

Aufgabenblatt 9

Lösen Sie die folgenden Aufgaben unter Beachtung der Anforderungen an den Programmierstil:

- a) Vergleichen Sie die Beispiellösungen zu Aufgabe 4a) und 8a) bezüglich ihrer Effizienz, d.h., des Verbrauchs an Rechenzeit. Fügen Sie dazu vor/nach der Berechnung Aufrufe der Methode `System.currentTimeMillis()` in den Programmtext ein, z.B.

```
...
double start = System.currentTimeMillis();
int result = a(n);
double end = System.currentTimeMillis();
Out.println("Benoetigte Zeit: " + (end - start) + " Millisekunden");
...
```

Messen Sie mit einem einigermaßen großen Wert von n (je nach Rechner z.B. $n=40$). Geben Sie an, welche Rekursionsform die Beispiellösung zu 8a) verwendet und erklären Sie den Unterschied in den gemessenen Laufzeiten.

(7 Punkte)

- b) Odd.java: Schreiben Sie ein Java-Programm, welches zu einer gegebenen natürlichen Zahl n ausgibt, ob sie ungerade ist. Benutzen Sie dazu verschränkte Rekursion und beziehen Sie sich auf die folgende Definition von gerade / ungerade:

- Die Zahl 0 ist gerade, also nicht ungerade.
- Eine Zahl ist gerade, wenn ihr Vorgänger ungerade ist.
- Eine Zahl ist ungerade, wenn ihr Vorgänger gerade ist.

(7 Punkte)

- c) Gegeben seien die folgenden Variablen- und Methodendeklarationen:

```
short s = 0; long g = 0;
static void meth(double d, long g) { Out.println("A"); }
static void meth(long g, float f) { Out.println("B"); }
static void meth(long g, int i) { Out.println("C"); }
```

Geben Sie für jeden der folgenden Methodenaufrufe an, ob er zulässig ist und welche Ausgabe er liefert. Für unzulässige Aufrufe ist die Antwort zu begründen.

- (1) `meth(1, 2.0f);`
- (2) `meth(g + 1, g + 2)`
- (3) `meth(3, 4);`
- (4) `meth(s, 1 < 2 ? 3.0f : 4.0);`

(6 Punkte)

d) Shorten.java: Gegeben sind Muster

- A) 1101
- B) 0011
- C) 110
- D) 0

sowie eine Zeichenreihe `str` aus 0en und 1en. Streicht man aus `str` wiederholt Vorkommen der Muster A bis D bis keines mehr enthalten ist, so bleibt eine Zeichenreihe aus 1en übrig.

Je nachdem, welche Muster man in welcher Reihenfolge von welchen Positionen der Zeichenreihe streicht, erhält man unterschiedliche Ergebnisse, z.B.

$$\begin{aligned} 1100110 &\xrightarrow{D} 110110 \xrightarrow{A} 10 \xrightarrow{D} 1 \\ 1100110 &\xrightarrow{B} 110 \xrightarrow{C} \varepsilon \\ 1100110 &\xrightarrow{C} 1100 \xrightarrow{D,D} 11 \\ 1100110 &\xrightarrow{D,D,D} 1111 \end{aligned}$$

Schreiben Sie ein Java-Programm, welches die Zeichenreihe `str` als Kommandozeilenparameter übergeben bekommt und alle Ersetzungsmöglichkeiten am Bildschirm ausgibt. Das Programm **muss** Rekursion verwenden. Hinweis: Für das Ermitteln des Ersetzungsweges können Sie einen Akkumulator vom Typ `String` benutzen. Eventuell ist es einfacher, die Aufgabe zunächst ohne diesen Akkumulator zu lösen und nur die Endergebnisse (also nur die übrig bleibenden Zeichenreihen aus 1en) auszugeben. Auf diese Teillösung werden bis zu 15 Punkte vergeben. Sobald Debugging notwendig wird, kann der Einbau des Akkumulators allerdings sinnvoll sein. Beispiel:

```
java Shorten 1100110
1100110 -> 110 -> leer
1100110 -> 110 -> 11
1100110 -> 0110 -> 0 -> leer
1100110 -> 0110 -> 110 -> leer
1100110 -> 0110 -> 110 -> 11
1100110 -> 0110 -> 011 -> 11
1100110 -> 1100 -> 0 -> leer
1100110 -> 1100 -> 110 -> leer
1100110 -> 1100 -> 110 -> 11
1100110 -> 1100 -> 110 -> leer
1100110 -> 1100 -> 110 -> 11
1100110 -> 110110 -> 10 -> 1
1100110 -> 110110 -> 110 -> leer
1100110 -> 110110 -> 110 -> 11
1100110 -> 110110 -> 110 -> leer
1100110 -> 110110 -> 110 -> 11
1100110 -> 110110 -> 11110 -> 11
1100110 -> 110110 -> 11110 -> 1111
...
```

(20 Punkte)

Abgabetermin: Die Lösungen sind bis spätestens Donnerstag, den 21.12.2017 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.