Einführung in Software Engineering

Department of Computer Science
Software Technology Group
Dr. Michael Eichberg, Sven Amann

TECHNISCHE UNIVERSITÄT DARMSTADT

Übungszettel 08: Generelle Informationen zur Abgabe

Abgabefrist: 14.01.2016, 08:00 Uhr
(Alle Lösungen müssen bis zu den jeweiligen Zeitpunkten eingereicht sein)

Erzeugen Sie aus den Lösungen aller Teilaufgaben, die keinen Quelltext erfordern, ***eine*** *Datei „****solution_xxx_ex08.pdf****". Fügen Sie für xxx Ihre Gruppennummer ein. Gliedern Sie die Abgabedatei den Teilaufgaben entsprechend.* Achten Sie auf korrekte Rechtschreibung und Grammatik. Die Tutoren sind angewiesen chaotische, unverständliche oder fehlerüberhäufte Lösungen zu ignorieren.

Alle Lösungen sind in Moodle einzustellen. Abgaben per E-Mail oder auf anderem Weg werden nicht berücksichtigt. Beachten Sie unbedingt die in der Übung vorgestellten Namenskonventionen. Ihre Abgabe wird nicht bewertet, wenn der Tutor die Lösungen nicht finden oder nicht zuordnen kann.

Die Korrekturen werden ebenfalls in Moodle eingestellt. Wir versuchen dies im Normalfall eine Woche nach Abgabe der Übung zu schaffen. Wenn Sie Fragen bzgl. der Korrekturen haben, schreiben Sie dem korrigierenden Tutor eine E-Mail.

Aufgabe 1: Systemsequenzdiagramm (7 Punkte)

Modellieren Sie den im Folgenden beschriebenen Ablauf eines Zuges im Online-Wettbewerb „Die Siedler von Catan" **aus der Übung 04** als Systemsequenzdiagramm. Es geht hierbei um die Interaktion des Benutzers mit der Anwendung, nicht um die Abläufe in der Anwendung selbst.

Hinweis: Für das modellieren von Systemsequenzdiagrammen ist das Online-Werkzeug https://www.websequencediagrams.com/ sehr hilfreich.

Einen Zug durchführen im Online-Wettbewerb „Die Siedler von Catan" – Ablaufbeschreibung

Der Benutzer startet die Onlineanwendung „Die Siedler von Catan". In der Anwendung öffnet sich ein Dialog, in dem der Benutzer seinen Namen einträgt und einem Onlinespiel beitritt. Der Spieler wartet bis alle Teilnehmer dem Spiel beigetreten sind und wird dann zum Spielbrett weitergeleitet. Das System wählt nach einem Zufallsverfahren aus, welcher Spieler beginnt. Ist ein Spieler am Zug, wird zunächst gewürfelt und danach die Rohstofferträge nach gewürfelter Augenzahl verteilt. Hierzu wird innerhalb des Systems das Spielbrett auf die gewürfelte Augenzahl überprüft und dem jeweiligen Spieler die Rohstofferträge nach Bebauung gutgeschrieben. Der Spieler hat nun die Auswahl zwischen Handeln, Bauen oder Aussetzen. Um aussetzen zu können, wartet der Spieler 5 Minuten oder wählt direkt fortfahren. Beim Handeln öffnet sich ein Dialogfenster, welches dem Spieler die Wechselkurse der Bank (4:1) und die Option mit anderen Spielern zu handeln anzeigt. Sollte der Spieler „mit anderen Spielern handeln" auswählen, kann er anderen Spielern jeweils einmal eine Nachricht mit Wechselkursen übermitteln. Sollte ein Spieler mit einem Wechselkurs einverstanden sein, kann er den Kurs auswählen und dem Tausch zustimmen. Sollte ein Spieler einem Tausch zustimmen oder ein Wechselkurs der Bank gewählt werden, werden die Rohstoffe direkt umverteilt. Wenn der Benutzer bauen möchte, werden auf dem Spielbrett alle Flächen ausgegraut, die schon bebaut sind und ihm seine Möglichkeiten der Bebauung in einem Dialogfenster angezeigt. In diesem Dialogfenster kann der Spieler seine Bebauungsstrategie auswählen und ausführen. Die Aktionen Handeln oder Bauen kann ein Spieler während seines Zuges so oft durchführen, wie er Rohstoffe zur Verfügung hat. Hat ein Spieler den Bauvorgang begonnen, kann er während dieses Zuges nicht mehr mit der Bank oder anderen Spielern handeln. Sollte ein Spieler die festgelegte Anzahl an Siegpunkten erreichen, die zum Gewinnen nötig sind, werden alle Spieler über ihren Sieg oder ihre Niederlage informiert und die Punkte des Siegers zu seiner Gesamtwertung aufsummiert. Spieler, die verloren haben, werden aus dem weiteren Wettbewerb ausgeschlossen, dazu kann die Anwendung nur per Verifikation der E-Mailadresse heruntergeladen werden.

Aufgabe 2: Darstellung komplexer Abläufe mit Hilfe eines Sequenzdiagramms (7 Punkte)

Im Folgenden soll die Kommunikation innerhalb des AWT/Swing-Frameworks (des JDK 1.8) verdeutlicht werden, die beim Drücken eines JButtons stattfindet.

