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 5

Abgabe: 27.11.2014, 12:00 — Übung: 01./02.12.2014

Hausaufgabe 5: (4 Theoriepunkte)

“Mondlandung” ist ein klassisches Computerspiel (wir haben uns von “Coding for Fun” von Arnold Willemer inspirieren lassen). Die Mondfähre startet z.B. 4000m über dem Mond mit einer Geschwindigkeit von 0 m/s. Durch die Gravitationsbeschleunigung von 1.62 m/s^2 nimmt die Geschwindigkeit jede Sekunde um diesen Wert zu. Die Entfernung vom Mond sinkt entsprechend. Der Spieler hat die Möglichkeit, für jede Sekunde des Anflugs die Bremsraketen mit einer Stufe von zwischen 0 und 10 zu aktivieren. Jede Stufe entspricht dabei einer Beschleunigung von 0.4 m/s^2 in Richtung vom Mond weg. Maximal kann die Geschwindigkeit in einer Sekunde also um 4 m/s verringert werden (sie wächst aber gleichzeitig um 1.62 m/s durch die Gravitation). Der Treibstoff ist allerdings begrenzt. Am Anfang enthält der Tank 500 l. Pro Sekunde Betrieb der Bremsraketen wird Treibstoff entsprechend der gewählten Stufe verbraucht. Man könnte also maximal 50s mit Stufe 10 bremsen, danach wirkt nur noch die Gravitationsbeschleunigung.

Ziel des Spieles ist es, mit möglichst geringer Geschwindigkeit auf dem Mond aufzusetzen. Das Spiel ist zu Ende, wenn die Entfernung erstmals ≤ 0 wird. (Zur Vereinfachung wird dabei nicht berücksichtigt, dass der letzte Schritt eventuell kürzer als eine Sekunde ist.) Die Bewertung ist:

- “`Perfekte Landung!`”, falls die Geschwindigkeit nach dem letzten Zug $\leq$ 5m/s ist.
- “`Ok. Anstaendig gelandet.`” bei Geschwindigkeiten > 5m/s, aber $\leq$ 10m/s.
- “`Autsch. Gerade so.`”, wenn die Geschwindigkeit immerhin $\leq$ 20m/s ist.
- “`Raumfaehre zerstoert.`” sonst.

Wir haben eine erste Version schon programmiert, die Eingaben einliest, und die Geschwindigkeit und Höhe berechnet. Insbesondere haben wir die Methode `format` der Klasse `PrintStream` benutzt, um die Daten als Tabelle auszugeben. Im Format-String steht z.B. `%10s` für die Ausgabe einer Zeichenkette mit 10 Zeichen Breite, entsprechend `%10d` für eine ganze Zahl in einem Feld dieser Breite und `%10.1f` wird für `double`-Werte mit einer Nachkomma-Stelle verwendet. Die jeweiligen Datenwerte werden in den weiteren Argumenten der Methode übergeben. Außerdem haben wir die Eingabe so programmiert, dass man auch einfach “Return”/“Enter” drücken kann, und damit Stufe 0 auswählt. Das Programm ist unten abgedruckt und kann unter folgender URI heruntergeladen werden:

`http://www.informatik.uni-halle.de/~brass/oop14/homework/Mondlandung.java`

Ihre Aufgabe ist einerseits die Bewertung am Ende des Spiels und andererseits eine kleine Graphik aus Leerzeichen und Sternchen, die den Verlauf der Landung zeigt:

```
Ok. Anstaendig gelandet.
                                        *
                                        *
                                     *
                                 *
                           *
                    *
             *
        *
    *
   *
  *
 *
```

Dazu merken Sie sich in einem Array die aktuelle Entfernung vom Mond aller 10 Spielzüge (wenn die Zeit in Sekunden durch 10 teilbar ist). Die Deklaration und Erzeugung des Arrays und eventuell weitere nötige Variablen müssen Sie in das gegebene Programm einbauen, ebenso die Speicherung der Werte. Ein Array der Größe 100 sollte in jedem Fall ausreichen.

