

## Aufgabenblatt 6

Arbeiten Sie im Lehrbuch Kapitel 6 ab Abschnitt 6.4 sowie Kapitel 7 bis zur Überschrift „Freigabe von Arrayspeicher“ durch. Das Thema Verifikation mittels Schleifeninvariante können Sie überspringen, die übrigen Abschnitte sollten Sie vollständig verstehen. Lösen Sie danach die folgenden Aufgaben. Beachten Sie, dass schlechter Programmierstil zu einem Punktabzug von bis zu **40%** pro Aufgabe führen kann.

- a) Die folgende Java-Anweisung ordnet  $n$  eine Zufallszahl zwischen 1 und 6 zu:

```
int n = (int) (Math.random() * 6) + 1;
```

Dabei liefert der Aufruf `Math.random()` einen zufälligen `double`-Wert aus dem Intervall  $[0, 1)$ . Geben Sie an, wie  $n$  eine Zufallszahl zwischen 20 und 30 (*inklusive* der Werte 20 und 30) zugewiesen werden kann. Es ist nicht erlaubt, andere Bibliotheksfunktionen als `Math.random()` zu verwenden.

(3 Punkte)

- b) `Prefixsums.java` Schreiben Sie ein Java-Programm, welches in seiner `main`-Methode:

- einen `int`-Wert  $n > 0$  vom Nutzer abfragt,
- ein Array `a` der Länge  $n$  mit Zufallszahlen zwischen 1 und 6 initialisiert,
- das Array `a` am Bildschirm ausgibt,
- in einem Array `b` die Präfixsummen von `a` berechnet. Darunter versteht man Zahlen `b[i]` mit

$$b[i] = \sum_{k=0}^i a[k]$$

- und das Array `b` am Bildschirm ausgibt.

Da wir die Übergabe von Array-Parametern noch nicht behandelt haben, müssen Sie vorerst keine weiteren Methoden verwenden. Um die volle Punktzahl zu erreichen, beachten Sie bitte die folgenden Zusatzanforderungen:

- Die Berechnung der Präfixsummen soll möglichst effizient erfolgen, d.h., Ihr Programm soll zur Laufzeit nicht mehr Additionen als nötig ausführen. (2 der 13 Punkte)
- In der Ausgabe sollen ein- bzw. zweistellige Zahlen “richtig” untereinander geschrieben werden (wie im Beispiel unten). Hinweis: Dies können Sie durch eine geeignete Zahl von Leerzeichen erreichen. Sollte die Ausgabe nicht wie gewünscht aussehen, kann es an der Schriftart Ihrer Konsole liegen und Ihr Programm trotzdem korrekt sein. (2 der 13 Punkte)

Beispiel:

```
n = 10
a:  6   2   6   1   3   2   5   5   2   2
b:  6   8  14  15  18  20  25  30  32  34
```

(13 Punkte)

Die Aufgaben c) bis f) beziehen sich auf den folgenden Programmtext, welcher auch von der Moodle-Seite der Vorlesung geladen werden kann:

```
1 class Example {
2
3     static int n = 3;
4
5     public static void main(String[] args) {
6         double result = methA(n);
7         Out.print("n = ");
8         int n = In.readInt();
9         Out.println("result = " + result);
10    }
11
12    static double methA(int k) {
13        for (int m = 0; m < k; ++m) {
14            int i = m / 2;
15            n += i;
16        }
17        return methB(k);
18    }
19
20    static double methB(double d) {
21        return d * n;
22    }
23
24 }
```

- c) Warum berechnet das Programm immer das gleiche Ergebnis, auch wenn der Nutzer unterschiedliche Werte für n eingibt?

(1 Punkt)

- d) Geben Sie für die Variablen n aus Zeile 3, n aus Zeile 8, k, m und i Gültigkeitsbereich, Sichtbarkeitsbereich und Lebensdauer an. Achten Sie auf präzise Formulierungen.

(12 Punkte)

- e) Welche Nebeneffekte haben die Methoden des Programms?

(1 Punkt)

- f) Beseitigen Sie alle Nebeneffekte von Methoden (nicht die von Ausdrücken). Die Funktionalität des Programms darf nicht verändert und Methodenaufrufe dürfen nicht gestrichen werden.

(4 Punkte)

Die Aufgaben g) und h) beziehen sich auf den folgenden Programmtext, welcher auch von der Moodle-Seite der Vorlesung geladen werden kann:

```
1  class Convert {
2
3      public static void main(String[] args) {
4          double d1 = 5;
5          double d2 = 2 * meth(3 * d1);
6          int fract1 = meth(2.0, 3.0);
7          int fract2 = meth(2, 3);
8          Out.println(fract1 == fract2 ? "gleich" : "ungleich");
9      }
10
11     static float meth(int i) {
12         return 1.0 / i;
13     }
14
15     static double meth(double x, double y) {
16         return x / y + 0.5;
17     }
18
19     static int meth(int x, int y) {
20         return x / y + 0.5;
21     }
22
23 }
```

- g) Fügen Sie alle erforderlichen expliziten Typkonvertierungen ein, so dass das Programm fehlerfrei übersetzt und ausgeführt werden kann. Es dürfen nur die erforderlichen Konvertierungen eingefügt werden. Anderweitige Veränderungen des Programms sind unzulässig.

(4 Punkte)

- h) Das korrigierte Programm aus g) sollte die Ausschrift `ungleich` liefern. Begründen Sie, warum `fract1 ≠ fract2` gilt, obwohl in beiden Fällen  $2/3 + 0.5$  gerechnet wird. (Sollte Ihr Programm `gleich` liefern, erklären Sie stattdessen, wie die Berechnung von `fract1` und `fract2` durchgeführt wird.)

(2 Punkte)

**Abgabetermin:** Die Lösungen sind bis spätestens Donnerstag, den 30.11.2017 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.