

Themenblatt

GUI in Java

1 Allgemeines

Die Umsetzung einer graphischen Benutzeroberfläche (englisch Graphical User Interface (GUI)) ist ein zentraler Bestandteil des Simulator-Projekts.

Aus eigener Erfahrung empfiehlt es sich die GUI zunächst bei der Erfassung der Problemstellung nicht zu berücksichtigen. Es sollte also davon ausgegangen werden, dass eine Ausgabe aller relevanten Informationen des Simulators auf der Konsole erfolgt. Dies hilft dabei sich auf die Wesentlichen Problemstellungen zu fokussieren. Nach Erfassung der grundlegenden Problemstellung sollte ein Mockup erstellt werden. Durch ein Mockup werden die Ergebnisse aus der Problemanalyse visualisiert. Außerdem kann mithilfe eines sorgfältig geplanten Mockups die Implementierung der GUI beschleunigt werden. Für die Erstellung existieren zahlreiche Tools wie Microsoft Blend, Pencil oder Balsamiq.

Diese Vorgehensweise ermöglicht es die Konsolenausgaben einfach in eine graphische Benutzeroberfläche abzubilden.

Dieses Themenblatt bespricht Implementierungsdetails mit der Programmiersprache Java. In Java gibt es mehr oder weniger brauchbare Frameworks, um eine GUI zu implementieren. Beispiele für brauchbare Frameworks sind:

- Swing
- JavaFX
- Standard Widget Toolkit (SWT)

Jedes dieser Frameworks hat seine Feinheiten (bzw. Tücken). Die Auswahl des Frameworks sollte in erster Linie anhand des eigenen Wissenstands erfolgen. Dieses Themenblatt beschreibt das Erstellen einer GUI mit JavaFX. Jedoch können die diskutierten Inhalte auch abstrahiert und auf andere Frameworks übertragen werden.

2 Installation von JavaFX

Eine Anleitung für die Installation ist auf der offiziellen Webseite von JavaFX zu entnehmen. <https://openjfx.io/openjfx-docs/>

Für diejenigen die mit Build-Management-Tools, wie Maven oder Gradle, noch keine Erfahrungen gesammelt haben sollten sich an dem Kapitel „Run HelloWorld using JavaFX“ orientieren. Ansonsten kann das entsprechende Unterkapitel zu dem Build-Management-Tool als Anleitung gewählt werden. Es muss bei beiden Varianten darauf geachtet werden, dass eine kompatible Java-Version auf dem System installiert ist. Diese Information kann in dem Kapitel „Install Java“ abgerufen werden.

3 GUI-Komponenten

Im Rahmen der Erstellung der GUI muss festgelegt werden, welche Komponenten zur Darstellung der entsprechenden Funktionen verwendet werden. Es folgt eine Auflistung von Komponenten, die sehr

wahrscheinlich eingesetzt werden müssen. Diese Liste ist mehr als Hilfestellung anzusehen und nicht als Vorgabe.

- Labels
- Textfelder
- Tabellen
- Buttons
- Slider

Gute Quellen für die Verwendung der GUI-Komponenten sind unter anderem die aufgelisteten Webseiten. Gerne können nützliche Quellen auch an den Dozenten/Tutoren weitergeleitet werden, sodass diese in das Themenblatt aufgenommen werden können.

- https://docs.oracle.com/javafx/2/ui_controls/jfxpub-ui_controls.htm
- https://docs.oracle.com/javafx/2/ui_controls/table-view.htm
- <http://tutorials.jenkov.com/javafx/index.html>

4 UI-Thread

In der Vorlesung „Software Ergonomie“ wird bereits behandelt, dass Ausgaben in einer graphischen Benutzeroberfläche immer von einem UI-Thread getriggert werden müssen. Im Falle von JavaFX ist das der Main-Thread. Dieser wird automatisch beim Start der Anwendung erzeugt.

Wird innerhalb des Simulators ein neuer Thread gestartet (mit `new Thread().start()`) muss der oben beschriebene Sachverhalt eingehalten werden. In JavaFX kann dies mit dem folgenden Code erreicht werden.

```
Platform.runLater(() -> {  
    // UI operation here  
});
```

5 Beispiel einer GUI für den Simulator

The screenshot displays the PIC16F84 Simulator interface. At the top, there are control buttons: Run, Stop, Step, and Reset. A frequency slider is set to 4000000 Hz, and a time scale is shown as 0.0 µs.

The main area is divided into two panes. The left pane shows assembly code with addresses and comments. The right pane shows the current state of the processor registers.

Assembly Code (Left Pane):

```

00004 ;(c) St. Lehmann
00005 ;Ersterstellung: 23.03.2016
00006 ;19.05.2020
00007 ;mod. 18.10.2018 Version H50
00008 ;
00009 list c=132 ;Zeilenlänge im LST auf 132 Zeichen setzen
00010
00011
00012 ;Definition des Prozessors
00013 device 16F84
00014
00015 ;Festlegen des Codebeginns
00016 org 0
00017 start
--> 0000 3011 00018 movlw 11h ;in W steht nun 11h, Statusreg. unverändert
0001 3930 00019 andlw 30h ;W = 10h, C=x, DC=x, Z=0
0002 3800 00020 iorlw 00h ;W = 10h, C=x, DC=x, Z=0
0003 3C30 00021 sublw 30h ;W = 20h, C=1, DC=1, Z=0
0004 3A20 00022 xorlw 20h ;W = 00h, C=1, DC=1, Z=1
0005 3E25 00023 addlw 25h ;W = 25h, C=0, DC=0, Z=0
00024
00025

```

Registers (Right Pane):

Address	Value (HEX)	Value (Binary)
0x0C	0x00	0b00000000
0x0D	0x00	0b00000000
0x0E	0x00	0b00000000
0x0F	0x00	0b00000000
0x10	0x00	0b00000000
0x11	0x00	0b00000000
0x12	0x00	0b00000000
0x13	0x00	0b00000000
0x14	0x00	0b00000000
0x15	0x00	0b00000000
0x16	0x00	0b00000000
0x17	0x00	0b00000000
0x18	0x00	0b00000000
0x19	0x00	0b00000000
0x1A	0x00	0b00000000
0x1B	0x00	0b00000000
0x1C	0x00	0b00000000
0x1D	0x00	0b00000000
0x1E	0x00	0b00000000

Console (Bottom Pane):

```

2020/09/10 01:58:37: Leaving state 'SimStateNoFileLoaded'
2020/09/10 01:58:37: Entering state 'SimStateFileLoaded'
2020/09/10 01:58:37: Leaving state 'SimStateFileLoaded'
2020/09/10 01:58:37: Entering state 'SimStateReset'
2020/09/10 01:58:37: Leaving state 'SimStateReset'
2020/09/10 01:58:37: Entering state 'SimStateIdle'

```