OOD Report

Arian Atapour

In the following table the issue, fix and which part of the software is fixed is explained.

|  |  |  |
| --- | --- | --- |
| **Issue** | **Fix** | **Part fixed** |
| 1 – Go to, can go over the limit of the presentation slide wise and will stay stuck on the slide selected previously. Recreate the error by inputting a bigger number than the slide counter limit. | If the number of the Go to input dialog box is over the total size of slides inside the presentation (getSize() method) then display the latest slide. This is done inside the MenuController class where the Go To menu item resides in. More accurately the fix is inside the actionPerfomed() method, inside the if statement.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality, reliability and usability. | Functionality  Reliability and Usability |
| 2 – By using Go to the slide counter goes over the limit. To recreate this error just input in the Go to input box a number bigger than the limit of the slides. You should get for example 6 out of 3 if you inputted 6 as a number. | If the number of the Go to input dialog box is over the total size of slides (getSize() method) then set the slide counter selected on the left to the total size since it displays the last slide. So, if we have a presentation with 3 slides and we input the number 6, the last slide is going to appear (as seen in fix 1) and the counter will display Slide 3 of 3. This is done inside the MenuController class where the Go To menu item resides in. More accurately the fix is inside the actionPerfomed() method, inside the if statement.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality, reliability and usability. | Functionality  Reliability and Usability |
| 3 – Too much text inside one method show() (About box text) | Breaking down the text into a separate method called text() and the utilization of StringBuilder.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and adaptability as well as testability. | Functionality  Adaptability  Testability |
| 4 – Unused variables (dead code) found in the Accessor class (DEMO\_NAME and DEFAULT\_EXTENSION never used). | Removed dead code which was never used.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 5 – Unused constructor (dead code) found in the BitmapItem class (BitmapItem constructor never used). | Removed dead code which was never used.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 6 – Unused method (dead code) found in the Slide class (getSlideItem() method never used). | Removed dead code which was never used.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 7 – Unused method (dead code) found in the SlideItem class (SlideItem constructor never used). | Removed dead code which was never used.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 8 – Unused constructor (dead code) found in the TextItem class (TextItem constructor never used). | Removed dead code which was never used.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 9 – Unused variables (dead code) found in the XMLAccessor class (DEFAULT\_API\_TO\_USE never used). | Removed dead code which was never used.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 10 – imageName variable from BitmapItem class is not final. | Made the imageName variable from BitMapItem final since no further value overriding is required.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
|  |  |  |
| 11 – presentation variable from KeyController class is not final. | Made the presentation variable from KeyController class final since no further value overriding is required.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 12 – Switch statement from keyPressed method taking too many lines of code from KeyController class | Made switch statement inline to save code lines.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 13 - Making parentF variable final from MenuController class. | Made the parent variable from MenuController class final since no further value overriding is required.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 14 - Making presentation variable final from MenuController class. | Made the presentation variable from MenuController class final since no further value overriding is required.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality. | Functionality |
| 15 - Using @Serial annotation for serialVersionUID to give compile-time assurance from MenuController class. | Using @Serial annotation for serialVersionUID to give compile-time assurance from MenuController class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. | Functionality  Supportability |
| 16 – 23 - Used lambda expression for ActionListener to enable functional programming and improve functionality from MenuController class. | Used lambda expression for ActionListener to enable functional programming and improve functionality from MenuController class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise. | Functionality  Supportability  Performance |
| 24 - Casting PAGENR to Object makes it redundant so remove it from MenuController class. | Casting PAGENR to Object makes it redundant so remove it.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise. | Functionality  Supportability  Performance |
| 25(Lib) - Removing unused import statements ActionListener and ActionEvent from MenuController class. | Removing unused import statements ActionListener and ActionEvent from MenuController class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise. | Functionality  Supportability  Performance |
| 25(Init) - slideViewComponent initialiser null is redundant, so I removed the null from Presentation class | slideViewComponent initialiser null is redundant, so I removed the null.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise. | Functionality  Supportability  Performance |
