Sesión 10. Adición de capacidades a la sesión 9 de laboratorio

Esta sesión es la continuación de la sesión previa. Realiza una copia del proyecto sesión9 y renombra el proyecto como sesión10.

Descripción

En esta sesión vamos a completar toda la funcionalidad de la aplicación. Existen algunas partes pendientes:

- La aplicación debe listar todos los productos ordenados por nombre y por número de ejemplares vendidos.(Lo haremos en la sesión 11)
- Las opciones "Cargar de un fichero" y "Guardar a un fichero" tienen que leer de un fichero y escribir en un fichero. Ambos serán indicados por el usuario.
- Las opciones "Exportar a un fichero" e "Importar a un fichero" tienen que hacerlo con ficheros comprimidos gzip.

Lectura de ficheros

Las clases `FileUtil` y `ZipFileUtil` deben ser implementadas para la lectura real de los ficheros (de texto o de bytes). Cree ambas implementaciones y ponga atención a las excepciones producidas y cómo debe manejar cada una de ellas:

- La excepción cuando el fichero no se encuentra debe ser manejada como un fallo del usuario (ha proporcionado un nombre de fichero erróneo), y debe seguir la ejecución pidiendo una nueva opción del menú.
- El resto de excepciones genéricas, `IOExceptions` deben ser manejadas como errores del Sistema, enviando un mensaje al usuario por consola que indique la finalización del programa y mostrando el mensaje y la traza de la pila (en el log) antes de la finalización.

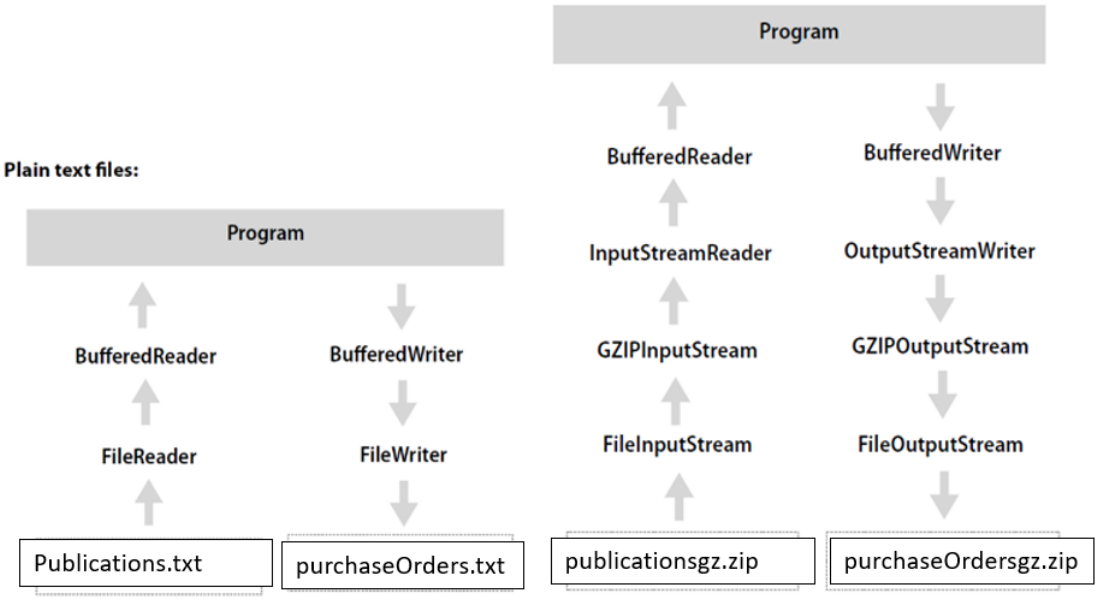
Escritura en ficheros

Las mismas clases `FileUtil` y `ZipFileUtil` deben también ser capaces de guardar datos a un fichero (de bytes o de texto).

Antes de escribir, se necesita obtener una lista de cadenas para todos los pedidos. Esta serialización (en este caso una representación textual) se hace a través de la clase `OrderSerializer`.

Nota: Como ambas implementaciones son muy similares, puede aplicar herencia para evitar la duplicación de código.

Cadena de procesamiento para FileUtil y ZipFileUtil



Ampliación: Registro de errores (log) en un fichero

El sistema de registro de errores básico actual (log) graba en el flujo `System.err` pero necesitamos que el de registro de errores se guarde en un fichero de salida.

Hay que tener en cuenta que cada vez que se llama al método `logger.log(..)` debe añadirse una nueva línea al fichero existente.