

GUI-Entwicklung

Editor-Anwendung unter Eclipse/SWT

Team:

Marius Schenzle

Joel Wielath

Joey Gopp

25.10.2019

Inhaltsverzeichnis

1 Entwurf der Benutzeroberfläche	1
1.1 Aufbau der Präsentationsschicht	1
1.2 Beschreibung der Workflows der Editor-Anwendung	2
1.3 Darstellung der Klassenstruktur	5
2 Implementierung unter Eclipse/SWT	8
3 Internationalisierung der Anwendung	9

1 Entwurf der Benutzeroberfläche

1.1 Aufbau der Präsentationsschicht

Die fertige Applikation ist in Abbildung 1 zu sehen.

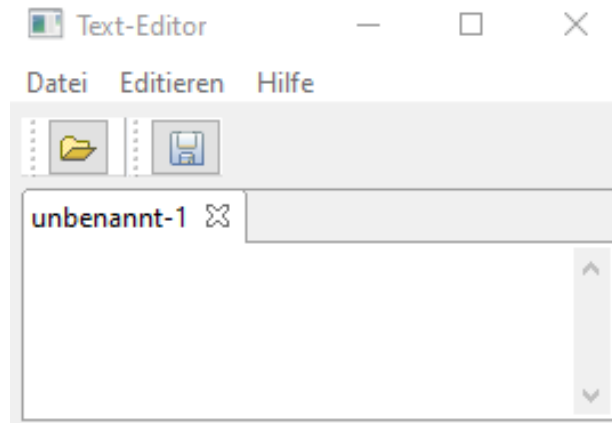


Abbildung 1: Editor Applikation

Das Menu ist folgendermaßen aufgebaut

1. Datei-Menu

- (a) Neu
- (b) Öffnen...
- (c) Speichern...
- (d) Beenden

2. Editieren

- (a) Text-Farbe

3. Hilfe

- (a) Version

Unter der Menu-Leiste befindet sich eine sogenannte 'Coolbar' (siehe Abbildung 1). Über diese können die Kommandos 'Datei öffnen' und 'Datei speichern' ausgeführt werden. Text wird über die Textfelder in den Tabs editiert. Über den Farb-Dialog kann pro Tab die Textfarbe ausgewählt werden. Der Farb-Dialog ist in Abbildung 2 zu sehen.

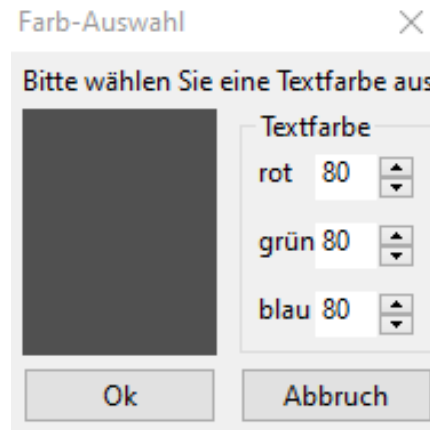


Abbildung 2: Auswahl der Textfarbe im aktuellen Tab

1.2 Beschreibung der Workflows der Editor-Anwendung

Zur Veranschaulichung der Editor-Workflows werden Aktivitätsdiagramme nach der UML2-Spezifikation verwendet. Die folgenden beiden Diagramme veranschaulichen den Workflow für das Öffnen und Schließen eines Tabs.

