Einführung in die Programmierung
Prof. Claudia Fohry

Universität Kassel
Wintersemester 2017/2018
Abgabe: 16.11.2017, 8:00 Uhr

Aufgabenblatt 4

Arbeiten Sie Kapitel 4 des Lehrbuchs durch. Beim Abschnitt über Schleifeninvarianten genügt informatives Lesen, alle übrigen Teile sollten Sie vollständig verstehen. Beachten Sie, dass schlechter Programmierstil zu einem Punktabzug von bis zu **30%** pro Aufgabe führen kann.

a) `Sequence.java` In Aufgabenblatt 1 hatten wir ein Ablaufdiagramm zur Berechnung des n-ten Wertes der Zahlenfolge

$$\begin{aligned} a_0 &= 2 \\ a_1 &= 2 \text{ und} \\ a_{i+1} &= 5 * a_{i-1} - 4 * a_i + 7 \end{aligned}$$

entwickelt. "Übersetzen" Sie das *in der Musterlösung* angegebene Ablaufdiagramm in ein Java-Programm. Überlegen Sie sich dazu, welcher Schleifentyp im Ablaufdiagramm verwendet wird und verwenden Sie den gleichen Typ auch im Java-Programm.

Zusätzlich zur Berechnung soll das Java-Programm den Wert für n vom Nutzer abfragen und das Ergebnis am Bildschirm ausgeben. Ausserdem soll es, ähnlich wie ein Schreibtischtest, die Werte der Programmvariablen vor Betreten der Schleife sowie am Ende jeder Iteration ausgeben. Beispiel für Programmablauf:

```
Eingabe: n = 6
i = 1 w = 2 v = 2 a = 2
i = 2 w = 2 v = 9 a = 9
i = 3 w = 9 v = -19 a = -19
i = 4 w = -19 v = 128 a = 128
i = 5 w = 128 v = -600 a = -600
i = 6 w = -600 v = 3047 a = 3047
Ausgabe: a = 3047
```

(6 Punkte)

b) `While.java`, `For.java`, `Do.java`: Schreiben Sie drei äquivalente Java-Programme, die jeweils für eine Zahl $n \geq 2$ testen, ob sie Primzahl ist. Die Zahl n soll vom Nutzer abgefragt werden. Für $n < 2$ soll eine Fehlermeldung am Bildschirm ausgegeben werden. Ansonsten sollen die Programme in einer Schleife die Zahlen $i = 2 \ldots n-1$ durchlaufen und jeweils überprüfen, ob n durch i teilbar ist. Sobald ein Teiler gefunden wurde, soll die Schleife abbrechen (d.h., die übrigen i sollen *nicht* mehr getestet werden). Verwenden Sie in den drei Programmen eine

- `while`-,
- `for`- bzw.
- `do-while`-Schleife

und benennen Sie die Programme entsprechend. Die Verwendung der `break`-Anweisung ist nicht erlaubt! Beispiele:

```
Bitte Zahl n >= 2 eingeben: 17
Ist Primzahl


Bitte Zahl n >= 2 eingeben: 18
Ist keine Primzahl


Bitte Zahl n >= 2 eingeben: 1
Unzulaessige Eingabe!
```

(12 Punkte)

c) `Short.java` Lassen sich Ihre Programme aus Aufgabe b) durch Verwendung der `break`-Anweisung verkürzen?

Wenn ja, schreiben Sie ein möglichst *kurzes* Programm zur Lösung von Aufgabe a). Entscheiden Sie sich für die am besten geeignete der drei Schleifenarten. Die Kürze darf nicht zu Lasten des Programmierstils gehen! Ausserdem dürfen nur die bereits behandelten Java-Konstrukte verwendet werden.

Wenn nein, begründen Sie, warum es nicht möglich ist.

(5 Punkte)

d) `Pattern1.java` Schreiben Sie ein Java-Programm, das vom Nutzer einen Wert n abfragt und dann am Bildschirm ein Muster aus Quadraten der Größen $1 \times 1, 2 \times 2, \ldots n \times n$ ausgibt. Die Quadrate sollen linksbündig untereinander stehen und mit $*$-Zeichen gefüllt sein. Es genügt, wenn Ihr Programm für $n \geq 1$ funktioniert. Beispiel:

```
Eingabe: n = 4
*
**
**
***
***
***
****
****
****
****
```

(10 Punkte)

e) `Pattern2.java` Ändern Sie Ihr Programm aus Aufgabe d) so ab, dass statt der Quadrate Dreiecke mit der Spitze nach unten dargestellt werden. Beispiel (soll genau so aussehen, aber für beliebige $n \geq 1$ funktionieren):

```
Eingabe: n = 4
*
**
*
***
**
*
****
***
**
*
```

(3 Punkte)

f) Welche der folgenden Schleifen terminieren? Begründen Sie jeweils kurz.

```
int a = 0;
int b = 5;
while (a != b) {
    a++;
    b--;
}
```

```
int sum = 0;
for (int i = 0; i < 1000; ++i) {
    sum = i;
    i--;
}
```

(4 Punkte)

Abgabetermin: Die Lösungen sind bis spätestens Donnerstag, den 16.11.2017 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.