Object Oriented Design – 1

JabberPoint Refactoring Report

Author:

Mathew Shardin

Mathew.Shardin@student.nhlstenden.com

Student Code: 4951735

Drafted for:

Rob Smit Rob.smit@nhlstenden.com

3rd March 2023

Emmen, Netherlands

NHL Stenden of Applied Sciences

Table of changes

	Architectural Problem	Proposed Solution	Reasoning
1.	Naming conventions are not met	Rename variable and method names to meet naming criteria	Improve code readability, obey industry standards
2.	getCurrentSlide() method in Presentation.java accesses the currentSlideNumber field directly instead of using a getter	Use a getter method to access currentSlideNumber field	Improve security of the class
3.	setSlideNumber() method in Presentation.java doesn't check for out of bounds slide number values	Introduce a out of bounds slide number check before updating currentSlideNumber	Bug fix
4.	Names of xml tags of attributes in XMLAccessor.java are defined as static final fields, which might create duplicate code	Move names to an Enum class	Improve maintainability of code, allow for reusing of code
5.	Names of menu items in MenuController.java are defined as static final fields, which might create duplicate code and makes code less readable in general	Move names to an Enum class	Improve maintainability of code, allow for reusing of code, improve readability of code
6.	Some classes have unused empty constructors, which make the code less readable and are useless	Delete unused empty constructors	Improve code readability, declutter code
7.	Style class has a field "styles", which is an Array that contains instances of Style class itself	Move the Styles array, and a static getStyle() method into JabberPoint.java as styles of slides are application wide. All presentations use the same styles	Make architecture less confusing. Remove a circular dependency.
8.	SlideViewerComponent.java and Presentation.java have a circular dependency (SlideViewerComponent has Presentation and vice versa). There is also a confusing relationship with SlideViewerFrame.java	Create a clear relationship, where: SlideViewerComponent **has** Presentation. SlideViewerFrame **has** SlideViewerComponent. Create a method updateSlide() in SlideViewerFrame()	Create a clearer and more understandable relationship between elements. Lower the number of parameters that have to be passed to methods in mentioned classes.
9.	Standing from refactoring 8: KeyController.java is connected to Presentation.java, where as MenuController.java is connected to	Connect both controllers to SlideViewerFrame.java. Make their constructors take the same argument.	Make two controllers more unified and understandable. Lower the number of parameters that have to

	Presentation.java and SlideViewerFrame.java.	Allow for correct implementation of refactoring 8.	be passed to methods in mentioned classes.

Table of files changed

Change Number	Files Affected
1.	Accessor.java BitMapItem.java MenuController.java Presentation.java Slide.java
2.	Presentation.java
3.	Presentation.java
4.	XMLAccessor.java XMLTags.java
5.	MenuController.java MenuItemsNames.java
6.	BitMapItem.java SlideItem.java TextItem.java
7.	Style.java JabberPoint.java
8.	JabberPoint.java SlideViewerComponent.java SlideViewerFrame.java MenuController.java KeyController.java Presentation.java
9.	MenuController.java KeyController.java SlideViewerFrame.java