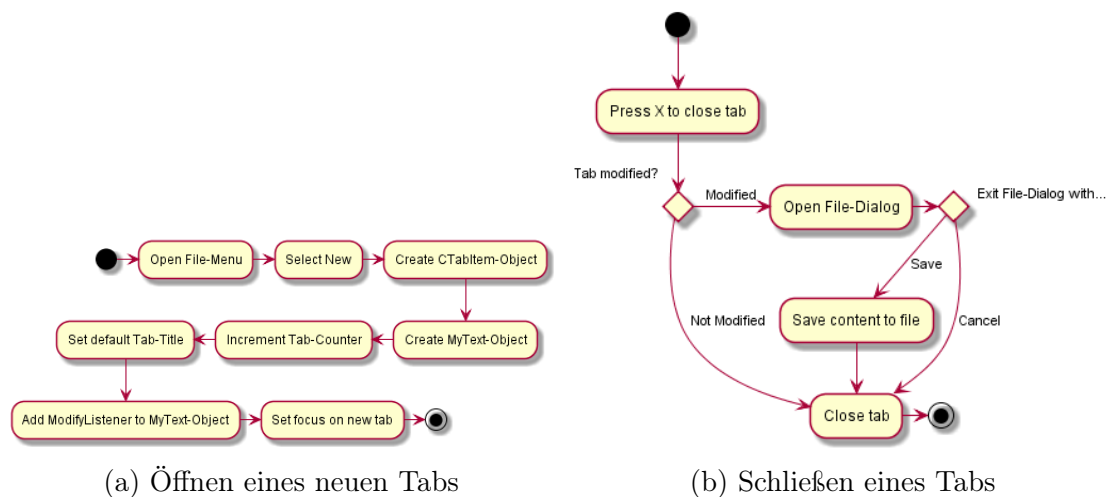


Abbildung 3: Tabs öffnen und schließen

Weiterhin werden in unserer Editor-Applikation verschiedene Dialogfenster benötigt. Ein Dialog zur Anzeige der Versionsinformation sowie einen zur Auswahl der Schriftfarbe im aktuell geöffneten Tab.

