Prof. Dr. Stefan Brass
Dipl.-Inform. Annett Thüring
Dipl.-Inform. Steffen Schiele
Dipl.-Inform. Andreas Bienert

Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik

Wintersemester 2014/15

Objektorientierte Programmierung: Hausaufgabenblatt 2

Abgabe: 06.11.2014 — Übung: 10./11.11.2014

Hausaufgabe 2: (4 Theoriepunkte)

Schreiben Sie ein Programm für folgendes Spiel:

- Am Anfang wird eine natürliche Zahl $g > 0$ ("Grenze") festgelegt. Eine natürliche Zahl s ("Summe") ist am Anfang 0.
- Es gibt zwei Spieler, die abwechselnd eine Zahl zwischen 1 und 3 eingeben. Diese Zahl wird jeweils auf die Summe s addiert. Der Spieler, bei dem erstmals die Grenze g überschritten wird (d.h. nach dem Spielzug ist $s > g$) hat verloren.

Das Programm soll also zunächst die Grenze g abfragen, und dann immer abwechselnd die beiden Spieler ("`Spieler 1`" und "`Spieler 2`") nach ihrem Zug fragen. Der Zug muss ein Wert zwischen 1 und 3 sein. Sollte ein Spieler eine andere ganze Zahl eingeben, soll das Programm "`Ungueltige Eingabe.`" ausgeben, und den gleichen Spieler erneut fragen. (Bei Eingaben, die keine Zahlen sind, darf das Programm mit einer "Exception" abbrechen — wir besprechen erst später, wie man das vermeidet.) Wenn das Spiel beendet ist, soll ausgegeben werden, wer gewonnen hat. Ein Dialog mit dem Programm könnte so aussehen (die Benutzer-Eingaben sind unterstrichen):

```
Grenze: 8
Spieler 1: 2
Spieler 2: 12
Ungueltige Eingabe.
Spieler 2: 3
Spieler 1: 3
Spieler 2: 1
Spieler 1 hat gewonnen.
```

Die aktuelle Summe wird im Verlauf des Spiels nicht ausgegeben — die Spieler müssen sie sich merken. Noch ein Tipp: Es wäre geschickt, in einer Variable zu speichern, welcher Spieler am Zug ist.

Schreiben Sie das Programm in die Datei "`Summe.java`". Geben Sie diese Datei über die Übungsplattform bis zum Donnerstag, 06.11.2014, 12:00 ab. Beachten Sie, dass bei der Einreichung geprüft wird, ob das Programm compilierbar ist. Falls nicht, können Sie es gar nicht hochladen. Wir verwenden eine automatische Vorkorrektur, deswegen halten Sie sich bitte genau an die Vorgaben (die Texte aus dem Beispiel-Lauf). Bitte achten Sie auch auf sinnvolle Einrückungen und allgemein ein verständliches Programm. Für schlechten Programmierstil können auch Punkte abgezogen werden.

Übungsaufgabe 2A: (ohne Abgabe)

Bitte bearbeiten Sie die Übungsaufgaben auf dem Hausaufgabenblatt, aber geben Sie diese Aufgaben nicht ab. Diese Aufgaben werden in der Übung besprochen. Sie müssen Ihre Lösung eventuell in der Übung vorführen.

Beschreiben Sie den Algorithmus, der Ihrem Spiel-Programm (Hausaufgabe 2) zugrunde liegt, auf etwas höherer Abstraktionsebene als den Java-Quellcode. Zeichnen Sie z.B. einen Programmablaufplan (Flussdiagramm) oder schreiben Sie einfach einen kurzen Text.

Sie dürfen auch "Pseudocode" verwenden — das ist ein Text, der Java-Quellcode relativ ähnlich sieht, aber nur für einen Menschen verständlich sein muss (nicht für den Compiler). Den Programm-Rahmen können Sie dabei weglassen, und die wesentlichen Strukturen z.B. mit "`if`" und "`while`" zeigen, aber Teile auch einfach als deutschen Text schreiben. Selbstverständlich dürfen Sie in Pseudocode auch Variablen benutzen, aber es könnten etwas weniger sein als im realen Java-Code. Das Ziel ist ja, dass Sie durch das Weglassen von Details das eigentliche Verfahren verständlicher und übersichtlicher machen.

Übungsaufgabe 2B: (ohne Abgabe)

Entwickeln Sie ein Syntaxdiagramm (oder mehrere) für ein ganz einfaches Textadventure-Spiel. Es soll folgende Kommandos verstehen:

- `quit`
- `go north`
  (und dasselbe mit `south`, `east`, `west`, außerdem soll "`go`" optional sein, d.h. die Himmelsrichtung alleine reicht)
- `take` ⟨Objekte⟩
  Dabei kann ⟨Objekte⟩ ein einzelner Gegenstand sein, oder auch eine mit `and` verknüpfte Liste von Gegenständen. Im Spiel gibt es `key`, `lamp`, `rope`, `crown`. Z.B. soll folgendes Kommand erlaubt sein: "`take rope and key and lamp`". Es soll legal sein, in einer solchen Liste den gleichen Gegenstand mehrfach aufzuführen, also z.B. "`take rope and rope`". Wenn Sie wollen, dürfen Sie optional den Artikel "`the`" vor Gegenständen erlauben.

Sie sollen also eine syntaktische Kategorie "Kommando" definieren, die genau die oben beschriebenen Folgen von Worten zulässt. Die einzelnen Worte sind das, was Sie in die Ovale schreiben. Selbstverständlich können Sie mehrere Syntaxdiagramme verwenden, z.B. zunächst ein Diagramm "Gegenstand" zeichnen, dessen Sprache gerade aus den Namen der vier Gegenstände besteht.

Auch diese Aufgabe bitte nicht einsenden, aber bis zur Übung bearbeiten.