



Sesión 9. Trabajo con excepciones

1 Instrucciones

- Descarga el proyecto inicial del campus virtual, descomprímelo y renómbralo.
- Abre el eclipse e importa el proyecto. Parte del proyecto ya está implementado.
- **En esta sesión se completará la funcionalidad básica y se añadirá la gestión de excepciones.**
- En la próxima sesión de laboratorio, se añadirá la E/S. Es importante tener acabada esta sesión 10 antes de la próxima sesión de laboratorio.

2 Descripción

Se desea diseñar e implementar un sistema de gestión de ventas y pedidos de un quiosco (NewsStand) que vende publicaciones de dos tipos: periódicos (newspaper) y revistas (magazine), de los que se guarda la siguiente información:

- **name** (cadena) Nombre del producto
- **stock** (entero) Número de ejemplares sin vender
- **sold** (entero) Número de ejemplares vendidos

Los periódicos son publicados diariamente y para las revistas se debe guardar información sobre su periodicidad:

- **regularity** (entero) Periodicidad, o número de días entre dos ediciones consecutivas

3 Interacción con el usuario

La aplicación ofrece un menú de usuario con las siguientes opciones:

1. *Cagar productos de un fichero*
Añade a la lista de productos del newsStand todos los contenidos en un fichero de texto. El nombre del fichero se le pregunta al usuario.
2. *Mostrar publicaciones*
Muestra la lista de publicaciones
3. *Añadir una publicación*
Pide al usuario toda la información necesaria para crear una nueva publicación. Una publicación se identifica por el nombre y no puede haber publicaciones repetidas.
4. *Eliminar una publicación*
El programa pregunta al usuario por el nombre de la publicación.
5. *Generar publicaciones*
Crea una lista con los pedidos necesarios. Cada pedido contendrá el nombre de la publicación y el número de ejemplares a pedir. Ver más adelante.
6. *Guardar pedidos en un fichero*
Se le pide al usuario el nombre del fichero donde guardar la información.



7. Importar de zip

Similar a la opción 1, pero carga de un fichero comprimido .gz todas las publicaciones y elimina las anteriores.

8. Exportar a zip

Guarda en fichero comprimido .gz todas las publicaciones.

4 Generación de pedidos.

A partir de la lista de publicaciones, se debe generar una lista de pedidos, con un pedido por cada publicación que tenga menos de 10 ejemplares de stock.

Para cada pedido se guarda el nombre de la publicación y el número de ejemplares a pedir, que se calculará de forma diferente según el tipo de publicación y el stock del mismo:

- a) **Periódico:** Se piden tantos ejemplares como los vendidos más el doble de los ejemplares en stock.
- b) **Revista:** Si el stock es menor que 5 se piden 20 ejemplares. Si es mayor o igual que 5, dependerá de la periodicidad. Si es semanal, se piden tantos ejemplares como los vendidos y en otro caso, se piden tantos como los vendidos más los ejemplares en stock.

El fichero de entrada contiene una línea para cada publicación. Las líneas tienen una representación textual de las publicaciones y cada campo está separado del siguiente por el carácter tabulador ('`\t`'). Se proporciona el fichero `publicaciones.txt` con el siguiente contenido por línea:

`<tipo de publicación> \t <Nombre> \t <stock>\t <vendidos> [\t <periodicidad>]`

periodico	Nueva España	14	30	
periodico	El Mundo	4	10	
revista	Hola	14	30	7
revista	PCWord	14	30	30
revista	Diez Min	4	10	7
revista	El Mueble	4	10	30
revista	Muy Inter	8	20	7
revista	Quo	8	10	30

5 Ejercicio

En primer lugar, se debe completar la aplicación siguiendo los siguientes pasos:

1. Analice el código y compárelo con el diagrama UML (realizado previamente). Es necesario familiarizarse con el diseño y entender el propósito de las clases y métodos.
2. Haga que funcione la aplicación adecuadamente.
3. Añada control de errores y su tratamiento.

5.1 Hacer que el programa funcione

1. Ejecute el programa. El código compila y ejecuta, pero rompe la ejecución con excepciones `NotYetImplementedException`.



2. Empiece haciendo funcionar la opción 1 de cargar los datos de un fichero. Complete la clase PublicationParser para conseguir una lista de publicaciones List<Publication> a partir de una lista de cadenas List<String> devuelta por la implementación actual (falsa) de la clase FileUtil.
3. Complete los métodos getPublications() y createOrders() de la clase NewsStand.

5.2 Añadir control de errores

Ahora ya tenemos una funcionalidad básica, pero no hay gestión de errores. Tenemos que preparar el código para manejar algunos errores relativos a cada operación. Se deben contemplar al menos las siguientes situaciones de error:

1. Gestionar **errores lógicos** (por fallos de los usuarios finales de la aplicación)
 - Los nombres de ficheros deben contener, al menos, 5 caracteres. Se considera también un error lógico cuando necesita leer y el fichero no existe.
 - Añadir una publicación por teclado con nombre repetido
 - Eliminar una publicación cuyo nombre no existe.

Si ocurre alguno de estos errores, el programa debe informar al usuario del error, mostrando un mensaje en la consola **System.err**. Se debe mostrar el menú otra vez para que el usuario pueda seguir realizando operaciones.

- Añadir una publicación con nombre repetido a partir de la lista de publicaciones cargada de un fichero.

En este caso, la publicación no se añade (pero se continúa añadiendo el resto de las publicaciones). Debe enviarse un mensaje de error a un **log**.

Se debe crear una excepción propia llamada NewsStandException para gestionar estos errores.

2. Gestionar **errores de programación o del sistema**
 - Cualquier excepción RuntimeException
 - Cualquier excepción IOException

Si ocurre alguno de estos errores, el programa debe finalizar la ejecución de manera controlada, informar al usuario de un problema interno y un mensaje de error a un **log**.

3. Gestionar **errores de carga del fichero**:
 - Líneas en blanco
 - Número de campos incorrecto
 - Tipo de publicación incorrecto (diferente de periódico o revista)
 - Formato del número erróneo

Si ocurre alguno de estos errores, la línea es inválida y tiene que ser descartada (ignorando la publicación, pero continuando el proceso). Debe enviarse un mensaje de error a un **log**.

Se debe crear una excepción propia llamada InvalidLineFormatException para gestionar estos errores. La excepción debe almacenar al menos el número de línea que provoca el error.