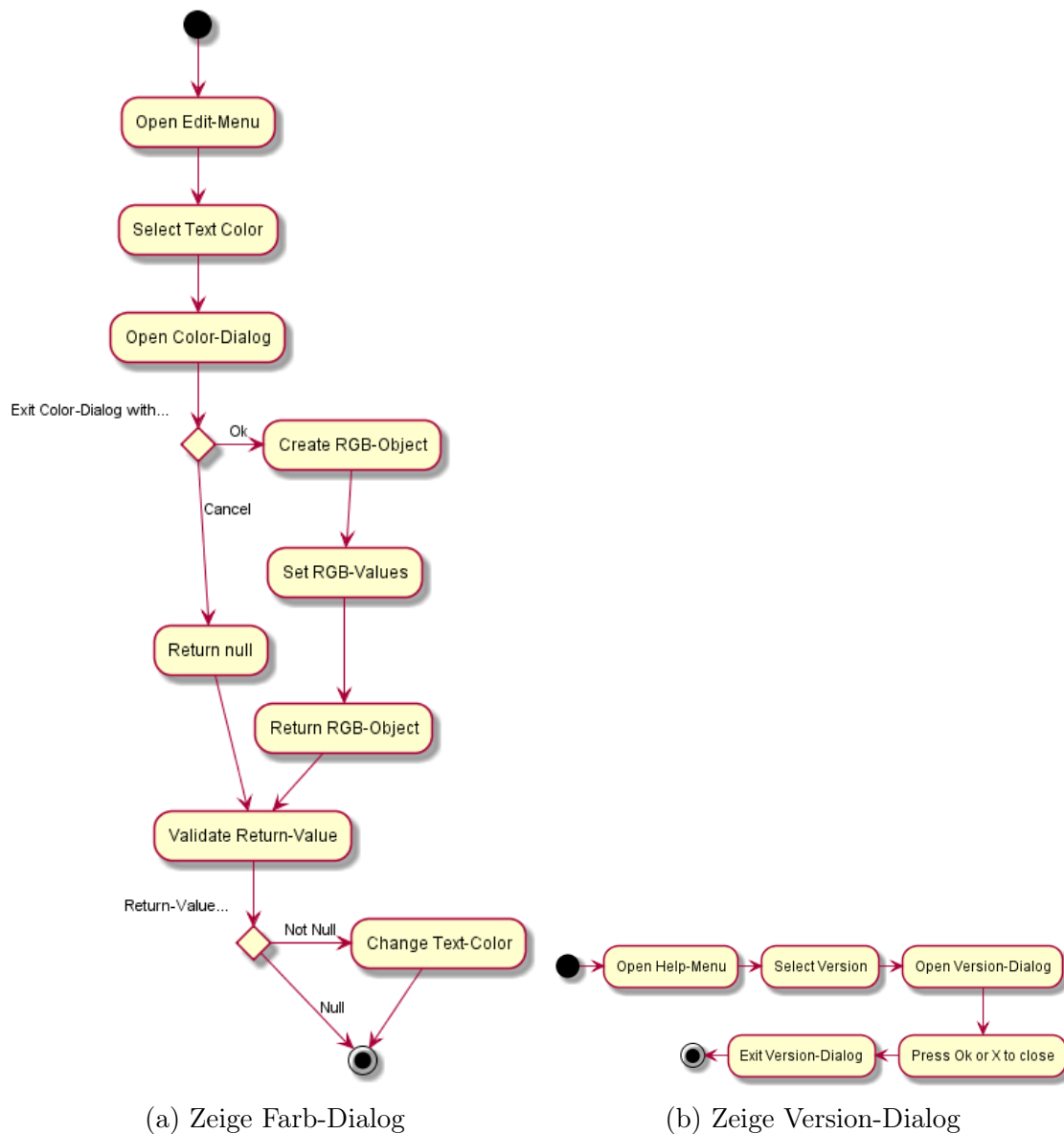
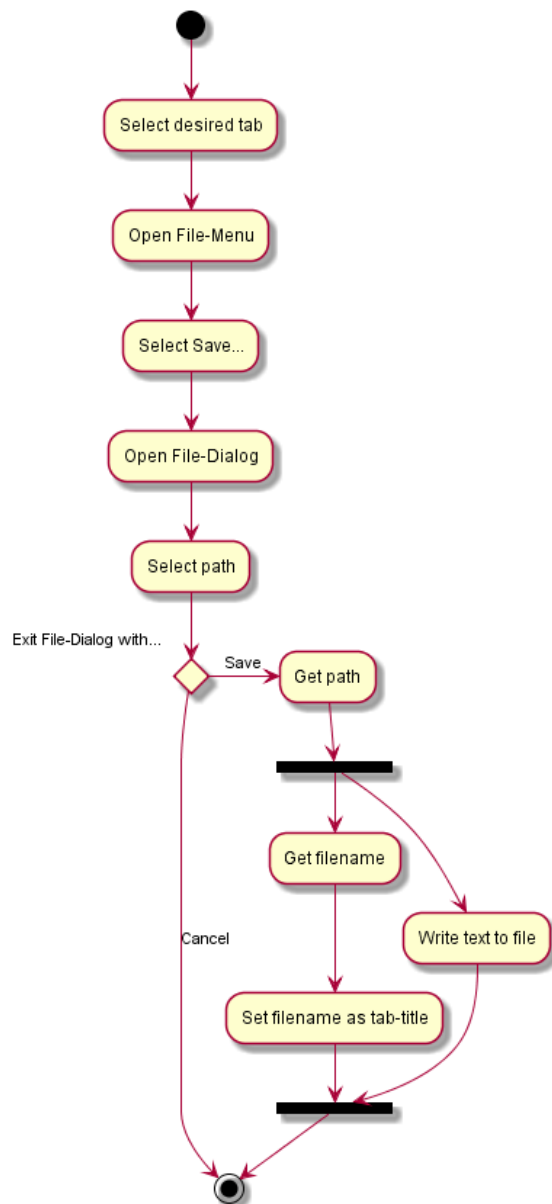
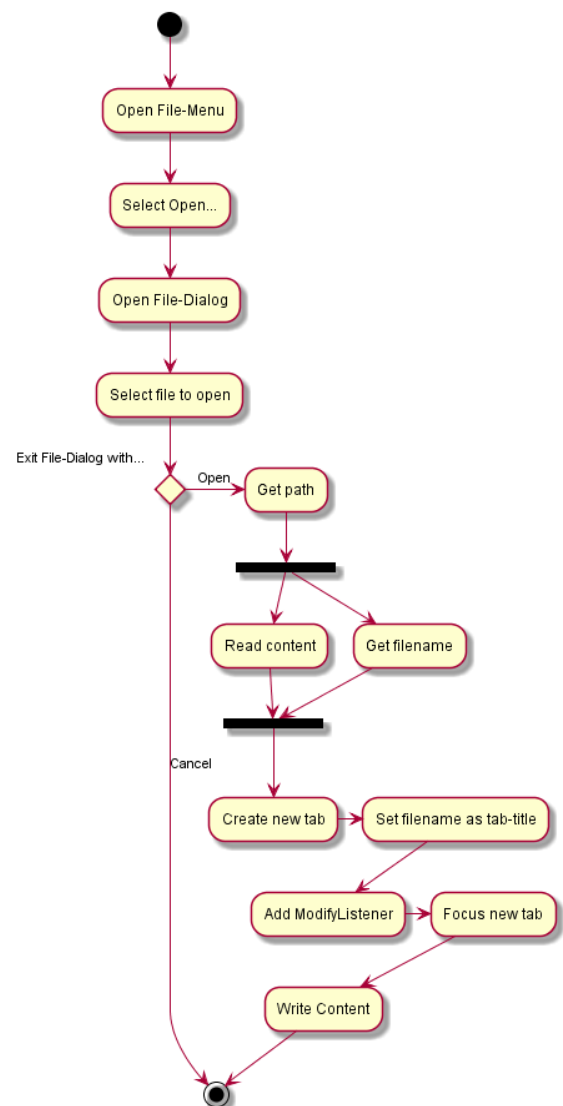


