

# CS 471: Software Engineering

Fall 2021

## Homework 4 – Implementation of an Advanced Maintenance Task in an Unfamiliar Open-Source Software (jEdit)

Software system and version: jEdit 5.5.0

### Change Request:

Add a new feature to jEdit which allows lines to be drawn beneath the text to simulate the look of a paper notepad. Also, there should be an option to allow the user to disable/enable this feature. The location of this control should be under “Utilities->Global Options...->jEdit->Text Area”.

### Detailed Development Environment Description:

MacOS Big Sur Version 11.5

Java jdk1.8.0\_202.

Apache Ant(TM) version 1.10.11 compiled on July 10 2021

Visual Studio Code Version: 1.60.2 (Universal)

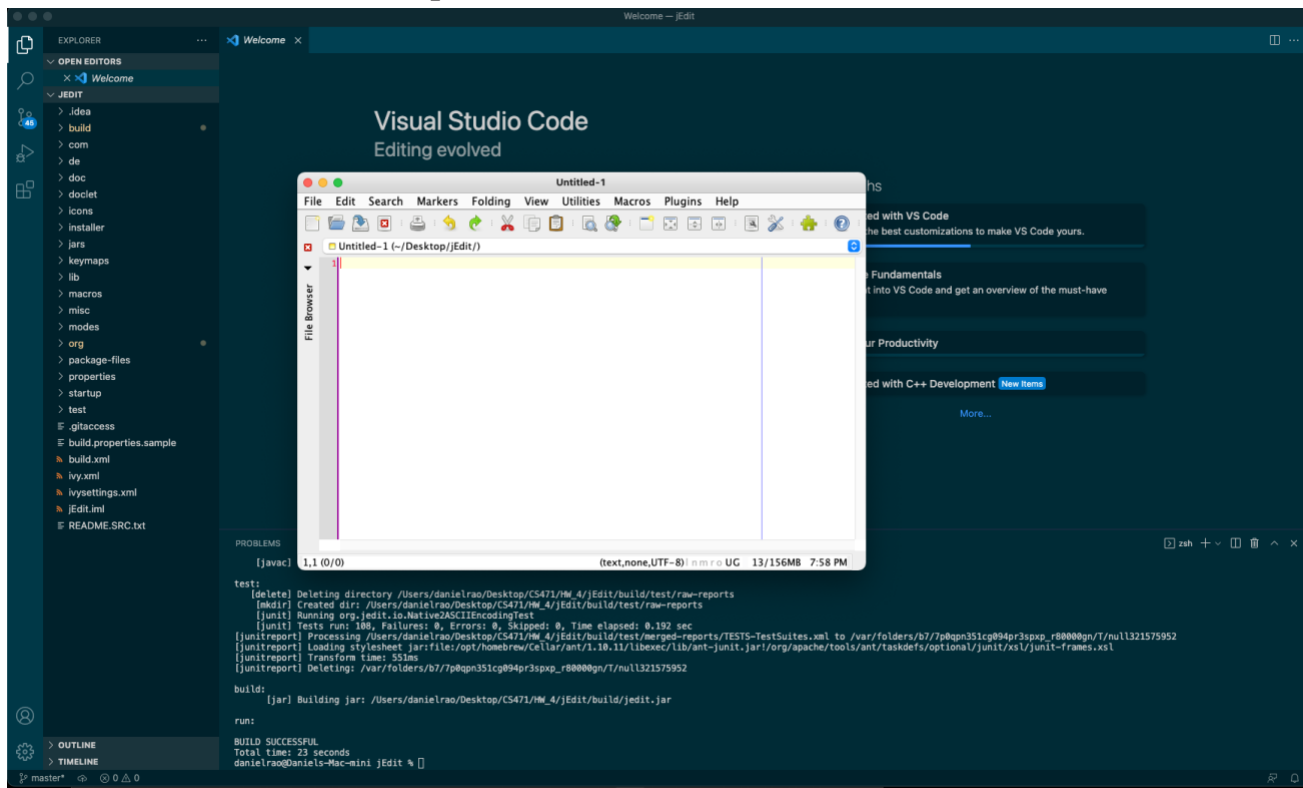
git version 2.30.1 (Apple Git-130)

Terminal

### Description of Problems Encountered during Compiling/Building and Running:

Main problem that I encountered was switching from Java version 11 to Java version 8. Other than that once imported into the visual studio code everything seemed to work just fine.

