y exportación de librerías y proyectos.

La mala aplicación de cualquiera de estos puntos hará que el examen sea desaprobado.

No deben modificarse las estructuras ni los métodos provistos, salvo en los casos indicados.

La entrega de proyectos con Warnings será penalizada.

Enunciado

La empresa para la que trabajamos está por lanzar un sistema de control de compras en línea. Este equipo dispondrá de productos que podrán ser pedidos para su posterior compra en base a la selección del usuario. En una primera etapa se nos encarga el desarrollo de la selección y armado del pedido de productos.

La aplicación tendrá una cierta capacidad de productos que se indicará al momento de iniciar el sistema (instanciación) pero esta cantidad no podrá superar la máxima de 30 productos, ni podrá tener menos de 5. Adicionalmente el sistema tendrá un nombre que no podrá ser nulo ni vacío.

Cada producto, que puede ser simple o por bulto, poseerá un ID numérico (entero mayor a 0), NOMBRE (no nulo ni vacío), una DESCRIPCIÓN (no nula ni vacía) y un artista (no nulo). Adicionalmente si es un producto que se vende por bulto tendrá la cantidad de unidades que posee. Cada producto debe ser "Pedible, Identificable y Mostrable" según se ve en el DC del sistema.

La marca, que será Mostrable, posee un NOMBRE (no vacío ni nulo) y un ORIGEN (no nulo).

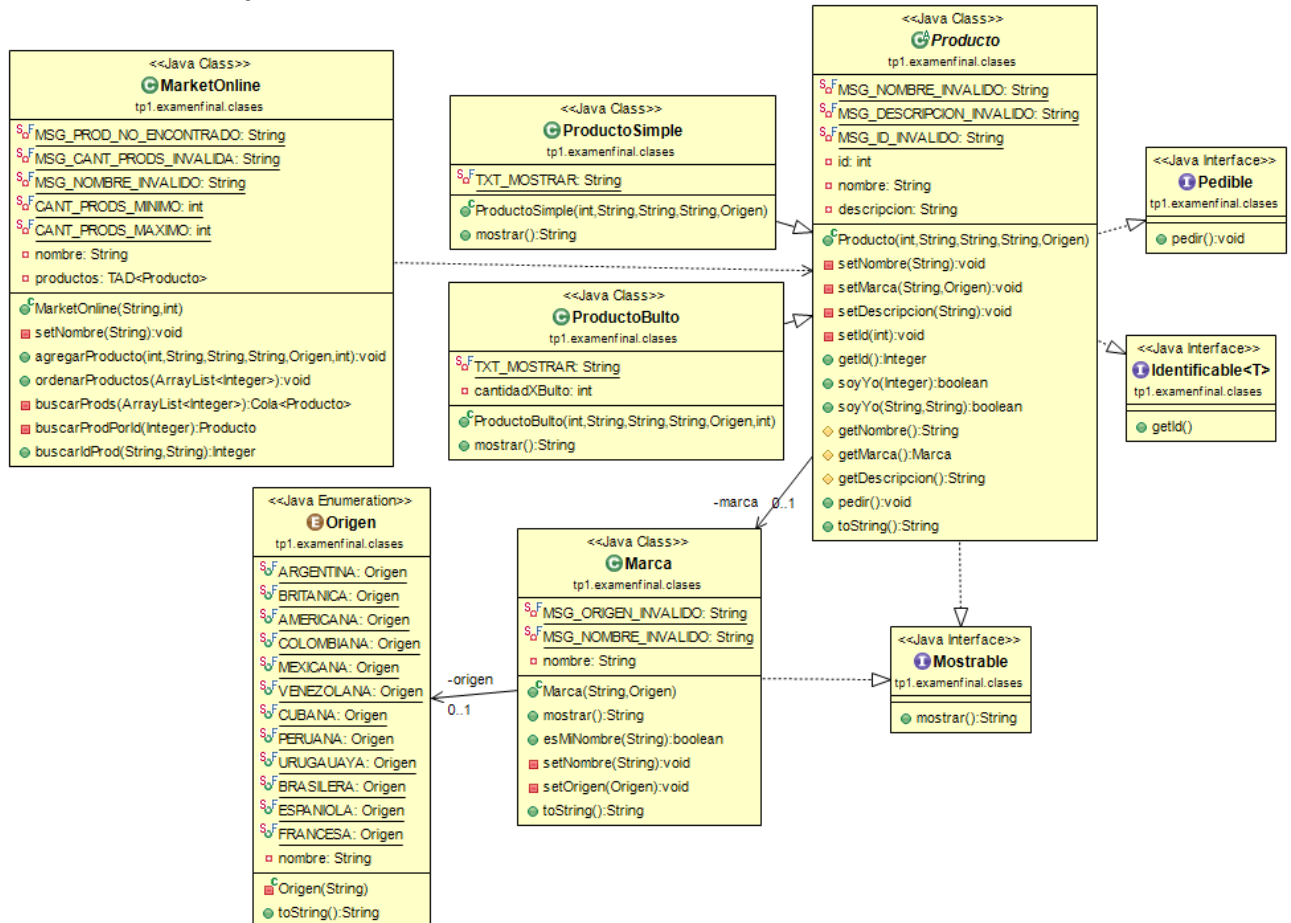
Al sistema se le deberán poder agregar productos indicando el id, el nombre y descripción del producto, así como el nombre y el origen de la marca, y por último la cantidad de unidades por bulto (1 si es un producto simple, 2 o más si se vende por bulto). Además el sistema deberá permitir buscar Ids de productos en base al nombre del producto y nombre de su marca (devolviendo un entero o lanzando una excepción en caso de no encontrar el producto). Por último el sistema deberá poder pedir los productos cuyos ids se le indiquen como parámetro.