Abbildung 4: Farb- und Versions-Dialog

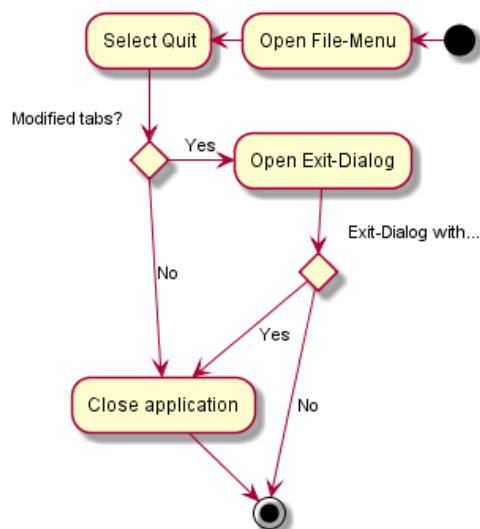
Über das Datei-Menü können Funktionen wie öffnen einer neuen Datei, speichern des geöffneten Tabs und weiteres ausgeführt werden. Die möglichen Aktionen sind in der Abbildung 5 zu sehen.



(a) Datei öffnen



(b) Text in Datei speichern



(c) Anwendung beenden

Abbildung 5: Aktionen im Datei-Menü

1.3 Darstellung der Klassenstruktur

Die Applikation soll in drei Schichten aufgeteilt werden: Präsentations-, Dialogmanagement- und Applikationsschicht. In Abbildung 6, 7 und 8 sind die drei Schichten im UML2-Klassendiagramm dargestellt.

