

Parte 1. SOLID

AUTORES

Cristopher Álvarez Martínez, Edgar Conde Nóvoa

OBJETIVOS

Comprender y saber aplicar los principios SOLID sobre una aplicación existente.

LA APLICACIÓN

Sea una aplicación que hace lo siguiente:

1. Pide dos nombres de ficheros al usuario: uno de entrada y otro de salida.
2. Transforma el fichero de entrada a un XML escribiéndose en el fichero de salida.

TAREAS

- A. Identifica las diferentes responsabilidades que existen en este programa. Haz una lista con ellas.
 - B. Refactoriza la aplicación para que:
 1. La interfaz de usuario pueda ser reutilizada en un proyecto que haga otra cosa dados dos ficheros (la interfaz se encarga de pedir los ficheros). ¿Qué patrón o patrones SOLID has empleado, cómo y por qué?
 2. El origen de datos (ahora ficheros) pueda ser distinto **y/o** el destino de los datos también. ¿Qué patrón o patrones SOLID has empleado, cómo y por qué?
 3. La transformación que se hace de la entrada pueda ser otra representación basada en texto cualquiera (no a XML). ¿Qué patrón o patrones SOLID has empleado, cómo y por qué?
 - C. Modifica la aplicación para que la salida se produzca por pantalla y no a fichero. ¿Tuviste que cambiar código existente a mayores que el método *main*? Describe brevemente la modificación.
-

A. RESPONSABILIDADES

1. Pedir 1 fichero a usuario
2. Leer líneas de fichero/origen.
3. Escribir líneas al fichero/destino.
4. Informar de error al abrir fichero (Y otros errores).
5. Transformar 1 línea TAB a XML / en un formato a otro.
6. Recorrer fichero de entrada, transformando y escribiendo en destino.

B. REFACTORIZACIÓN

CLASE	RESPONSABILIDAD
FileReader	Leer línea del fichero
FileWriter	Escribir línea al fichero
TabtoXMLTransformer	Transformar línea TAB a XML
Converter	Recorrer fichero de entrada, transformando y escribiendo en destino
ConverterApp	Informar de error al abrir fichero (Y otros errores).
FilesUserInterface	Pedir 2 ficheros al usuario

OCP (Open closed principle)		
CLASE	MODIFICACIÓN POSIBLE	PUNTO DE EXTENSIÓN
Converter	Cambio en el destino de las líneas	Interfaz Writer
Converter	Cambio en el origen de las líneas	Interfaz Reader
Converter	Cambio en el formato de una línea a otra	Interfaz Transformer

C. MODIFICACIÓN

Bastaría con crear un nuevo tipo de Writer que escribiese por consola las líneas, del estilo:

```
public class ConsoleWriter implements Writer {  
  
    @Override  
    public void write(String output) {  
        System.out.println(output);  
    }  
  
    @Override  
    public void close() {  
        System.out.close();  
    }  
}
```

De este modo en el método main() solo tendríamos que hacer un cambio que sería pasarle al constructor de Converter un objeto ConsoleWriter en vez de uno de tipo FileWriter, sin cambiar el código de ninguna clase.