| 25(<>) - Slide argument from ArrayList can be replaced with <> because of less verbose from Presentation class. | Slide argument from ArrayList can be replaced with <> because of less verbose from Presentation class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise. | Functionality  Supportability  Performance |
| 26 - Casting showList.get() to Slide is redundant so remove the casting from Presentation class. | Casting showList.get() to Slide is redundant so remove the casting from Presentation class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 27 - To save verbose delete SlideItem argument from Vector so leave diamond <> from Slide class. | To save verbose delete SlideItem argument from Vector so leave diamond <> from Slide class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise. | Functionality  Supportability  Performance  Reliability and usability |
| 28 - Casting getSlideItems().elementAt(number) to SlideItem is redundant so delete the casting from Slide class. | Casting getSlideItems().elementAt(number) to SlideItem is redundant so delete the casting from Slide class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 29 - Level variable set to 0 is redundant since you initialize the variable already from SlideItem class. | Level variable set to 0 is redundant since you initialize the variable already from SlideItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 30 - Level variable is final so change it to final from SlideItem class. | Level variable is final so change it to final from SlideItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 31 - labelFont variable set to null is redundant from SlideViewerComponent class. | abelFont variable set to null is redundant from SlideViewerComponent class, so erase the null since is initialized.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 32 - labelFont variable is final so make it final from SlideViewerComponent class. | labelFont variable is final so make it final from SlideViewerComponent class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 33 - presentation variable set to null is redundant from SlideViewerComponent class. | presentation variable set to null is redundant from SlideViewerComponent class so make delete the null value since it is initiated.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 34 - frame variable set to null is redundant from SlideViewer component class. | frame variable set to null is redundant from SlideViewer component class so delete the null value since it is instantiated.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 35 - frame variable is final so make it final from SlideViewerComponent class. | frame variable is final so make it final from SlideViewerComponent class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 36 - Annotate @Serial serialVersionUID variable from SlideViewerComponent class. | Annotate @Serial serialVersionUID variable from SlideViewerComponent class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 37 - @Serial annotation for serialVersionUID variable from SlideViewerFrame. | @Serial annotation for serialVersionUID variable from SlideViewerFrame.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 38 - Variable text is final so make it so from TextItem class. | Variable text is final so make it so from TextItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 39 - DEAD CODE EMPTYTEXT variable is never used from TextItem class. | DEAD CODE EMPTYTEXT variable is never used from TextItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 40 - Replacing while loop with for loop for better readability and functionality from TextItem class. | Replacing while loop with for loop for better readability and functionality from TextItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 41 - Replacing while loop with for loop for better readability and functionality from TextItem class. | Replacing while loop with for loop for better readability and functionality from TextItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 42 - Removing unused library Iterator because of for loop change which integrates it inside from TextItem class. | Removing unused library Iterator because of for loop change which integrates it inside from TextItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 43 - To save verbose removing TextLayout argumanent and leaving diamond <> in TextItem class. | To save verbose removing TextLayout argumanent and leaving diamond <> in TextItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 44 - max initialiser to 0 is redundant from XMLAccessor class. | max initializer to 0 is redundant from XMLAccessor class so remove the 0 value since it is initialized.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 45 - maxItems initialiser to 0 is redundant from XMLAccessor class. | maxItems initializer to 0 is redundant from XMLAccessor class so remove the 0 value since it is initialized.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 46 - Unnecessary .toString() from XMLAccessor class. | Unnecessary .toString() from XMLAccessor class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 47 - Changing System.err.println to .printStackTrace for more error information from XMLAccessor class. | Changing System.err.println to .printStackTrace for more error information from XMLAccessor class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 48 - Collapsed ParserConfigurationException sax into SAXException for the similarity and lines of code saved from XMLAccessor class. | Collapsed ParserConfigurationException sax into SAXException for the similarity and lines of code saved from XMLAccessor class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| - | - | - |