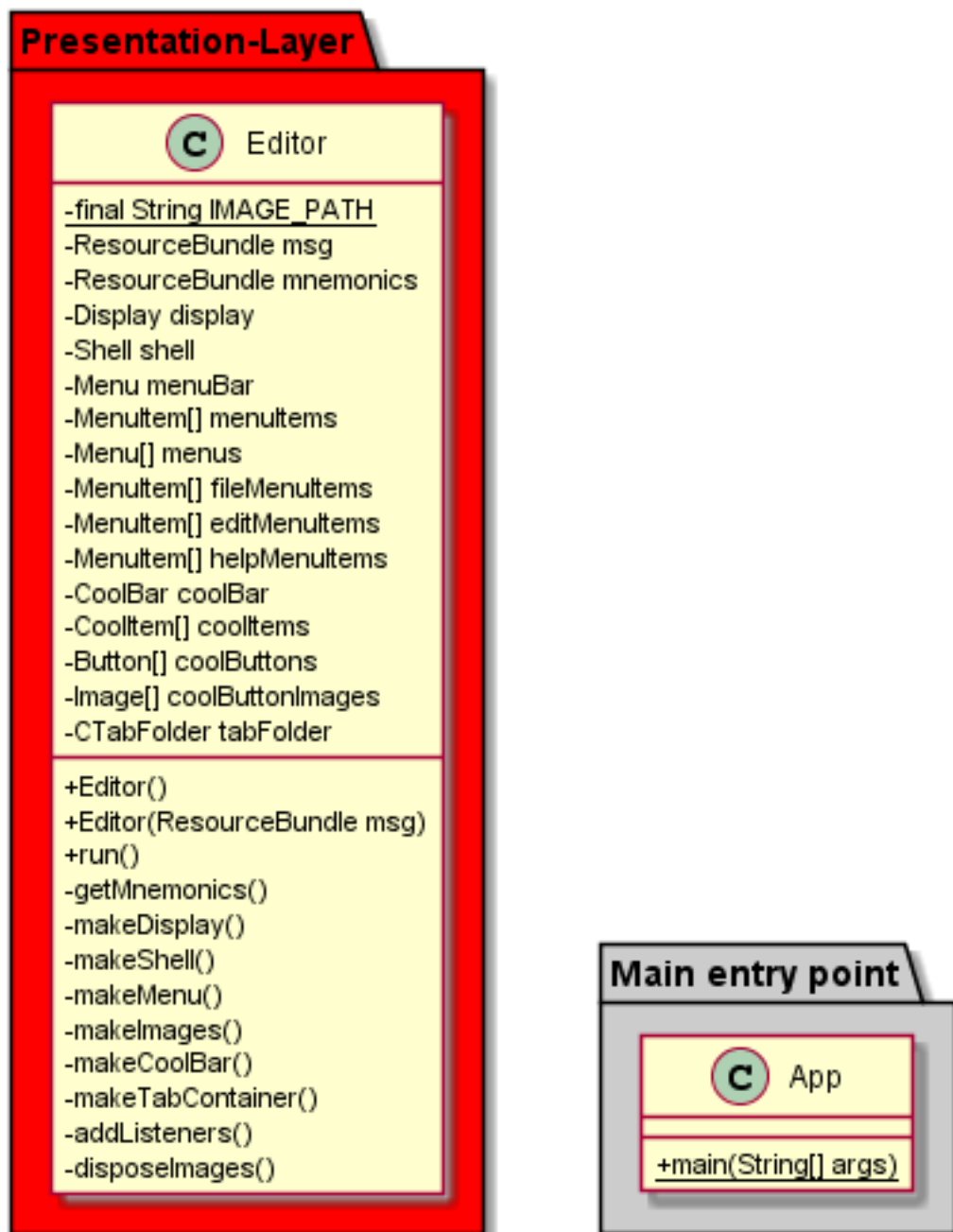


Abbildung 6: Präsentationsschicht

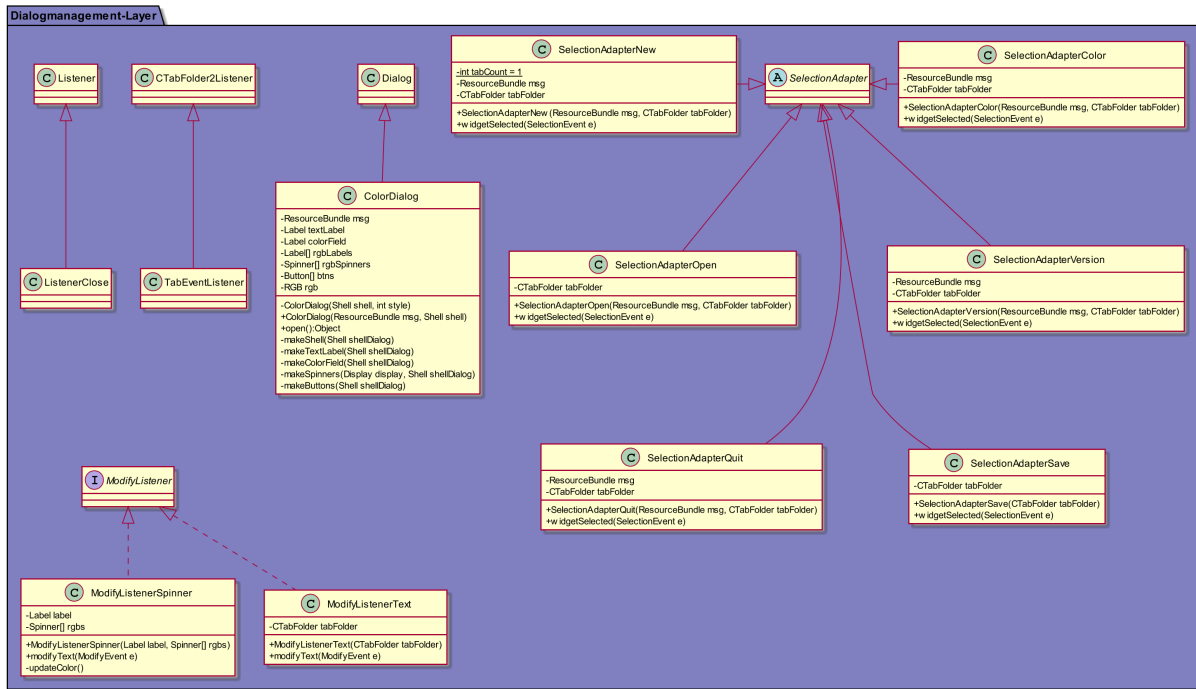


Abbildung 7: Dialogmanagementschicht

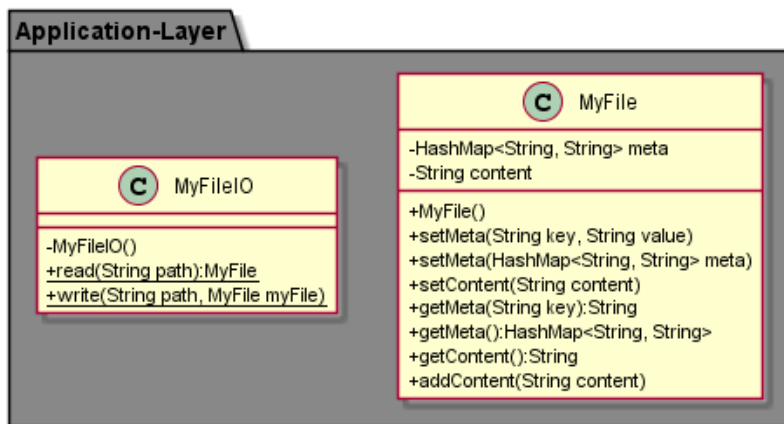


Abbildung 8: Applikationsschicht

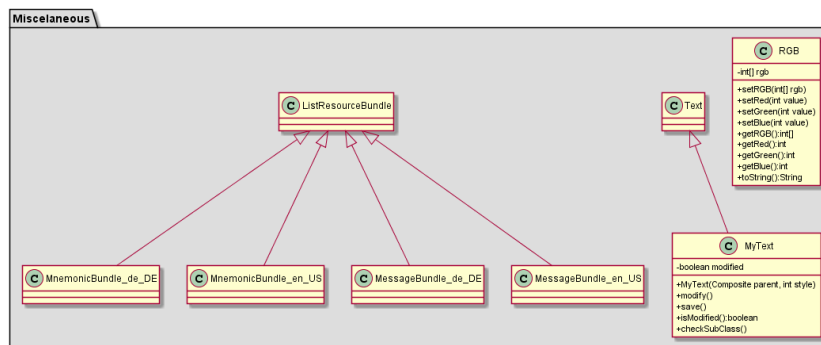


Abbildung 9: Weitere erstellte Klassen

Da das Klassendiagramm bereits relativ viele und große Klassen enthält, haben wir uns dagegen entschieden alle Assoziationen zwischen den Klassen darzustellen (lediglich die Vererbungshierarchien sind dargestellt). Abbildung 10 zeigt einen Teil der Gesamtapplikation mit entsprechenden Assoziationen.

