
EXPLICACIÓN DEL PROYECTO

Introducción:

La idea del proyecto era simular una Luthería o Lutería, que no es ni mas ni menos que una fabrica de instrumentos musicales, cuya fabricación incluso hoy en día no esta completamente automatizada ya que, entre los instrumentos criollos, algunos detalles de fabricación requieren un detalle humano que no puede ser completamente automatizado.

De todas maneras, me tome la libertad de crear procesos y maquinas imaginarias que representan las estaciones por las que tienen que pasar algunos instrumentos luego de ser fabricados por manos humanas, para obtener el grado de terminación necesaria como para distribuirse y venderse, llegando únicamente hasta el momento en que estos pasan a un almacén donde se stockean y no tocando nada del proceso de venta.

El uso correcto de mi programa se encuentra especificado en el manual de uso.

Paso a detallar en que partes de mi programa utilice los temas requeridos para este Trabajo Practico.

Clase 15 – Excepciones

Utilizo Excepciones en la mayor parte del proyecto, en algunos casos usando Excepciones propias, (las cuales se encuentran en la carpeta Excepciones dentro de Entidades) y en otros casos algunas predeterminadas del lenguaje.

A continuación, marcare algunas de sus implementaciones (no todas ya que son bastantes, en especial entre los botones del form principal)

Implementación de excepciones (throws):

Propias:

- Clase Stock, Métodos ReStock y DeStock
- Clase Almacén, Método HaySuficienteMaterial

Predeterminadas:

- Clase Almacén, Sobrecarga de Constructor
- Clase Almacén, Método ValidarPasaje

Implementación de excepciones (try-catches):

- Clase Stock, Métodos ReStock y DeStock
- Formulario FrmRestock, Método btnGenerarPedido_Click
- Formulario FrmCrearInstrumento, Método btnContinuar_Click

Clase 16 - Test Unitarios

Utilizo Tests Unitarios con el proyecto LutheriaUnitTest, el cual testea las funcionalidades principales del Core de la clase Lutheria, la cual es la más importante del trabajo practico, ya que maneja todo.

Clase 17 - Tipos Genéricos

Utilizo tipos genéricos en varios métodos y en una interfaz, para ayudarme con el pasaje de un proceso al otro, es decir de cada una de los objetos que representan las máquinas y sus listas con los instrumentos que se encuentran en cada máquina en ese momento.

Implementación de Tipos Genéricos:

- Interfaz IProceso
- Metodo PasarA<T>, en todas las clases que implementan IListaDeInstrumentos (Lutheria, Almacén, Barnizadora, Limpiadora, Encordadora, Afinadora)

Clase 18 - Interfaces

Implemento 8 interfaces ya que cuando resultaron ser bastante útiles (estas se pueden encontrar en la carpeta Interfaces en Entidades)

Clases que implementan Interfaces:

- Violín
- Guitarra
- Flauta

Estas tres clases implementan interfaces que me obligan a declarar propiedades que demuestren sus estados de fabricación (ej. EstaBarnizado de la interfaz IBarnizable) y métodos que permitan cambiar esos estados, así también las utilizo a modo validación para verificar a que maquinas puede dirigirse un instrumento según si implementan la interfaz correspondiente a la maquina y los estados que implican que el instrumento paso por el proceso anterior (ej. Un violín solo puede ser afinado luego de ser encordado y para eso tiene que pasar por la Encordadora antes de la Afinadora y luego de ser barnizado, ya que no se podría barnizar con las cuerdas puestas)

- Lutheria
- Almacén
- Barnizadora
- Encordadora
- Limpiadora
- Afinadora

Estas clases implementan Interfaces que las obligan a tener una lista de instrumentos y métodos para pasar un instrumento de una maquina a otra (si bien Lutheria y Almacén no cuentan como máquinas, ambas tienen una lista de instrumentos para contener la etapa inicial y final del proceso de fabricación, es decir el Instrumento crudo y el guardado en Stock respectivamente)

De todas formas, dejo el diagrama de clases para que puedan revisar

Clase 19 - Archivos

Utilizo archivos mediante la Clase Archivos que puede encontrarse en Entidades.

Utilizo esta clase en el Form Principal.

Uso un archivo binario (data.bin) para guardar todos los datos de la instancia de la clase Lutheria que se genera al guardar el programa, de esta manera puedo cerrar el programa y cuando lo abra todo estará en la misma posición que se guardó, a excepción quizá del Visor de Detalles (ver en el manual de uso).

Uso un archivo de texto (almacen.txt) para poder mostrar como quedo el Almacén cuando se cerro el programa, es decir, mantener un inventario y no tener que abrir el programa para verlo.

Finalmente utilizo un archivo XML para guardar una lista de las direcciones o "paths" de los archivos que se utilizan en el programa de esta manera, se puede modificar que archivo binario se va a estar leyendo y en que archivo de texto se está guardando el inventario.

EN CASO DE NO ENCONTRARSE ALGUNO DE LOS ARCHIVOS DE CONFIGURACION, el programa mostrara una excepción al comenzar, pero solo debe cerrarse y el Programa continuara de manera predeterminada, como si fuera la primera vez que se ejecuta.

En el proyecto, dejo el archivo binario con el cual realice las pruebas mientras lo desarrollaba, pero pueden probar borrando este mismo si quieren ver como seria el programa desde 0.

Espero que el TP se ajuste a las normas de lo pedido ya que al principio fue medio complicado entender lo que se pedía.