| 50 - Casting slideItems.elementAt(itemNumber) to SlideItem is redundant so remove the casting from XMLAccessor class. | Casting slideItems.elementAt(itemNumber) to SlideItem is redundant so remove the casting from XMLAccessor class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 51 - Adding a private constructor to hide the implicit public one in the AboutBox class. | Adding a private constructor to hide the implicit public one in the About box class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| - | - | - |
| 53 - Converting the accessor class into an interface in the Accessor interface class. | Abstract classes without fields should be converted to an interface.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| - | - | - |
| - | - | - |
| 56 - Adding @Override annotation above the method signature in the KeyController class. | Adding @Override annotation above the method signature in the KeyController class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 57 - EMPTY DEFAULT so I moved the comment inside of it so it is not empty in KeyController class. | EMPTY DEFAULT so I moved the comment inside of it so it is not empty in KeyController class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 58 - Changing parent name to something else (parentF) since it is used a field with the same name in MenuController class. | Changing parent name to something else (parentF) since it is used a field with the same name in MenuController class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 59 - Making presentation variable transient in MenuController class. | Fields in a serializable class should either be transient or serializable.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 60-65 - Extracted the assignment out of the expression (filemMenu.add) from MenuController class. | Extracted the assignment out of the expression (filemMenu.add) from MenuController class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 66 - Improving performance by replacing the synchronized class with an unsynchronized one like ArrayList from Slide class. | Improving performance by replacing the synchronized class with an unsynchronized one like ArrayList from Slide class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| - | - | - |
| 68 - Removed Vector library since it is not needed anymore from Slide class. | Removed Vector library since it is not needed anymore from Slide class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 69 - Reordering modifiers in Slide class. | Reordering modifiers in Slide class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 70 – Changing the visibility of the constructor from public to protected in SlideItem class. | Changing the visibility of the constructor from public to protected in SlideItem class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 71 - Making slide variable transient in SideViewerComponent class. | Making slide variable transient in SideViewer component class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 72 - Making presentation variable transient in SideViewerComponent class. | Making presentation variable transient in SideViewerComponent class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 73 - Adding @Override annotation in SlideViewerComponent class. | Adding @Override annotation in SlideViewerComponent class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 74 - Adding @Override annotation in SlideViewerComponent class. | Adding @Override annotation in SlideViewerComponent class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 75 - Adding @Override annotation in SlideViewerFrame class. | Adding @Override annotation in SlideViewerFrame class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 76 - Reordering modifiers in SlideViewerFrame class. | Reordering modifiers in SlideViewerFrame class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 77 - Reordering modifiers in SlideViewerFrame class. | Reordering modifiers in SlideViewerFrame class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 78 - Extracting the method out of the expression in Style class. | Extracting the method out of the expression in Style class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 79 - Removed Vector library since we don't use it anymore from XMLAcessor class. | Removed Vector library since we don't use it anymore from XMLAcessor class.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 80 - To disable XXE attacks in XMLAccessor class. | To disable XXE attacks but still allow DOCTYPE declarations prohibited external enttities declarations, also had to rework the document part by adding DocumentBuilderFactory.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 81 - Don't allow negative numbers in Go to method in MenuController class. | Don't allow negative numbers and go back to first slide if a negative number is detected.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 82 - Data clump code smell fixed by implementing fields in another class (MenuController). | Data clump code smell fixed by implementing fields in another class (MenuController).  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 83 - Data clump code smell fixed by implementing fields in another class (SlideViewerComponent). | Data clump code smell fixed by implementing fields in another class (SlideViewerComponent).  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |
| 84 - Extracted methods from constructor into separate methods | Extracted methods from constructor into separate methods.  This fix is the most logical one because it does not affect the overall look and feel of the app and it also improves its functionality and supportability. As well as performance wise and reliability and usability wise. | Functionality  Supportability  Performance  Reliability and usability |