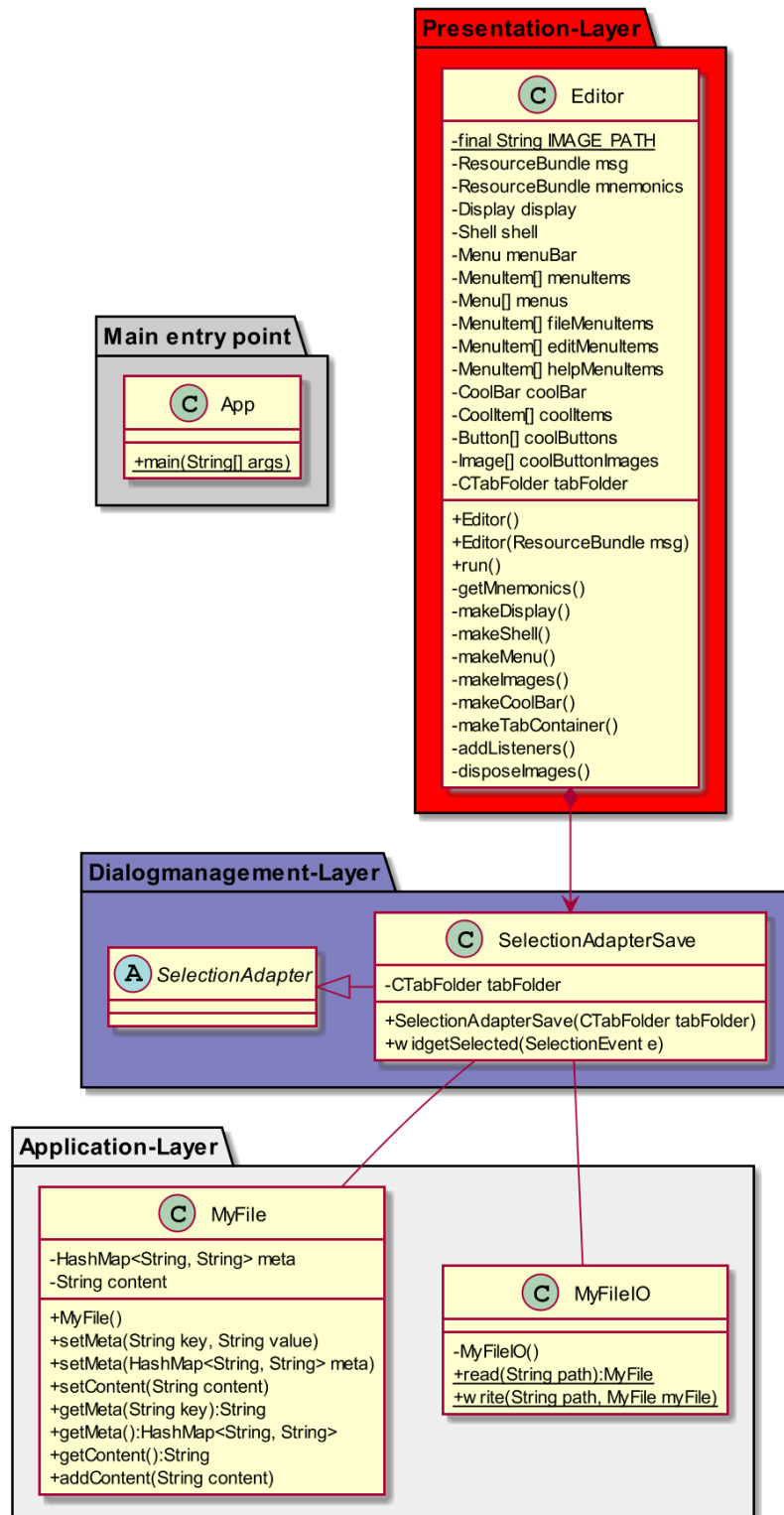


Abbildung 10: Tab in Datei speichern

2 Implementierung unter Eclipse/SWT

Den Quelltext für die unter Eclipse/SWT programmierte Editor-Anwendungen ist auf Github zu finden: <https://github.com/xsjad0/simple-editor>.

