

Übungen zu Einführung in die Software-Entwicklung

Sommersemester 2021

Blatt 10

Dies ist das letzte reguläre Aufgabenblatt für diese Veranstaltung. Sollten Sie bereits 9 Testate in diesem Semester bestanden haben, ist das letzte Testat für Sie optional. Sollten Sie vorhaben nicht zum Testat zu kommen, informieren Sie ihren Tutor bitte frühzeitig darüber. Beachten Sie bei ihrer Entscheidung, dass alle Inhalte auf diesem Blatt prüfungsrelevant sind.

Aufgabe 10.1: MVC - MineSweeper (80 Punkte)

Implementieren Sie mit den Swing-Komponenten von Java eine vereinfachte grafische Version des (hoffentlich bekannten) Spiels „MineSweeper“: Auf einem rechteckigen Spielfeld sind eine bekannte Anzahl von Minen versteckt, die vom Spieler gefunden werden müssen. Dazu kann der Spieler zwei Operationen ausführen:

- Ein Klick mit der linken Maustaste deckt ein Feld auf. Lag auf dem Feld eine Mine, hat der Spieler verloren und das Spiel ist beendet. Andernfalls wird die Anzahl der Minen in den (maximal 8) umliegenden Feldern auf dem Feld angezeigt. Ist diese Zahl 0, werden automatisch auch alle direkt benachbarten Felder aufgedeckt
- Ein Klick mit der rechten Maustaste markiert ein Feld. War das Feld bereits markiert, wird die Markierung wieder entfernt. Ein markiertes Feld wird bei einem Klick mit der linken Maustaste nicht aufgedeckt.

Das Spiel endet, wenn der Spieler eine Bombe aufdeckt oder alle Felder aufgedeckt worden sind, auf denen keine Bomben liegen.

Richten Sie sich bei der Implementierung nach dem in der Vorlesung vorgestellten *Model-View-Controller*-Entwurfsmuster in Verbindung mit dem *Beobachter-Muster*. Gehen Sie dazu schrittweise vor:

1. **Model** Entwerfen Sie Klassen zur Repräsentation der Spiellogik. Das Spiel selber als auch alle darin enthaltenen Felder sollten durch eigene Objekte dargestellt werden können. Achten Sie darauf, dass solche Klassen, die später mit der *View* kommunizieren sollen, vom Typ `java.util.Observable` sind.
2. **View** Implementieren Sie die grafische Oberfläche für Ihr Spiel unter der Verwendung von *Swing-Komponenten*. Verknüpfen Sie die jeweiligen Komponenten wenn nötig mit dem *Model*, in dem Sie das Interface `java.util.Observer` implementieren. Ist ein Spiel beendet, soll der Benutzer über einen Dialog darüber benachrichtigt werden, ob er gewonnen oder verloren hat. Ein laufendes Spiel könnte beispielsweise wie in der Abbildung unten aussehen.

3. **Controller** Verknüpfen Sie die Eingabemöglichkeiten der grafischen Benutzeroberfläche über eine oder mehrere Erweiterungen der notwendigen *Listener*-Klassen mit dem *Model*. Wenn ein *Controller* Änderungen am *Modell* verarbeiten soll, sollte er das Interface `java.util.Observer` implementieren. Eingaben in der *View* sollten von dem *Controller* direkt an das *Model* weitergegeben werden.
4. Führen Sie *Model*, *View* und *Controller* in einer separaten Klasse zusammen. Diese sollte über die Kommandozeilenparameter mit der Höhe und Breite des Spielfeldes sowie die Anzahl der zufällig verteilten Bomben aufgerufen werden können und ein entsprechendes Spiel erzeugen und starten. Es ist nicht nötig, dass aus der grafischen Benutzeroberfläche heraus ein neues Spiel gestartet werden kann.



Abbildung 1: View eines laufenden Minesweeper-Spiels, das mit 5 * 15 Feldern und 10 Bomben erzeugt wurde. Markierte Felder sind mit einem ! gekennzeichnet.

Hinweise

- Ihr *Model* sollte vollständig sein, also auch Methoden zum Aufdecken der Felder und zum Setzen der Markierungen etc. enthalten. Implementieren Sie das *Model* derart, dass *View* und *Controller* später leicht ausgetauscht werden können. Dazu sollte auch das *Model* wiedergeben, ob ein Spiel gewonnen bzw. verloren wurde oder ob das Spiel noch im Gange ist.
- Sie könnten beispielsweise zuerst ausschließlich das *Model* implementieren und testen, ob alle Elemente der Spiellogik abgedeckt sind, bevor Sie die restlichen Komponenten umsetzen.
- Es ist hilfreich, wenn Sie sich vor der eigentlichen Implementierung intensiv mit der Entwicklung Ihrer Klassenhierarchie beschäftigen und festlegen wie und wann die einzelnen Komponenten miteinander kommunizieren.

Aufgabe 10.2: Fragen (20 Punkte)

Beantworten Sie Ihrer Tutorin / Ihrem Tutor Fragen zu den Themen Observer-Pattern, Model-View-Controller und der Umsetzung von GUIs in den Java-Frameworks AWT und SWING.