READ ME

Proyecto 2 de Modelado y Programación:

Implementación de los patrones de diseño “Strategy”, “Proxy”, “Singleton”, “Iterator” y “Modelo-Vista-Controlador”.

Nombre del equipo:

“Lo que se dañó fue el haswer”.

Integrantes:

- Martínez Santana Diego Maximiliano - 320318429
- Suárez Peñaloza Luis Alberto – 320222337
- Villegas Maldonado César Ismael – 320108226

Justificaciones:

El programa que hoy les estamos entregando simula una base de datos de biblioteca, en donde puedes consultar existencias de libros mediante distintos métodos de búsqueda y apartarlos para después recogerlos en las instalaciones de la facultad (suponiendo que la facultad implementa este programa y que los pdf no son tan cómodos y populares como son ahora, más que nada para facilitarnos las cosas, porque intentamos hacer mil cosas para que se pudiera desplegar un pdf en nuestra aplicación y descargarlo, pero es bastante difícil usando JavaFX).

Para la ideación de este proyecto pensamos en implementarla usando un servidor real con una base sql perfectamente formada, pero debido a que pensamos también en que este proyecto debería de ser totalmente portable para que cualquiera de ustedes queridos profesores pudiera probar la app sin ningún problema, esto porque de por sí ya estamos causando bastantes problemas con haberlo implementado en JavaFX; entonces surgió la idea de usar árboles rojinegros, ya que como éste proyecto debería de ser pensado para ser “escalable”, nos dimos cuenta de que éstos tienen mucho mejor desempeño que un diccionario por ejemplo.

Strategy: Decidimos usar este patrón debido a que como la base de datos tiene distintos métodos de búsqueda, implementamos strategy para simular cada uno de esos métodos de búsqueda mediante comparadores definidos en la clase “BookComparators”. Este patrón está presente en las clases “SearchStrategy”, “SearchByName”, etc.

Proxy: Este patrón de diseño es un clásico ejemplo de un uso “apropiado” de la información en un intercambio dinámico de la misma entre un servidor y un cliente. Es por esto que usamos una arquitectura parecida a nuestro proyecto 1 de hace un mes. Está presente en las clases “UserInterface”, “User”, y “UserProxy”.

Singleton: Ideamos a nuestros libros de manera de que siempre haya manera de identificarlos fácilmente, esto porque es posible que haya colisiones entre nombres, y

pues en autores, categorías y editoriales siempre va a haber muchísimos ejemplares del mismo tipo si lo comparamos así. Es aquí donde entra Singleton. Singleton nos ayudará a preservar un id único para libro que se ingrese en la base de datos, y como los rojinegros están integrados con comparadores de la interfaz "Comparable" y aparte la comparación por defecto entre libros ya es entre ids, pues es más fácil tener un control sobre la información. Este patrón se encuentra en la clase "IdGenerator".

Iterator: Dada la facilidad de implementación de éste patrón, siempre lo incluimos para permitir un manejo estable y controlado de la información que fluye por todos lados. Este patrón está presente tanto en los árboles binarios como en las búsquedas de libros.

Anotaciones:

Este programa fue hecho con Visual Studio Code, por lo que el programa en sí ya viene compilado dentro de la carpeta "bin" del proyecto. Dentro de ella pueden ejecutar dos comandos para correr el programa. Uno se hace con "java library/Server 1234" para iniciar un pequeño servidor que implementamos, y el otro con "java --module-path /path/to/your/javafx-sdk-22.0.1/lib --add-modules javafx.controls,javafx.fxml library/LibraryClient, donde el "path to your javafx sdk" es la dirección en memoria de la sdk de javafx, esto porque el programa está hecho en javafx. En caso de que el segundo comando no funcione, pueden ingresar a visual studio code y correr el programa en la ventana "Run and Debug", siempre y cuando especifiquen en el archivo "launch.json" la ruta a su sdk de javafx.

La versión de java utilizada para construir este proyecto fue:

java 21.0.1 2023-10-17 LTS
Java(TM) SE Runtime Environment (build 21.0.1+12-LTS-29)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.1+12-LTS-29, mixed mode, sharing)

Esperamos que el programa sea de su agrado.