Am Ende des Spiels, nachdem Sie die Bewertung ausgegeben haben, schreiben Sie eine Schleife über den Einträgen im Array. Pro volle 100m geben Sie dann ein Leerzeichen aus (in einer geschachtelten Schleife), und am Ende der Zeile ein “`*`”.

```java
// Objektorientierte Programmierung - Hausaufgabe 5
// Name:

import java.util.Scanner;

class Mondlandung {
        public static void main(String[] args) {
                int zeit = 0;
                double entfernung = 4000;
                double geschwindigkeit = 0;
                int tank = 500;
                final double mondbeschleunigung = 1.62;
                int schub;
                final int dauer = 1; // Dauer eines Zuges in s
                Scanner scan = new Scanner(System.in);

                // Kopfzeile der Tabelle:
                System.out.format(
                        "%4s␣|␣%10s␣|␣%10s␣|␣%4s␣|␣%10s\n",
                        "Zeit", "Entfernung", "Geschwind.",
                        "Tank", "Bremsschub");

                // Schleife solange nicht gelandet:
```

```
        while(entfernung > 0) {
            // Maximal moeglichen Schub berechnen:
            int max = 10;
            if(tank < 10)
                max = tank;

            // Daten formatiert ausgeben:
            System.out.format(
                "%3ds␣|␣%9.1fm␣|␣%7.1fm/s" +
                    "␣|␣%4d␣|␣[0..%d]>␣",
                zeit, entfernung, geschwindigkeit,
                tank, max);

            // Eingabe Bremsschub (leere Eingabe = 0):
            String eingabe = scan.nextLine();
            if(eingabe.equals(""))
                schub = 0;
            else {
                schub = Integer.parseInt(eingabe);
                if(schub < 0)
                    schub = 0;
                if(schub > max)
                    schub = max;
            }

            // Neue Daten berechnen:
            double beschleunigung = mondbeschleunigung
                                    - schub * 0.4;
            entfernung = entfernung
                        - geschwindigkeit * dauer
                        - 0.5 * beschleunigung *
                            dauer * dauer;
            geschwindigkeit = geschwindigkeit +
                        beschleunigung * dauer;
            zeit = zeit + dauer;
            tank = tank - schub;
        }

        // Gelandet:
        System.out.format(
            "\nLandegeschwindigkeit:␣%7.1f␣m/s\n",
            geschwindigkeit);

        // Bewertung:

        // Graphik des Landeverlaufs ausgeben:
    }
}
```


Übungsaufgabe 5A: (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.

Im folgenden Programmausschnitt sind einige Fehler vorhanden. Finden Sie die Fehler und geben Sie an, wie diese behoben werden!

```
String text = "Test!";

System.out.println("Erstes␣Zeichen:␣" + text.charat(0));

System.out.println("Zweites␣Zeichen:␣" + text.charAt(2.0));

System.out.println("Drittes␣Zeichen:␣" + charAt(3);

System.out.println("Letztes␣Zeichen:␣" +
                        text.charAt(text.length()));

System.out.println("Es␣gibt␣" + text.length(3) +
                        "␣viele␣Zeichen.");

System.out.println("Text␣ist␣gleich␣\"Test!\":␣"
                        text.equals("Test!"));

System.out.println("Text␣kleingeschrieben␣ist␣gleich␣\"Test!\":␣"
                + text.toLowerCase().equals("Test!"));
```

Übungsaufgabe 5B: (ohne Abgabe)

Welche der folgenden Ausdrcke enthalten Typ-Fehler? Erklären Sie diese!
Deklarationen: `int i, j; boolean b; double d;`
(Initialisierungen sind bereits erfolgt.)

a) `i = (j==0)`

b) `b && i = j`

c) `d = i - j`

d) `b || i < 5`

e) `j = d - 2`

f) `j == 0 && d < 2.0`

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