Einführung in Software Engineering

Übung #06

TECHNISCHE UNIVERSITÄT DARMSTADT

Aufgabenblatt im Web:

https://cage.st.informatik.tu-darmstadt.de/eise/public/exercise/ex06

Dr.-Ing. Michael Eichberg
Dipl.-Inform. Ralf Mitschke

Übungsinhalte

System-Sequenz-Diagramme

- **Überblick über einen Ablauf in der Flashcards-Anwendung**

Sequenz-Diagramme

- **Detaillierte Veranschaulichung von AWT-Events**

Aufgabe 1

Ablaufbeschreibung des Lernens mit der Lernkartei

Lernen einer Lernkartei – Ablaufbeschreibung

Der Benutzer startet den Lernprozess in der Flashcards-Anwendung. In der Anwendung öffnet sich ein Lerndialog, in dem der Benutzer nacheinander seine Karteikarten lernt. Der Lernprozess soll in einem Durchlauf alle im System vorhanden Karten abfragen. Die abgefragten Karten werden in einer zufälligen Reihenfolge gelernt. Hierzu wird innerhalb des Systems jeweils zufällig eine noch nicht gelernte Karte ermittelt. Dann präsentiert die Flashcards-Anwendung dem Benutzer die Frageseite der nächsten Karteikarte. Dies geschieht auch innerhalb des Lerndialogs. Der Benutzer überlegt sich eine Antwort auf die Frage der Karteikarte. Anschließend wendet er die Karteikarte, um seine Antwort auf Korrektheit zu überprüfen. ...

System-Sequenz-Diagramm erstellen

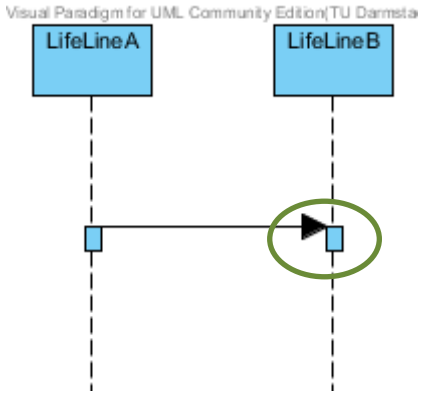
Achtung:

Die Interaktion endet an der Systemgrenze

(das System wird als Black-Box aufgefasst)

Aufgabe 1 – Diagramm Notation

System-Sequenz-Diagramme haben eine Notation **ohne** Aktionsbalken



In Visual-Paradigm **nicht unterstützt**!

Wir akzeptieren Lösungen mit Aktionsbalken

Aufgabe 2 - Einleitung

Was passiert wirklich in AWT,

wenn Events für einen Button gefeuert werden!

Zum Beispiel - Vom Mausklick bis zum Aufruf an `learn()`

```
playButton = Utilities.createToolBarButton(" Learn ",
      "media-playback-start.png", "learn flashcards");
playButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent event) {
                  learn();
            }
});
```

Aufgabe 2 - Einleitung

`JButton` unterstützt drei Arten von Events:

- Action-Events, Change-Events und Item-Events

Interessenten implementieren ein Interface

- **`ActionListener`**`, `**`ChangeListener`** `oder` **`ItemListener`**

Interessenten registrieren sich an einem Button

- z.B. mit **`addActionListener`**

Aufgabe 2 - Einleitung

Vom Mausklick bis zu den Events des Buttons passiert sehr viel!

Die ersten Schichten sollen **nicht modelliert** werden.

Eventhandling Loop von AWT

- erhält Initialen Mausklick mir Koordinaten

Dispatcher in AWT

- ermittelt die konkreten Komponenten (Buttons, Listen, andere Widgets) die in den Koordinaten des Mausklicks liegen

Die Modellierung für diese Aufgabe beginnt in:

javax.swing.plaf.baisc.BasicButtonListener

Aufgabe 2

Erstellen Sie **ein** Sequenz-Diagramm für die Interaktion in AWT bei

Drücken eines Buttons

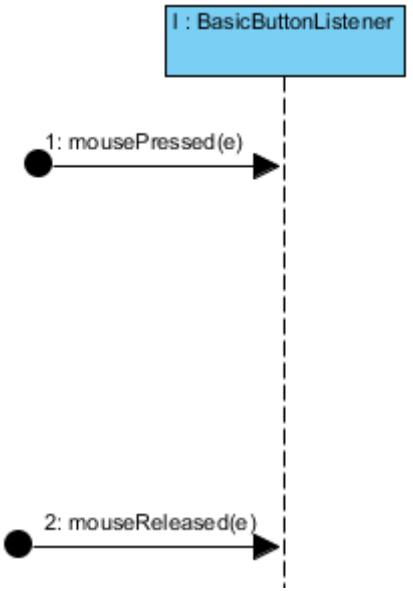
(Dies gliedert sich in „Maus drücken“ und „Maus loslassen“)

