VERSIÓN 3 Librería

CONCEPTOS A TRATAR

- Ficheros
- Excepciones
- Herencia
- Serialización
- Programación en capas

INTRODUCCIÓN

Para el desarrollo de esta versión, partimos del código optimizado de la anterior, sustituyendo la colección de objetos(arraylist) libro por un archivo. Haremos la gestión de ficheros binarios, apoyándonos en un arrayList. Indicar que esta no es la mejor forma de tratar los ficheros, lo que pretendemos con esta aplicación, es que el alumno conozca lo más básico de la gestión de los mismos en C#(p.e la serialización).

Se valorará introducir una clase Errores en la capa de Negocio, que gestionará todas las excepciones que se produzcan en los ficheros. En la próxima versión de bases de datos si estará la clase Errores.

Se cambiará la codificación interna de los tres métodos de la clase Estanteria para realizar la gestión de los ficheros, pero en cuanto a su declaración y parámetros, no sufrirán ningún cambio. Esto llevará al alumno, a la idea de crear una Interface con los métodos de Estanteria, para que fuese implementada en las distintas versiones de la librería(arraylist, ficheros). En esta clase se añadirá la codificación de métodos de gestión de archivos, como son los clásicos abrir, leer, escribir y cerrar.

Al codificar la clase Estanteria para los distintos tipos de ficheros, en nuestro caso binarios, nos damos cuenta que hay una serie de métodos comunes en su gestión. En la codificación de esta aplicación en el lenguaje JAVA, se creó una nueva clase llamada **AccesoF** donde se encuentran los métodos primitivos de gestión de ficheros. En la siguiente versión 4 de Libreria, se creará la clase AccesoBD que gestionará el acceso a datos pero mendiante base de datos. El proyecto quedará dividido en tres capas, Vista, Negocio(modelo) y Datos.

La capa de Presentación(Vista) de la versión anterior sufrirá muy pocos cambios y utilizará las mismas clases.

EXPLICACIÓN DE CONCEPTOS

Se hará un repaso de las **excepciones**, **ficheros** y se profundizará en el **modelado del desarrollo de aplicaciones en capas** y la utilización de **patrones**.

Se repasará el concepto de **serialización** que aplicamos en la gestión de ficheros binarios.

DESARROLLO

- **Versión 3 Librería. Ficheros Binarios**

Versión 3 Librería

Diagrama de paquetes y clases:

CLASES (Versión 3 Librería)	
Vista	**Negocio**
Libreria	Estanteria Libro Mensajes Errores

Tabla 3.3 Clases Versión 3 Librería

Descripción de clases:

Clase Estanteria

Las propiedades de esta clase indican los ficheros(File) de trabajo junto a las rutas donde se encuentran y los objetos lectores/escritores utilizados para escribir y leer en los ficheros binarios.

Métodos
Interfaz pública

```
public Estanteria() // En esta versión se crea un constructor vacío
public boolean insertarLibro(Libro obLibro)
```

public boolean borrarLibro(String nombre)
public Libro buscarLibro(String nombre)
boolean Libro modificarLibro(Libro obLibro)
public ArrayList<Libro> cargarLibros()

Interfaz privada (métodos de Ficheros binarios)

private ArrayList<Libro> leerFichero()
private void escribirFichero(List<Libro> libros)
private void cerrarFichero()

El método constructor *Estanteria().Introducirlo sin código.*
El método *boolean insertarLibro(Libro libro),* inserta un libro en el fichero binario data.dat

El método *boolean borrarLibro(Libro libro),* borra un libro del fichero binario data.dat

El método *Libro buscarLibro(String nombre),* busca un libro dentro del fichero binario data.dat

El método *Libro modificarLibro(Libro obLibro), modifica* un libro dentro del fichero binario data.dat

El método *List<Libro>cargarLibros(),* devuelve el conjunto de libros del fichero binario data.dat.

El método List<Libro> leerFichero(), lee un fichero binario(data.dat) y nos devuelve todos los libros encontrados en un arrayList.

El método escribirFichero(List<Libro> libros), escribe en un fichero binario todos los objeto libros del arrayList.

El método cerrarFichero() cierra el fichero binario.

Clase Libro

Las mismas propiedades y métodos que las versiones anteriores.

Clase Mensajes

Las mismas propiedades y métodos que las versiones anteriores.

Clase Errores

Métodos
Interfaz pública

public Errores(Exception e)

El método constructor *Errores(Exception e)* construye un mensaje de error a través del tipo de excepción.

Clase Libreria

Las mismas propiedades y métodos que las versiones anteriores.

Métodos

La diferencia con las versiones anteriores, radica en que esta versión, el método de evento Load realiza la carga de los libros grabados en el fichero mediante el método cargarLibros() de la clase Estanteria.

CODIFICACIÓN

Descripción de tareas:

1.- El alumno realizará un proyecto Libreria-V3 que parte del código optimizado de la versión anterior. Se sustituirá la colección de objetos(arraylist) libro por un archivo binario. Se incluye una clase Errores en la capa de Negocio que gestionará todas las excepciones que se produzcan en los ficheros.

En esta tarea el alumno se enfrenta a la dificultad de tener que codificar métodos de gestión de archivos binarios, como son los clásicos abrir, leer, escribir y cerrar. También tiene que realizar el control de errores que se produzcan en la aplicación.