### Screenshot of successful compilation:



```

| default | 1 | 0 | 0 | 0 | 1 | 0 |
| master  | 0 | 0 | 0 | 0 | 0 | 0 |
| compile | 1 | 0 | 0 | 0 | 1 | 0 |
| runtime | 1 | 0 | 0 | 0 | 1 | 0 |
| test    | 5 | 0 | 0 | 1 | 4 | 0 |
| provided | 0 | 0 | 0 | 0 | 0 | 0 |
| sources | 3 | 0 | 0 | 0 | 3 | 0 |
| javadoc | 3 | 0 | 0 | 0 | 3 | 0 |
| ant-contrib | 2 | 0 | 0 | 0 | 2 | 0 |
| docbook | 5 | 0 | 0 | 0 | 5 | 0 |
| fop     | 26 | 0 | 0 | 1 | 25 | 0 |
| launch4j | 13 | 0 | 0 | 1 | 16 | 0 |
| scripting | 5 | 0 | 0 | 0 | 5 | 0 |
| default-plugins | 2 | 0 | 0 | 0 | 2 | 0 |

[ivy:retrieve]
[ivy:retrieve] :: problems summary ::
[ivy:retrieve] :::: WARNINGS
[ivy:retrieve] circular dependency found: org.apache.xmlgraphics#batik-svg-dom;1.7->org.apache.xmlgraphics#batik-anim;1.7->...
[ivy:retrieve] circular dependency found: org.apache.xmlgraphics#batik-anim;1.7->org.apache.xmlgraphics#batik-svg-dom;1.7->...
[ivy:retrieve] circular dependency found: org.apache.xmlgraphics#batik-bridge;1.7->org.apache.xmlgraphics#batik-gvt;1.7->...
[ivy:retrieve] circular dependency found: org.apache.xmlgraphics#batik-gvt;1.7->org.apache.xmlgraphics#batik-bridge;1.7->...
[ivy:retrieve] circular dependency found: org.apache.xmlgraphics#batik-bridge;1.7->org.apache.xmlgraphics#batik-script;1.7->...
[ivy:retrieve] circular dependency found: org.apache.xmlgraphics#batik-bridge;1.7->org.apache.xmlgraphics#batik-script;1.7->...
[ivy:retrieve] circular dependency found: org.apache.xmlgraphics#batik-script;1.7->org.apache.xmlgraphics#batik-bridge;1.7->...
[ivy:retrieve]
[ivy:retrieve] :: USE VERBOSE OR DEBUG MESSAGE LEVEL FOR MORE DETAILS
[ivy:retrieve] :: retrieving :: org.jedit#jedit [sync]
[ivy:retrieve] confs: [default, master, compile, runtime, test, provided, sources, javadoc, ant-contrib, docbook, fop, launch4j, scripting, default-plugins]
[ivy:retrieve] 0 artifacts copied, 68 already retrieved (0KB/55ms)
[ivy:retrieve] :: resolving dependencies :: org.apache.ivy#ivy-caller;working
[ivy:retrieve] confs: [default]
[ivy:retrieve] found org.apache.ivy#ivy;2.2.0 in public
[ivy:retrieve] :: resolution report :: resolve 42ms :: artifacts dl 2ms

| conf | modules | artifacts | | | | | |
|---|---|---|---|---|---|---|---|
| | number | search | dwnlded | evicted | | number | dwnlded |
|-----|-----|-----|-----|-----|
| default | 1 | 0 | 0 | 0 | 1 | 0 |

[ivy:retrieve] :: retrieving :: org.apache.ivy#ivy-caller
[ivy:retrieve] confs: [default]
[ivy:retrieve] 1 artifacts copied, 0 already retrieved (925kB/8ms)

setup:
compile:
compile-test:
test:
[delete] Deleting directory /Users/danielrao/Desktop/jEdit/build/test/raw-reports
[mkdir] Created dir: /Users/danielrao/Desktop/jEdit/build/test/raw-reports
[junit] Running org.jedit.io.Native2ASCIIEncodingTest
[junit] Tests run: 100, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.177 sec
[junitreport] Processing /Users/danielrao/Desktop/jEdit/build/test/merged-reports/TESTS-TestSuites.xml to /var/folders/b7/7p0qpn351cg094pr3sxp_r80000gn/T/null1089843744
[junitreport] Loading stylesheet jar:file:/opt/homebrew/Cellar/ant/1.10.11/libexec/lib/ant-junit.jar!/org/apache/tools/ant/taskdefs/optional/junit/xsl/junit-frames.xsl
[junitreport] Transform time: 614ms
[junitreport] Deleting: /var/folders/b7/7p0qpn351cg094pr3sxp_r80000gn/T/null1089843744

build:
[jar] Building jar: /Users/danielrao/Desktop/jEdit/build/jedit.jar

run:
BUILD SUCCESSFUL
Total time: 9 seconds
danielrao@Daniels-Mac-mini jEdit %
```