Da wir eine Gruppe von drei Studenten sind, haben wir zusätzlich den Quellcode von jedem einzelnen auf Ilias hochgeladen. Die Editor-Anwendungen basiert zudem auf Maven. Damit die Eclipse/SWT Bibliothek genutzt werden kann, muss dementsprechend die pom.xml erweitert werden.

```
1      ....
2      <dependency>
3          <groupId>org.eclipse.swt</groupId>
4          <artifactId>org.eclipse.swt.win32.win32.x86_64</artifactId>
5          <version>4.3</version>
6      </dependency>
7      ...
```

Auflistung 1: Maven SWT Abhängigkeit

3 Internationalisierung der Anwendung

Mit der Internationalisierung einer Anwendung kann die Anwendung ohne Änderung des Quelltextes um beliebige Sprachen erweitert werden. Grundsätzlich gibt es in Java zwei Möglichkeiten eine Anwendung zu internationalisieren: entweder werden die entsprechenden Texte in Ressourcen-Dateien ausgelagert und diese wiederum von der Anwendung geladen, um die Texte je nach Sprache angezeigt zu bekommen oder es werden separate Java-Klassen geschrieben, die der Anwendung ein Array von Objekten zurückliefert. In unserer Anwendung wurden beide Wege realisiert, jedoch ist die zweite Methode über die Java-Klassen aktiv. Folgend ist der Inhalt der entsprechenden Dateien für die deutsche Sprache zu sehen.

```
1 package dev.swt.gui;
2 import java.util.ListResourceBundle;
3 /**
4  * German Messages
5  */
6 public class ResourceBundle_de_DE extends ListResourceBundle {
7     @Override
8     protected Object[] [] getContents() {
9         return contents;
10    }
11    private Object[] [] contents = { { "editorTitle", "Text-Editor" },
12        { "editorTabText", "unbenannt" }, { "editorFile", "&Datei" },
13        { "editorEdit", "&Editieren" }, { "editorHelp", "&Hilfe" },
14        { "editorNew", "&Neu" }, { "editorOpen", "&Oeffnen" },
15        { "editorSave", "&Speichern..." }, { "editorQuit", "&Beenden"
16        },
17        { "editorColor", "Text &Farbe" }, { "editorVersion", "&Version"
18        },
19        { "colorTitle", "Farb-Auswahl" }, { "colorInfo", "Bitte waehlen
20        Sie eine Textfarbe aus" },
21        { "colorGroup", "Textfarbe" }, { "colorRed", "rot" },
22        { "colorGreen", "gruen" }, { "colorBlue", "blau" },
23        { "colorOkay", "Ok" }, { "colorCancel", "Abbruch" },
24        { "version", "Version 1.3" }, { "owner", "Marius Schenzle" },
25        { "stopClosing", "Modifizierte Tabs entdeckt.\nWollen Sie
26        trotzdem schliessen?" } };
```

Auflistung 2: Java Ressourcen-Klasse

```
1 editorTitle=Text-Editor
2 editorFile=&Datei
3 editorEdit=&Editieren
4 editorHelp=&Hilfe
5 editorNew=&Neu\tCtrl+N
6 editorOpen=&Oeffnen...\tCtrl+O
7 editorSave=&Speichern...\tCtrl+S
8 editorQuit=&Beenden\tCtrl+Q
9 editorColor=Text &Farbe\tCtrl+C
10 editorVersion=&Version
11 editorTabtext=unbenannt
12
13 colorTitle=Farb-Auswahl
14 colorInfo=Bitte waehlen Sie eine Textfarbe aus
15 colorGroup=Textfarbe
16 colorRed=rot
17 colorGreen=gruen
18 colorBlue=blau
19 colorOkay=Ok
20 colorCancel=Abbruch
```

Auflistung 3: Java Ressourcen-Datei

Abbildungsverzeichnis

1	Editor Applikation	1
2	Auswahl der Textfarbe im aktuellen Tab	2
3	Tabs öffnen und schließen	2
4	Farb- und Versions-Dialog	3
5	Aktionen im Datei-Menü	4
6	Präsentationsschicht	5
7	Dialogmanagementschicht	6
8	Applikationsschicht	6
9	Weitere erstellte Klassen	6
10	Tab in Datei speichern	7

Auflistungen

1	Maven SWT Abhängigkeit	8
2	Java Ressourcen-Klasse	9
3	Java Ressourcen-Datei	10