Los productos agregados al sistema deben ser almacenados de forma tal que en funcionamiento sean más fácil encontrados los más recientemente pedidos (Aplicar la TAD que crea conveniente)

El pedido de productos deberá armar un listado (aplicando la TAD que corresponde) para luego pedirlos uno a uno, y volver a agregarlos en la lista de productos del sistema.

Materia: Taller De Programación 1
FINAL

Diagrama de clases incompleto



NO SE DEBEN IMPLEMENTAR LAS CARGAS POR TECLADO, la prueba del sistema se realizará mediante la ejecución del método main(...) de la clase Main, esta clase no debe ser modificada por el alumno **a excepción del manejo de errores**. De ser modificada fuera de lo indicado el examen quedará descalificado automáticamente. LA EJECUCIÓN DEL MÉTODO main() DE ESTA CLASE NO DEBE ARROJAR NINGUN ERROR NO CONTROLADO EN LA CONSOLA. Tras la ejecución de dicho método se deberá visualizar por consola el siguiente resultado:

```

----- Se crea el mercado -----
----- Se agregan productos al mercado -----
Producto agregado (blancaflor pizza - blancaflor ID: 1)
Error agregando producto al mercado: ID del producto inválido
Producto agregado (cocinero mezcla - cocinero ID: 2)
Error agregando producto al mercado: Nombre del producto inválido
Error agregando producto al mercado: Descripción del producto inválido
Error agregando producto al mercado: Origen de la marca inválida
Producto agregado (casancrem - casanto ID: 6)
Producto agregado (quesabores - sancor ID: 7)
Producto agregado (cantidadXBulto invalida - cantidadXBulto invalida ID: 8)
Producto agregado (yogs - sancor ID: 9)
Producto agregado (yogurísimo - la serenísima ID: 10)
Producto agregado (jugos exprimidos - Citric ID: 11)
Producto agregado (capeletinis - giacomino ID: 12)
----- Se compran productos -----
    
```

Materia: Taller De Programación 1
FINAL

```
----- Se genera Lista de IDs a pedir -----  
Error seleccionando producto: Producto no encontrado  
Error seleccionando producto: Producto no encontrado  
Error seleccionando producto: Producto no encontrado  
----- Se ordenan los productos con los IDs deseados -----  
-----  
El mercado SupermercadoORT ordenara los productos seleccionados  
Prod Id 77 no encontrado  
Ordenando el producto yogs, Descripción: yogur bebible, con marca Nombre: sancor Origen: Argentina  
Ordenando el producto yogurísimo, Descripción: yogur bebible, con marca Nombre: la serenísima Origen:  
Argentina  
Ordenando las 11 unidades del producto quesabores, Descripción: queso en hebras, con marca Nombre:  
sancor Origen: Argentina  
Ordenando el producto casancrem, Descripción: queso untable, con marca Nombre: casanto Origen: Argentina  
----- FIN -----
```

Completá y agrega las clases o estructuras que consideres necesarias. Completá todos los lugares indicados con el comentario “//TODO: Completar”, prestá atención a los JavaDocs de los métodos y atributos, y donde consideres necesario. Implementá interfaces o extendé las clases que consideres.

Para realizar el examen

- descargar el archivo TP1-FinalMar2020-2.zip. del aula virtual
- Importar en Eclipse el proyecto descargado y renombrarlo **TP1_2020_MAR_F2_ApellidoNombre**. Por ejemplo: TP1_2020_MAR_F2_PerezJuan
- Al finalizar el examen, exportarlo en un archivo ZIP y subirlo en el aula virtual en el link correspondiente al FINAL, el archivo a subir debe llamarse igual que el proyecto.

Importante. El proyecto entregado debe compilar (no debe contener errores) y no debe poseer ningún “Warning” en ninguna clase, interfaz o enum – borrar o comentar líneas con warning o errores no está bien.