Identifying the Relevant Code (i.e., Concept Location):

This feature was much harder to implement than muCommander. I was able to look up several keywords such as `TextArea`. Visual Studio Code helped me a lot to find it using the shortcut on Mac `cmd + shift + F`. This helps me to look up any keyword throughout its directories and sub-directories.

Once I was able to locate where the `text-area` was then I just had to look how they were getting initialized.

Once that was complete, I was able to successfully implement the GUI checkbox. As I was looking through sub-folders of several files I happened to stumble upon `TextAreaOptionPane` this also further gave me clarity to what parts of the code would I have to tackle in order to make a successful change. As I was thinking about the problem I was also able to figure out that I would be needing some kind of toggle to check or uncheck the checkbox and check its value accordingly as the program went into the launch phase.

Successful Implementation Video:

<https://youtu.be/-GRerIvw76w>

Git Diff of Source Code of the Implementation in a Side-by-Side readable format:

## A) EditPane.java

```

EditPane.java (Working Tree) 94, M, X
org > git > sp > jedit > EditPane.java > Language Support for Java(TM) by Red Hat > EditPane > initPainter(TextAreaPainter)
1035 painter.setLineHighlightColor(
1036     jEdit.getColorProperty("view.LineHighlightColor"));
1037 painter.setAntiAlias(new AntiAlias(jEdit.getProperty("view.antiAlias")));
1038 painter.setFractionalFontMetricsEnabled(jEdit.getBooleanProperty(
1039     "view.fracFontMetrics"));
1040
1041 painter.setSelectionFgColor(jEdit.getColorProperty(
1042     "view.selectionFgColor"));
1043 painter.setSelectionFgColorEnabled(jEdit.getBooleanProperty(
1044     "view.selectionFg"));
1045
1046 String defaultFont = jEdit.getProperty("view.font");
1047 int defaultFontSize = jEdit.getIntegerProperty("view.fontSize",12);
1048 painter.setStyle(SyntaxUtilities.loadStyles(defaultFont,defaultFontSize));
1035 painter.setLineHighlightColor(
1036     jEdit.getColorProperty("view.LineHighlightColor"));
1037 painter.setAntiAlias(new AntiAlias(jEdit.getProperty("view.antiAlias")));
1038 painter.setFractionalFontMetricsEnabled(jEdit.getBooleanProperty(
1039     "view.fracFontMetrics"));
1040+ painter.setNotebookLinesEnabled(jEdit.getBooleanProperty(
1041+     "view.notebookLines"));
1042 painter.setSelectionFgColor(jEdit.getColorProperty(
1043     "view.selectionFgColor"));
1044 painter.setSelectionFgColorEnabled(jEdit.getBooleanProperty(
1045     "view.selectionFg"));
1046
1047 String defaultFont = jEdit.getProperty("view.font");
1048 int defaultFontSize = jEdit.getIntegerProperty("view.fontSize",12);
1049 painter.setStyle(SyntaxUtilities.loadStyles(defaultFont,defaultFontSize));

