JabberPoint Refactoring

# AboutBox

Nothing out of ordinary. Small class, no need for changes

# Accessor

Changed to an interface due to the constructor not being used in any way. Removed “abstract” from the methods. Removed two static fields, because they weren’t used.

# BitmapItem

Removed second constructor. It wasn’t being used

public BitmapItem() {  
 this(0, null);  
}

# DemoPresentation

Unnecessary second object requirements in the constructor (due to the Accessor class extension) – *String unusedFilename*

# KeyController

Simplified switch case.

public void keyPressed(KeyEvent keyEvent) {  
 switch (keyEvent.getKeyCode()) {  
 case KeyEvent.VK\_PAGE\_DOWN, KeyEvent.VK\_DOWN, KeyEvent.VK\_ENTER, '+' -> presentation.nextSlide();  
 case KeyEvent.VK\_PAGE\_UP, KeyEvent.VK\_UP, '-' -> presentation.prevSlide();  
 case 'q', 'Q' -> System.*exit*(0);  
 *//Should not be reached* default -> {  
 }  
 }  
}

# MenuController

Constructor has a lot of code smell in it (bloaters). Made menuItem, fileMenu, viewMenu, helpMenu private variables outside constructor. Split bits of code responsible for control into smaller methods and put back to constructor.

Created separate class for all static fields.

Removed this variable because of no usage in any of the classes

private static final long serialVersionUID = 227L;

Function moveToSlide – added if statement to forbid from going to the slide number that doesn’t exist. Joption gets the current slide, then its parses it to the int and then I can prevent it from going over the ArrayList size.

# Presentation

Removed object type in ArrayList in method clear().

# Slide

public SlideItem getSlideItem(int number) {  
 return (SlideItem)items.elementAt(number);  
}

Made into this (IntelliJ signalized that it was redundant)

\/

public SlideItem getSlideItem(int number) {  
 return items.elementAt(number);  
}

Removed Vector object type from a constructor

public Slide() {  
 items = new Vector<SlideItem>();  
}

\/

public Slide() {  
 items = new Vector<>();  
}

getSideItem method can be removed since it’s not being used.

public SlideItem getSlideItem(int number) {  
 return items.elementAt(number);  
}

Doubled method name. Changed one to addSlideItem(). Changed few depending on it methods.

public void addSlideItem(SlideItem anItem) {  
 items.addElement(anItem);  
}

# SlideItem

Removed second constructor since it was not being used.

public SlideItem() {  
 this(0);  
}

# SlideViewerComponent

Removed, because of not actual use in any of the classes.

private static final long serialVersionUID = 227L;

# Style

Variables didn’t have the access modifiers. Added them just in case (yes color is very important component and needs to be protected) for limited access.

# TextItem

Removed second constructor because it was not being used.

public TextItem() {  
 this(0, EMPTYTEXT);  
}

Removed variable. It wasn’t being used

private static final String EMPTYTEXT = "No Text Given";

# Additions

Created *MenuControlStatic* and *SlideViewerStatic* classes for data clumps with static variables and separated them making the code cleaner.

RESIT CODE CHANGES

Code placed below is an extension to the changes above and should be treated as one complete body.

# XML Accessor

Has been split to XMLAccessorSave and XMLAccessorLoad. Those changes also required to change few naming inconsistencies/dependencies in other classes like:

public void actionPerformed(ActionEvent e) {  
 AccessorSave xmlAccessorSave = new XMLAccessorSave();  
 try {  
 xmlAccessorSave.saveFile(presentation, MenuControlStatic.SAVEFILE);

public void actionPerformed(ActionEvent actionEvent) {  
 presentation.clear();  
 AccessorLoad xmlAccessorLoad = new XMLAccessorLoad();  
 try {  
 xmlAccessorLoad.loadFile(presentation, MenuControlStatic.TESTFILE);

Vector collection remains unchanged since the speed of loading it won’t be that impacted.

Changing this class also lead to splitting interface:

\/

# Accessor

This interface has been split to AccessorSave and AccessorLoad to suit the needs of the implementation.

Started changing the Presentation and PresentationControl classes.