Bis zu abfeuern aller Events des Buttons

Das Sequenz-Diagramm startet mit den Aufrufen an **BasicButtonListener**

1. mousePressed(MouseEvent)

2. mouseReleased(MouseEvent)



Aufgabe 2

Der JButton informiert seine Listener über die Methoden

`fireActionPerformed(ActionEvent)`

`fireStateChanged()`

`fireItemStateChanged(ItemEvent)`

Für die Methode `fireActionPerformed(ActionEvent)` soll die Interaktion mit registrierten Instanzen von ActionListener aufgezeigt werden.

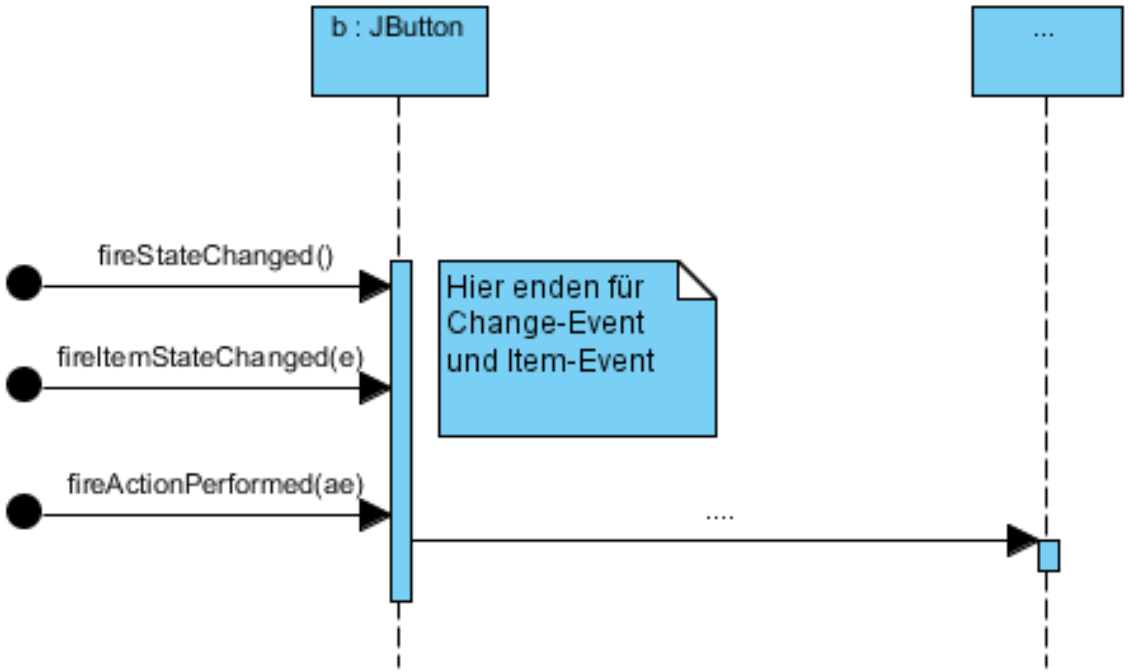
Anderen Events sollen nur modelliert werden bis:

`Jbutton.fireItemStateChanged(ItemEvent)`

`Jbutton.fireStateChanged()`

Aufgabe 2

Am ‚Ende' des Sequenzdiagramms nur Action-Events detailliert modellieren



Aufgabe 2

Alle Methodenaufrufe und Objekte sollen modelliert werden, die zum Pfad der Aufrufe an Event Listener beitragen

Vereinfachungen:

Konditionale, deren Blöcke (If, oder Else) in diesem Szenario nicht ausgeführt werden, müssen nicht modelliert werden

Konditionale, deren Auswertungsergebnis bereits bekannt ist führen nur den entsprechenden Block aus

Spielregel:

Ihr könnt eigene Vereinfachungen treffen! Dokumentiert diese nachvollziehbar in eurer Lösung!

Setup für JDK-Sourcen