

## Aufgabenblatt 5

Arbeiten Sie Kapitel 5 sowie Kapitel 6 bis einschließlich Abschnitt 6.3 des Lehrbuches durch und lösen Sie danach die folgenden Aufgaben. Dabei dürfen Sie nur die bisher behandelten Sprachkonstrukte von Java verwenden. Beachten Sie, dass schlechter Programmierstil zu einem Punktabzug von bis zu **40%** pro Aufgabe führen kann.

a) Welche der folgenden Zuweisungen sind in Java zulässig? Begründen Sie Ihre Antworten.

- (1) `float f1 = 3E4;`
- (2) `float f2 = 1 / 2f;`
- (3) `double d1 = 1f;`
- (4) `double d2 = 1.0 / 0.0;`

(4 Punkte)

b) Gegeben sei das nachstehende Java-Programm. Welche Ausgaben werden durch die Zeilen 7 bis 14 erzeugt? Erklären Sie, wie diese Ausgaben zustande kommen. Gehen Sie dabei insbesondere auf die Werte der Variablen `g` und `d` ein. Geben Sie dem Phänomen den Fachnamen nach Lehrbuch.

```
1 public class DoubleLongMystery {
2
3     public static void main(String[] args) {
4         long g = 10_000_000_000_000_001L;
5         double d = g;
6         Out.println("--- Initializing g and d -----");
7         Out.println("g          = " + g);
8         Out.println("d          = " + d);
9
10        if (d == g) {
11            Out.println("d == g ?          yes");
12        } else /* if (d != g) */ {
13            Out.println("d == g ?          no");
14        }
15
16        Out.println("--- double newD = d - 1L -----");
17        double newD = d - 1L;
18        Out.println("d          = " + d);
19        Out.println("newD = d - 1L = " + newD);
20
21        if (d == newD) {
```

```

22         Out.println("d == newD ?      yes");
23     } else /* if (d != newD) */ {
24         Out.println("d == newD ?      no");
25     }
26 }
27
28 }

```

(4 Punkte)

- c) Welche Ausgaben werden durch die Zeilen 18 bis 25 des in Aufgabe b) gegebenen Programms erzeugt? Erklären Sie, wie die Ausgaben zustande kommen. Gehen Sie dabei insbesondere auf die Werte der Variablen `d` und `newD` ein. Geben Sie dem Phänomen den Fachnamen nach Lehrbuch.

(4 Punkte)

- d) `DoubleLongMystery.java`: Schreiben Sie das Programm aus Aufgabe b) so um, dass anstatt der `if`-Anweisungen der Fragezeichenoperator verwendet wird.

(4 Punkte)

- e) `Max.java`: Bei der Übersetzung des nachstehenden Java-Programms meldet der Compiler den folgenden Fehler.

```

Max.java:17: error: missing return statement
    }
    ^

```

Beheben Sie den Fehler so, dass die Funktionalität des Programms erhalten bleibt.

```

1  public class Max {
2
3      public static void main (String[] args) {
4          Out.print("x = ");
5          int x = In.readInt();
6          Out.print("y = ");
7          int y = In.readInt();
8          Out.println("max(x, y) = " + max(x, y));
9      }
10
11     public static int max(int x, int y) {
12         if (x >= y) {
13             return x;
14         } else if (x < y) {
15             return y;
16         }
17     }
18
19 }

```

(2 Punkte)

f) Benennen Sie aus dem Quellcode von Aufgabe e) je ein Beispiel für

- einen aktuellen Parameter,
- einen formalen Parameter,
- einen Methodenkopf und
- einen Methodenrumpf.

(4 Punkte)

g) Login.java: Ziel dieser Aufgabe ist es, dass Sie ein rudimentäres Loginsystem zur Benutzerauthentifizierung schreiben. Ein Benutzer authentifiziert sich hierbei durch eine Benutzer-ID sowie einen Zugangscode. Benutzer-ID und Zugangscode werden in int-Variablen abgespeichert. Hierzu ist der folgende Quellcode bereits vorgegeben.

```
1 public class Login {
2
3     public static final int CORRECT_USER_ID = 42;
4     public static final int CORRECT_ACCESS_CODE = 1307;
5
6     public static void main(String[] args) {
7     }
8
9     public static void login() {
10         int userId;
11         int accessCode;
12         // do {
13             greetings();
14             // TODO: read values for userId and accessCode from console
15             // } while (!isCorrectLogin(userId, accessCode));
16
17         Out.println ("Authentication successful.");
18     }
19
20 }
```

1. Rufen Sie die Methode login() in der main(...) -Methode auf.

(2 Punkte)

2. Wie Sie sehen, fehlt die Methode greetings(). Schreiben Sie diese Methode und geben Sie eine Begrüßung an den Benutzer auf der Konsole aus. Nachfolgend finden Sie einen Beispielablauf des Programms nach korrekter Implementierung dieser Methode. Sie dürfen ihren Begrüßungstext gerne individuell gestalten.

```
$> java Login

#####
# Welcome to the Securotron 9000 #
#####
Authentication successful.
```

(6 Punkte)

3. Kommentieren Sie die Zeilen 12 und 15 ein und implementieren Sie die Methode `isCorrectLogin(...)`. Diese Methode soll `true` zurückliefern, wenn die übergebene Benutzer-ID gleich `CORRECT_USER_ID` und der übergebene Zugangscode gleich `CORRECT_ACCESS_CODE` ist und sonst `false`. Lesen Sie zusätzlich innerhalb der `do-while` Schleife der Methode `login` die Benutzer-ID (`userId`) und den Zugangscode (`accessCode`) mit Hilfe der Klasse `In` von der Konsole ein. Nachfolgend finden Sie einen Beispielablauf des Programms nach korrekter Implementierung dieser Methode.

```
$> java Login

#####
# Welcome to the Securotron 9000 #
#####
Please enter your user id: 7
Please enter your access code: 7

#####
# Welcome to the Securotron 9000 #
#####
Please enter your user id: 7
Please enter your access code: 1307

#####
# Welcome to the Securotron 9000 #
#####
Please enter your user id: 42
Please enter your access code: 7

#####
# Welcome to the Securotron 9000 #
#####
Please enter your user id: 42
Please enter your access code: 1307
Authentification successful.
```

(10 Punkte)

**Abgabetermin:** Die Lösungen sind bis spätestens Donnerstag, den 23.11.2017 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.