```

## B) StandAloneTextArea.java

```

StandAloneTextArea.java (Working Tree) 94, M, X
org > git > sp > jedit > textarea > StandAloneTextArea.java > Language Support for Java(TM) by Red Hat > StandAloneTextArea > initPainter()
291 view.blockCaret();
292 painter.setThickCaretEnabled(getBooleanProperty(
293     "view.thickCaret"));
294 painter.setLineHighlightEnabled(getBooleanProperty(
295     "view.lineHighlight"));
296 painter.setLineHighlightColor(
297     getColorProperty("view.LineHighlightColor"));
298 painter.setAntiAlias(new AntiAlias(getProperty("view.antiAlias")));
299 painter.setFractionalFontMetricsEnabled(getBooleanProperty(
300     "view.fracFontMetrics"));
301 painter.setSelectionFgColor(getColorProperty(
302     "view.selectionFgColor"));
303 painter.setSelectionFgColorEnabled(getBooleanProperty(
304     "view.selectionFg"));
305+ painter.setNotebookLinesEnabled(getBooleanProperty(
306+     "notebookLines"));
307 String defaultFont = getProperty("view.font");
308 int defaultFontSize = getIntegerProperty("view.fontSize",12);
309 painter.setStyle(SyntaxUtilities.loadStyles(defaultFont,defaultFontSize));
310
311 SyntaxStyle[] foldLineStyle = new SyntaxStyle[4];
312 for(int i = 0; i <= 3; i++)
313 {
314     foldLineStyle[i] = SyntaxUtilities.parseStyle(
315         getProperty("view.style.foldLine." + i),
316         defaultFont,defaultFontSize,true);
291 view.blockCaret();
292 painter.setThickCaretEnabled(getBooleanProperty(
293     "view.thickCaret"));
294 painter.setLineHighlightEnabled(getBooleanProperty(
295     "view.lineHighlight"));
296 painter.setLineHighlightColor(
297     getColorProperty("view.LineHighlightColor"));
298 painter.setAntiAlias(new AntiAlias(getProperty("view.antiAlias")));
299 painter.setFractionalFontMetricsEnabled(getBooleanProperty(
300     "view.fracFontMetrics"));
301 painter.setSelectionFgColor(getColorProperty(
302     "view.selectionFgColor"));
303 painter.setSelectionFgColorEnabled(getBooleanProperty(
304     "view.selectionFg"));
305+ painter.setNotebookLinesEnabled(getBooleanProperty(
306+     "notebookLines"));
307 String defaultFont = getProperty("view.font");
308 int defaultFontSize = getIntegerProperty("view.fontSize",12);
309 painter.setStyle(SyntaxUtilities.loadStyles(defaultFont,defaultFontSize));
310
311 SyntaxStyle[] foldLineStyle = new SyntaxStyle[4];
312 for(int i = 0; i <= 3; i++)
313 {
314     foldLineStyle[i] = SyntaxUtilities.parseStyle(
315         getProperty("view.style.foldLine." + i),
316         defaultFont,defaultFontSize,true);

