

Übungsaufgaben EIDI 2

Version 1.1.3

Christian Femers

9. April 2019

ACHTUNG: LÖSUNGEN

1. Was sind **keine** Java-Schlüsselwörter?

- ☐ final
- ☐ const
- ☒ var
- ☐ short
- ☒ false
- ☐ case
- ☐ class
- ☒ main
- ☐ static
- ☐ throw
- ☐ throws
- ☒ null

2. Zu was evaluieren die folgenden Java-Ausdrücke?

- (a) `3 - 0`: 3
- (b) `-7 / 2`: -3
- (c) `-7 / 2.0`: -3.5
- (d) `-7d / 2`: -3.5
- (e) `true ? 3 : 2 + 1`: 3
- (f) `0.25 * 8`: 2.0
- (g) `42 * 2 + "Niugnip" + 42 + 2`: "84Niugnip422"
- (h) `(long) 1.0 + "java" + 1`: "1java1"
- (i) `4 + ~-1 >= 5 ? 2 * 0 : 3 / 2 + 2`: 3
- (j) `'a' + 25 - 'z'`: 0
- (k) `(8745 / 61) + (83 / 0)`: java.lang.ArithmeticException: / by zero
- (l) `0 / 1 == 0 ? null : "hallo"`: null
- (m) `(byte) 127 + 1`: 128

3. Betrachten Sie den folgenden Code-Auszug:

```
1 Integer input = getUserInput();
2
3 if (input == (Integer) 42)
4     System.out.println("Antwort gefunden");
5 else
6     System.out.println("Weitersuchen");
7
8 if ("83".equals("" + (int) input))
9     System.out.println("83 gefunden");
```

Welche der folgenden Aussagen treffen zu? Nehmen Sie an, dass der Code kompiliert und betrachten sie ihn als Algorithmus. Gehen Sie nur von dem aus, was sie sehen können.

- ☐ In Zeile 3 wird der Wert von `input` mit 42 verglichen.
- ☒ In Zeile 3 werden Objekte auf Referenzgleichheit geprüft
- ☒ Antwort gefunden wird möglicherweise für eine Eingabe von 42 ausgegeben.
- ☐ Für die Eingabe 42 wird nie Antwort gefunden ausgegeben werden.
- ☒ Möglicherweise wird Weitersuchen für eine Eingabe von 42 ausgegeben.
- ☐ Möglicherweise wird Antwort gefunden und Weitersuchen für eine Eingabe von 42 ausgegeben.
- ☒ Bei der Eingabe von 83 wird immer Weitersuchen und 83 gefunden ausgegeben.
- ☐ 83 gefunden wird nie ausgegeben werden.
- ☐ In Zeile 8 wird auf Referenzgleichheit geprüft.
- ☒ In Zeile 8 werden zwei Strings zeichenweise miteinander verglichen.
- ☐ Für die Eingabe -83 wird 83 gefunden ausgegeben.
- ☒ Bei Zeile 3 wird nie eine `NullPointerException` geworfen werden.
- ☒ Zeile 8 wirft möglicherweise eine `NullPointerException`.
- ☐ Bei Zeile 8 wirft `equals` eine `IllegalArgumentException`.
- ☐ Die Ausgabe ist nicht deterministisch.
- ☒ Möglicherweise wird gar nichts in die Konsole ausgegeben.

4. Zu welchem Wert evaluieren die folgenden Ausdrücke, vorausgesetzt die `int`-Variable `x` hat vor jeder Teilaufgabe den Wert 0?

- (a) `x++`: 0
- (b) `x = x = x++`: 0
- (c) `x++ + ++x`: 2
- (d) `--x != ++x`: true
- (e) `--x - --x - x`: 3

