

## Modellierung und Programmierung 1 Übungsserie 6

Abgabetermin: 02.02.2014, 23:55 Uhr

Grundsätzlich sind Nebenrechnungen anzugeben und Antworten zu begründen.  
Einzureichen sind, bei mehreren Dateien als .zip-Archiv:  
Lösungen und UML-Klassendiagramme als .pdf-Datei, Programme als Quellcode,  
Ergebnisdateien, Online-Dokumentationen.

Für die Stimmabgabe bei einer Wahl soll eine Software entwickelt werden:  
Parteien können sich zur Wahl registrieren lassen. Die Stimmabgabe jedes Wählers erfolgt durch Auswahl genau einer dieser Parteien. Nach jeder Stimmabgabe wird der aktuelle Stand der Wahl für jede Partei prozentual ermittelt.

Die Entwicklung der Software erfolgt in zwei Schritten:  
Zunächst wird ein Modell für die Kernfunktionalität unabhängig von einer Oberfläche entworfen und getestet. Auf dessen Grundlage wird anschließend eine Oberfläche mit Hilfe der MVC-Architektur ausgearbeitet.

### 1. Modellierung

Entwickeln Sie eine UML-Klassenhierarchie für die Kernfunktionalität der Wahl. Dabei sind folgende Anforderungen und Szenarien zu berücksichtigen:

- Eine Partei meldet sich *vor* der ersten Stimmabgabe an. Sie wird durch ihren Namen und ihre Parteifarbe registriert. Eine wiederholte Anmeldung bzw. eine verspätete Anmeldung ist nicht möglich.
- Die Wahl läuft *mit* der ersten Stimmabgabe. Ein Wähler gibt seine Stimme für genau eine der Parteien ab. Die abgegebene Stimme wird der Partei gutgeschrieben.
- Der aktuelle Wahlzwischenstand in Prozent wird für eine angemeldete Partei anhand der insgesamt abgegebenen Stimmen und der Stimmen der Partei berechnet.

Hinweis: Beachten Sie bei der Entwicklung der Klassen, dass in der objektorientierten Programmierung Objekte miteinander kommunizieren.

### 2. Implementierung

Implementieren und testen Sie Ihre UML-Klassenhierarchie für die Kernfunktionatität der Wahl:

- Implementieren Sie Ihren Entwurf für das Modell aus Aufgabe 1.
- Schreiben Sie ein Testprogramm **WahlTest.java** als Konsolenanwendung, indem Sie drei fiktive Parteien anlegen und 100 Stimmen zufällig vergeben, anschließend für jede Partei sowohl die absolute Anzahl der Stimmen als auch das Wahlergebnis in Prozent ausgeben. Wiederholen Sie den Test mit 50 000 Stimmen. Leiten Sie die Ausgaben in eine Datei **WahlTest100.out** bzw. in eine Datei **WahlTest50000.out** um.

### 3. GUI / MVC - Architektur

Entwickeln Sie für die Stimmabgabe eine Oberfläche mit den zentralen Klassen **WahlModel.java**, **WahlView.java** und **WahlController.java** der MVC-Architektur. Dabei soll der Stand der

---

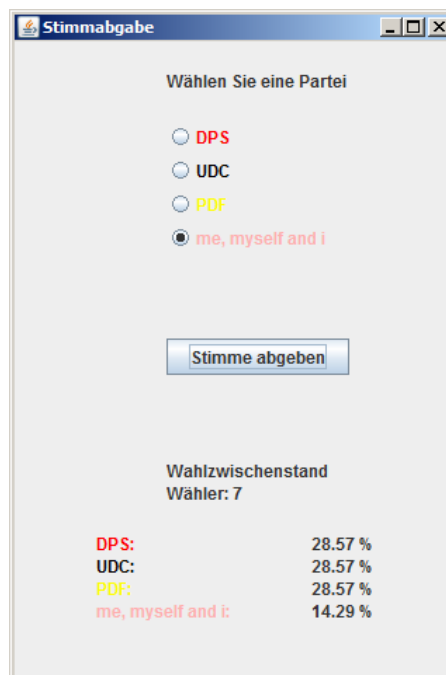
Wahl mit Hilfe der Überwachungsmechanismen der MVC-Architektur nach jeder Stimmabgabe automatisch aktualisiert werden.

a) **WahlModel.java**

Grundlage für das Model der MVC-Architektur bilden die in 1. entworfenen und in 2. implementierten Klassen. Geben Sie an, welche Änderungen Sie vornehmen müssen, damit Ihr Modell in einer MVC-Anwendung verwendet werden kann. Übernehmen Sie die Klassen mit den entsprechenden Ergänzungen.

b) **WahlView.java**

Entwickeln Sie für den View einen GUI-Container (*JPanel*) zur grafischen Darstellung der Wahl, ohne Funktionalitäten. Zur Stimmabgabe werden die Parteien durch ihren Namen in ihrer jeweiligen Farbe angezeigt. Der Wahlzwischenstand gibt die Anzahl der bisherigen Wähler insgesamt und die erreichten Prozente der einzelnen Partei, gerundet auf 2 Stellen nach dem Komma, an.



Beispiel für eine einfache GUI, gestaltet mit *BoxLayout*.

c) **WahlController.java**

Für die Interaktion mit den Wählern ergänzen Sie den View entsprechend und implementieren einen passenden Controller. Installieren Sie dabei die Überwachung der Stimmabgabe: Eine Stimmabgabe wurde ausgelöst (Button „Stimme abgeben“). Der Controller fragt den View nach der gewählten Partei (aktiver Radiobutton) und teilt diese dem Model mit. Das Model verbucht für die Partei die Stimme. Der View registriert die Veränderung im Model und veranlasst seinerseits die Aktualisierung des Wahlzwischenstandes.

Hinweis: Nur die sich mit einer Stimmabgabe ändernden GUI-Komponenten werden im View als Attribute aufgenommen.

d) **WahlMain.java**

Ein Startprogramm trägt die angemeldeten Parteien für die Wahl ein und startet das Fenster für die Stimmabgabe.

e) Erzeugen Sie eine Online-Dokumentation Ihres Programms mit

```
javadoc -private -d WahlDoc *.java.
```

Hinweis:

Beachten Sie, dass auch alle *privaten* Attribute und Methoden zu dokumentieren sind.