Hintergrundinformationen zu AWT/Swing:

Die Swing-Klasse JButton unterscheidet drei Arten von Events: Action-Events, Change-Events und Item-Events. Interessenten für diese Events implementieren entsprechend eines der Interfaces ActionListener, ChangeListener oder ItemListener und registrieren sich bei einer Instanz von JButton. Bis zum Aufruf der eigenen registrierten Methode(n) findet ein komplexer Prozess innerhalb des AWT-Frameworks statt. Beim Klicken der Maus bildet AWT die Mauskoordinaten auf diejenige AWT-Komponente (z.B. Button, Combobox, etc.) ab, die auf dem entsprechenden Bildschirmbereich der Mausinteraktion liegt. Diese Komponenten werden dann durch AWT darüber benachrichtigt welche Art von Interaktion stattfand.

Der ActionListener erweitert den MouseListener, der für die Verarbeitung von Mouse-Events zuständig ist. Der ActionListener berücksichtigt zusätzlich auch Tastaturevents. D.h. die Methode actionPerformed wird aufgerufen, wenn ein Button angeklickt wird oder die Leertaste der Tastatur gedrückt wird während der Button ausgewählt ist. Wenn man nur an Mausklicks interessiert ist, kann man sich auch nur als MouseListener registrieren.

Beispiel eines AWT/Swing ActionListeners:

In der folgenden Anwendung wird für einen Dialog der playButton erstellt, wie im untenstehenden Code-Beispiel gezeigt. Durch das Registrieren einer Instanz vom Typ ActionListener wird die Anwendung informiert, wenn der entsprechende JButton gedrückt wurde.

```
public FlashcardsWindow(FlashcardSeries series) {
  …
    playButton =
          Utilities.createToolBarButton("Learn","media-playback-start.png", "learn flashcards");
    playButton.addActionListener(ae -> { learn(); });
  …
}
```

Aufgabe:

Erstellen Sie **ein** Sequenzdiagramm, das für einen beliebigen JButton aufzeigt, welche Action-, Change- oder Item-Events im Laufe eines Mausklicks durch den Button ausgelöst werden. Ein Mausklick besteht aus zwei Events vom Typ MouseEvent mit den Identifiern MouseEvent.MOUSE_PRESSED und MouseEvent.MOUSE_RELEASED. Starten Sie Ihr Diagramm mit dem Aufruf der Methode mousePressed(MouseEvent), bis der Kontrollfluss zu dieser Methode zurückkehrt, und modellieren Sie anschließend im gleichen Diagramm den Aufruf mouseReleased(MouseEvent). Beide Methoden finden Sie in der JDK 1.8 Klasse javax.swing.plaf.basic.BasicButtonListener. Ihr Sequenzdiagramm soll das wesentliche Szenario, welches zum Auslösen von Action-, Change- oder Item-Events auf JButton führt, aufzeigen. Für jede Art von Event wird in einem JButton letzten Endes eine der folgenden Methoden aufgerufen, die alle entsprechenden Listener benachrichtigt:

- JButton.fireActionPerformed(ActionEvent)
- JButton.fireStateChanged()
- JButton.fireItemStateChanged(ItemEvent)

Sie müssen die Interaktion mit registrierten Listener-Instanzen nur für die Methode fireActionPerformed aufzeigen. Für die anderen Events müssen Aufrufe über fireItemStateChanged und fireStateChanged hinaus nicht weiter verfolgt werden. (**Vorsicht:** Es gibt gleichnamige Methoden auf anderen Typen, hier geht es ausschließlich um JButton!)

Für das Sequenzdiagramm sollen Sie folgende Vereinfachungen vornehmen, um dem Leser des Diagramms die wesentliche Funktion auf einfache Weise zu verdeutlichen:

1. Konditionale, deren Blöcke (if oder else) in diesem Szenario nicht ausgeführt werden, müssen nicht modelliert werden. D.h. Konditionale, deren Auswertungsergebnis bereits bekannt ist, führen nur den entsprechenden Block aus.
2. Es sollen nur Methodenaufrufe und Objekte modelliert werden, die zum Pfad der Aufrufe an EventListener beitragen. Dies kann im Wesentlichen erreicht werden, indem Sie Ihr Diagramm auf Instanzen der folgenden Typen bzw. deren Implementierungen oder Methoden aus Superklassen oder enthaltene innere/anonyme Klassen beschränken: JButton, MouseEvent, BasicButtonLis-

tener, ButtonModel, ActionListener.
Werte, die durch Aufrufe an andere Klassen errechnet werden, können als gegeben vorausgesetzt werden und müssen nicht modelliert werden.

3. Sie sollen für dieses Szenario davon ausgehen, dass die Maus zwischen den Klicks nicht bewegt wurde und dass es sich um einen Klick mit der linken Maustaste handelt.
4. Sie sollen davon ausgehen, dass der Button den Zustand ‚enabled' hat.

Sie können weitere Vereinfachungen für Ihr Diagramm treffen, sofern diese nachvollziehbar dokumentiert sind!

Hinweis zur Vorgehensweise:

Zur Verdeutlichung der Funktionsweise von AWT bietet es sich an, den Eclipse-Debugger zu verwenden, um die relevanten Aufrufe in einem selbst geschriebenen Beispiel zu analysieren.

Hinweis zur Modellierung:

Methoden-Dispatch wird nicht in UML Sequenzdiagrammen modelliert. Methodenaufrufe auf Interfaces oder abstrakte Klassen sind z.B. immer Nachrichten an konkrete Instanzen von Klassen, die das Interface implementieren.

Frohe Weihnachten und ein schönes neues Jahr!!!