(f) `x++ == x++ ? x-- - 1 : --x + 1`: 2

(g) `x++ * x++ * x++`: 0

5. Was sind erlaubte Bezeichner für Variablen in Java (ab Version 9)?

☐ ein name

☐ const

☒ var

☒ `uPC`

☒ `$name`

☐ `_`

☒ `--`

☐ public

☒ CLASS

☐ 42sinn

☒ main

☒ `_mäin_`

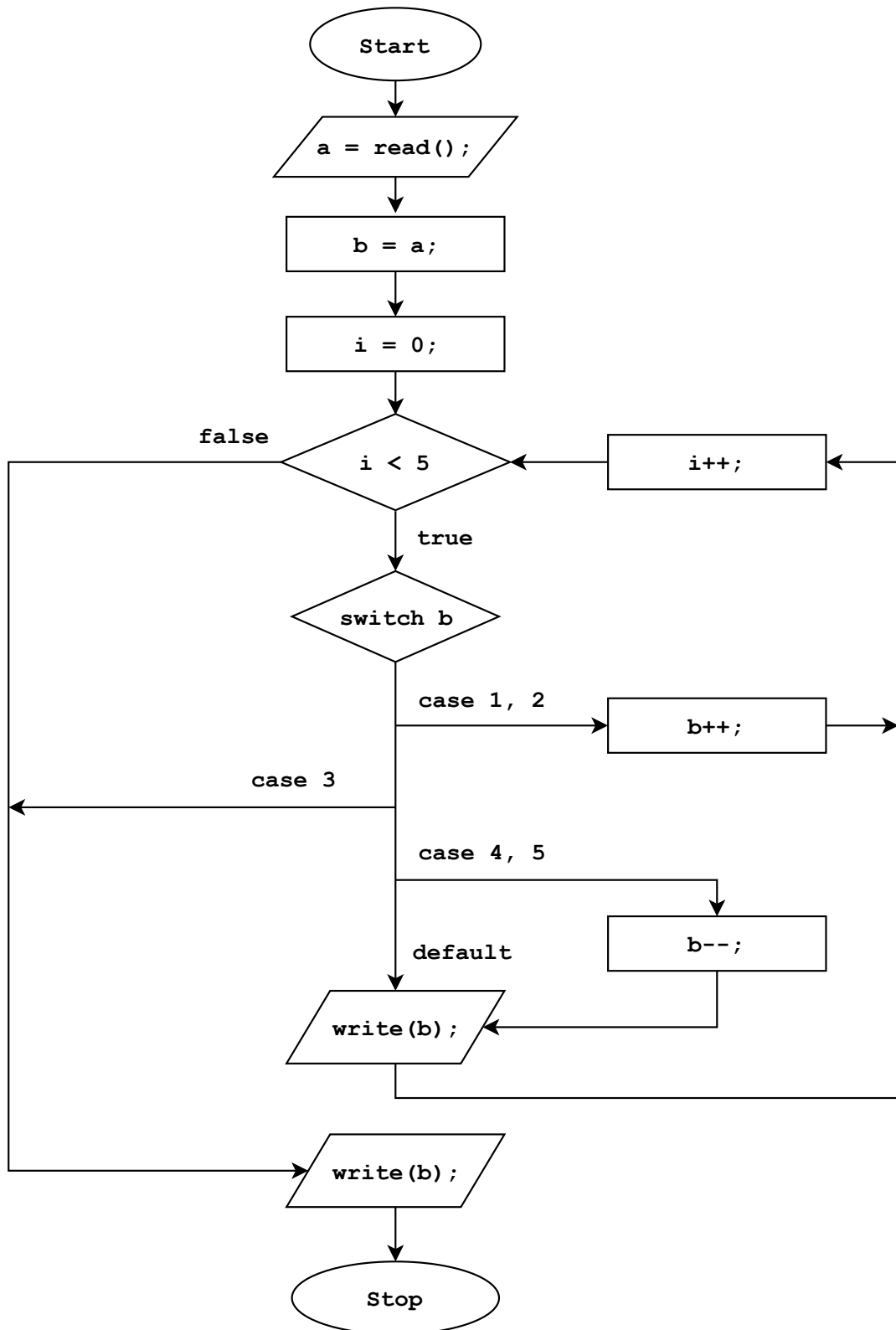
☐ pingu!n

☐ null

6. Erstellen sie das Kontrollflussdiagramm zu folgendem Java-Code-Auszug. Die Methode `read()` gibt dabei einen `int` zurück, `write(int)` gibt den übergebenen `int`-Wert auf der Konsole aus.

```
1  final int a = read();
2  int b = a;
3  OUTER: for(int i = 0; i < 5; i++) {
4      switch(b) {
5          case 1:
6              case 2: b++; continue;
7              case 5:
8                  case 4: b--; break;
9                  case 3: break OUTER;
10     }
11     write(b);
12 }
13 write(b);
```

Lösung:



Es gibt verschiedene richtige Wege, **switch** darzustellen. Je nachdem ist es sicherer, die **case** 1, 2 aufzuspalten in zwei Pfeile **case** 1 und **case** 2. Es müsste jedoch beides als richtig angesehen und gewertet werden.