```

## C) Jedit\_en.props

```

jedit_en.props (Working Tree) M, X
org > jedit > localization > jedit_en.props
1040 options.textarea.insertCompletionWithDigit=Complete word by typing corresponding digit (ot
1041 options.textarea.lineSpacing.label=Extra vertical line spacing (in pixels):
1042 #)}}
1043
1044 #((( Tool Bar pane
1045 options.toolbar.label=Tool Bar
1046 options.toolbar.showToolbars>Show tool bar
1047 options.toolbar.caption=Buttons to display on tool bar:
1048 options.toolbar.add=Add...
1049 options.toolbar.edit=Edit...
1050 options.toolbar.remove=Remove
1051 options.toolbar.moveUp=Move Up
1052 options.toolbar.moveDown=Move Down
1053 options.toolbar.reset=Reset to Default
1054 options.toolbar.edit.title=Edit Tool Bar Entry
1055 options.toolbar.edit.caption=Type:
1040 options.textarea.insertCompletionWithDigit=Complete word by typing corresponding digit (ot
1041 options.textarea.lineSpacing.label=Extra vertical line spacing (in pixels):
1042+ options.textarea.notebookLines=Enable notebook-style lines
1043 #)}}
1044
1045 #((( Tool Bar pane
1046 options.toolbar.label=Tool Bar
1047 options.toolbar.showToolbars>Show tool bar
1048 options.toolbar.caption=Buttons to display on tool bar:
1049 options.toolbar.add=Add...
1050 options.toolbar.edit=Edit...
1051 options.toolbar.remove=Remove
1052 options.toolbar.moveUp=Move Up
1053 options.toolbar.moveDown=Move Down
1054 options.toolbar.reset=Reset to Default
1055 options.toolbar.edit.title=Edit Tool Bar Entry
1056 options.toolbar.edit.caption=Type:

```

## D) TextAreaOptionPane.java

```

TextAreaOptionPane.java (Working Tree) 94, M, X
org > git > sp > jedit > options > TextAreaOptionPane.java > Language Support for Java(TM) by Red Hat > TextAreaOptionPane > _init()
236
237 /* Strip trailing EOL */
238 stripTrailingEOL = new JCheckBox(jEdit.getProperty(
239     "options.textarea.stripTrailingEOL"));
240 stripTrailingEOL.setSelected(jEdit.getBooleanProperty("stripTrailingEOL"));
241 addComponent(stripTrailingEOL);
242
243 completeFromAllBuffers = new JCheckBox(jEdit.getProperty(
244     "options.textarea.completeFromAllBuffers"));
245 completeFromAllBuffers.setSelected(jEdit.getBooleanProperty("completeFromAllBuffers"));
246 addComponent(completeFromAllBuffers);
247
248 insertCompletionWithDigit = new JCheckBox(jEdit.getProperty(
249     "options.textarea.insertCompletionWithDigit"));
250 insertCompletionWithDigit.setSelected(jEdit.getBooleanProperty("insertCompletionWithDigit"));
251 addComponent(insertCompletionWithDigit);
252
253+
254+ notebookLines = new JCheckBox(jEdit.getProperty(
255+     "options.textarea.notebookLines"));
256+ notebookLines.setSelected(jEdit.getBooleanProperty("view.notebookLines"));
257+ addComponent(notebookLines);
258
259 }
260
261 //((( _save() method
262 public void _save()
263 {
264     jEdit.setFontProperty("view.font",font.getFont());
265     jEdit.setBooleanProperty("view.enableFontSubst",fontSubst.isSelected());
266     fontSubstList.save();
267     jEdit.setBooleanProperty("view.enableFontSubstSystemFonts",
268         fontSubstSystemFonts.isSelected());
269
236
237 /* Strip trailing EOL */
238 stripTrailingEOL = new JCheckBox(jEdit.getProperty(
239     "options.textarea.stripTrailingEOL"));
240 stripTrailingEOL.setSelected(jEdit.getBooleanProperty("stripTrailingEOL"));
241 addComponent(stripTrailingEOL);
242
243 completeFromAllBuffers = new JCheckBox(jEdit.getProperty(
244     "options.textarea.completeFromAllBuffers"));
245 completeFromAllBuffers.setSelected(jEdit.getBooleanProperty("completeFromAllBuffers"));
246 addComponent(completeFromAllBuffers);
247
248 insertCompletionWithDigit = new JCheckBox(jEdit.getProperty(
249     "options.textarea.insertCompletionWithDigit"));
250 insertCompletionWithDigit.setSelected(jEdit.getBooleanProperty("insertCompletionWithDigit"));
251 addComponent(insertCompletionWithDigit);
252
253+
254+ notebookLines = new JCheckBox(jEdit.getProperty(
255+     "options.textarea.notebookLines"));
256+ notebookLines.setSelected(jEdit.getBooleanProperty("view.notebookLines"));
257+ addComponent(notebookLines);
258
259 }
260
261 //((( _save() method
262 public void _save()
263 {
264     jEdit.setFontProperty("view.font",font.getFont());
265     jEdit.setBooleanProperty("view.enableFontSubst",fontSubst.isSelected());
266     fontSubstList.save();
267     jEdit.setBooleanProperty("view.enableFontSubstSystemFonts",
268         fontSubstSystemFonts.isSelected());
269

```

ii)

```

org > git > sp > jedit > options > TextAreaOptionPane.java > Language Support for Java(TM) by Red Hat > TextAreaOptionPane > _init()
293 | | | | | JEdit.setColorProperty("view.wrapGuideColor",
294 | | | | | wrapGuideColor.getSelectedColor());
295 | | | | | JEdit.setBooleanProperty("view.pageBreaks", pageBreaks.isSelected());
296 | | | | | JEdit.setColorProperty("view.pageBreaksColor", pageBreaksColor.getSelectedColor());
297 | | | | |
298 | | | | | JEdit.setIntegerProperty("view.electricBorders",electricBorders
299 | | | | | .isSelected() ? 3 : 0);
300 | | | | | AntiAlias nv = new AntiAlias(JEdit.getProperty("view.antiAlias"));
301 | | | | | nv.set(antiAlias.getSelectedIndex());
302 | | | | | JEdit.setProperty("view.antiAlias", nv.toString());
303 | | | | | JEdit.setBooleanProperty("view.fracFontMetrics",fracFontMetrics.isSelected());
304 | | | | | JEdit.setBooleanProperty("stripTrailingEOL", stripTrailingEOL.isSelected());
305 | | | | | JEdit.setBooleanProperty("completeFromAllBuffers", completeFromAllBuffers.isSelected());
306 | | | | | JEdit.setBooleanProperty("insertCompletionWithDigit", insertCompletionWithDigit.isSelected());
307 | | | | |
308 | | | | | JEdit.setIntegerProperty("options.textarea.lineSpacing",
309 | | | | | Integer.valueOf(lineSpacing.getText()));
310 | | | | |
311 | | | | | }
312 | | | | |
313 | | | | | }
314 | | | | |
315 | | | | |
316 | | | | |
317 | | | | |
318 | | | | |
319 | | | | |
320 | | | | |
321 | | | | |

```

iii)

```

org > git > sp > jedit > options > TextAreaOptionPane.java > Language Support for Java(TM) by Red Hat > TextAreaOptionPane > _init()
331 | private ColorWellButton eoMarkerColor;
332 | private JCheckBox wrapGuide;
333 | private ColorWellButton wrapGuideColor;
334 | private JCheckBox pageBreaks;
335 | private ColorWellButton pageBreaksColor;
336 | private JCheckBox electricBorders;
337 | private JComboBox<String> antiAlias;
338 | private JCheckBox fracFontMetrics;
339 | private JCheckBox stripTrailingEOL;
340 | private JCheckBox completeFromAllBuffers;
341 | private JCheckBox insertCompletionWithDigit;
342 | private JTextField lineSpacing;
343 |
344 |
345 |
346 |
347 |
348 |
349 |
350 |
351 |
352 |
353 |
354 |
355 |
356 |
357 |
358 |
359 |
360 |
361 |
362 |
363 |
364 |
365 |
366 |
367 |
368 |
369 |
370 |

```

## Time Required for Completing this Assignment:

Activity	Approximate Time Required (in minutes)
How much time did it take you to figure out how to compile and run the software (i.e., total amount of time it took to read documentation, download the software, install Ant, etc.), and NOT how much time it took the builder to build the software.	40 mins
How much time did it take you to understand the code that you need to modify, to implement the change request, and to test it?	740 mins