

# Einführung in die Programmierung mit Java – Wiederholung

---

Dozent: Prof. Dr. Michael Eichberg  
Kontakt: [michael.eichberg@dhbw.de](mailto:michael.eichberg@dhbw.de), Raum 149B  
Version: 1.1

---

# Kontrollfragen

0.1. Welche primitiven Datentypen kennen wir?

1

0.2. Was sind Literale?

2

0.3. Welche der folgenden Bezeichner sind für Variablen gültig?

3

1. `fooBar`
2. `BarFoo`
3. `_fooBar`
4. `1fooBar`
5. `fooBar1`
6. `fooBar!`
7. `$fooBar`
8. `$_BarFoo`

0.4. Welche der folgenden Bezeichner sollte man verwenden?

4

1. `gewinn`
2. `Gewinn`
3. `_private_i`
4. `i`
5. `$i`
6. `_i`

0.5. Welchen Namen würden Sie für eine Konstante verwenden?

5

1. `ISOLAENDERCODE`
2. `ISO_LÄNDERCODE`
3. `ISO_LAENDERCODE`
4. `ISO_Ländercode`
5. `ISO_Laendercode`
6. `iso_Laendercode`

0.6. Welchen Typ hat die Variable x in folgendem Code?

6

1. `var x = 1;`
2. `var x = 1.0;`
3. `var x = '1';`
4. `var x = 1f;`
5. `var x = 2F;`
6. `var x = "x";`

0.7. Wie viele Bits hat ein int?

7

1. 8
2. 16
3. 24
4. 32
5. 40
6. 48

---

0.8. Wie ist der Wertebereich von byte?

8

1. 0 bis 255
2. -128 bis 128
3. -128 bis 127
4. -127 bis 127
5. -127 bis 128

---

0.9. Was passiert bei den folgenden Typumwandlungen?

9

1. `int i = 42; byte b = (byte) i;`
2. `int i = 255; byte b = (byte) i;`
3. `int i = 256; byte b = (byte) i;`

---

0.10. Warum ist der folgende Ausdruck wahr?

10

`(long) ((float) (Long.MAX_VALUE - Integer.MAX_VALUE)) == Long.MAX_VALUE;`

Bemerkung

Rein mathematisch betrachtet - d. h. ohne Betrachtung von Typen und Typumwandlungen - wäre dieser natürlich falsch.

---

0.11. Ist die Länge eines Strings gleich der Anzahl *sichtbarer* Zeichen?

11

---

0.12. Wie fügen Sie in einen String ein Anführungszeichen ein?

12

---

0.13. Muss ich bei der Variablendeklaration den Typ explizit angeben?

13

---

0.14. Wie deklariert man eine Konstante?

14

Sollte man Werte, die man nicht ändern möchte immer als Konstanten deklarieren?

---

0.15. Wie ist der Operator für die Modulorechnung in Java?

15

(D. h. wenn Sie eine Restwertberechnung in Java durchführen wollen.)

---

0.16. Wie sieht der ternäre Operator in Java aus?

16

---

0.17. Welchen Wert haben die folgenden Ausdrücke?

17

`x` hat vor der jeweiligen Auswertung den Wert 5.

1. `x++`

2. `++x`
3. `x += 1`
4. `x = (x = x - 2) + 3 * 4`
5. `x = x = x - 2 + 3 * 4`
6. `(x = (x = x - 2) + 3) * 4`
7. `x >= 5 || 2 / (x - 5) == 0`
8. `x >= 5 | 2 / (x - 5) == 0`
9. `x << 1 >> 2`

---

0.18. Was stellt ein Block in Hinblick auf eine Variable dar?

18

---

0.19. Können `while`- und `for`-Schleifen ineinander umgewandelt werden?

19

---

0.20. Wie unterscheidet sich eine `do-while`- von einer `while`-Schleife?

20

---

0.21. Schleifen und Variablen - wie ist die Ausgabe auf der *JShell*?

21

```
1 int i = 0;
2 for (int i = 0; i < 10; i++) {
3     if (i == 5) {
4         break;
5     }
6 }
7 System.out.println(i);
```

---

0.22. Schleife mit `break` - wie ist die Ausgabe?

22

```
1 int i = 0;
2 for (; i < 10; i++) {
3     if (i == 5) {
4         break;
5     }
6 }
7 System.out.println(i);
```

---

0.23. Ganz einfache Schleife - wie ist die Ausgabe?

23

```
1 int i = 10;
2 for (; i < 10; i++) {
3     System.out.println(i);
4 }
```

---

0.24. Schleife mit `continue` - wie ist die Ausgabe?

24

```
1 int i = 0;
2 for (; i < 10; i++) {
3     if (i % 2 == 0) {
4         continue;
5     }
6     System.out.println(i);
7 }
```

---

0.25. Verschachtelt Schleifen - wie ist die Ausgabe?

25

```

1 int i = 0;
2 outer : for (; i < 10; i++) {
3     if (i % 2 == 0)
4         continue;
5     System.out.println(i);
6     for (int j = 1; j < 10; j++) {
7         if (j % 3 == 0)
8             continue outer;
9         System.out.println(i + " " + j);
10    }
11 }
12 System.out.println(i);

```

## 0.26. Verschachtelt Schleifen - wie ist die Ausgabe?

26

```

1 outer : for (int i = 0; i < 10; i++) {
2     if (i % 2 == 0) {
3         i = 10;
4         continue outer;
5     }
6     System.out.println(i);
7     for (int j = 1; j < 10; j++) {
8         if (j % 3 == i % 5)
9             break;
10        System.out.println(i + " " + j);
11    } }

```

## 0.27. Rekursive Funktion

27

```

1 int f(int n) {
2     if (n == 0) return 0; return n + f(n-1);
3 }

```

- Was berechnet diese Funktion?
- Ist diese Funktion effizient?
- Ist eine Lösung mit for-Schleife besser?

## 0.28. Funktion mit „Tail-Call“

28

```

1 /* private */ int f(int n, int sum) {
2     if (n == 0) return sum; return f(n-1, n+sum);
3 }
4 int f(int n) { return f(n, 0); }

```

- Was berechnet diese Funktion?
- Ist diese Funktion effizient(er)?

## 0.29. Wie werden Parameter übergeben?

29

## 0.30. Wie bewerten Sie folgende Kommentierung?

30

```

1 /**
2  * Testet ob eine Zahl eine Primzahl ist.
3  *
4  * Die Laufzeit ist O(n/4).
5  *
6  * @param n Eine positive ganze Zahl.

```

```

7  * @return true, wenn n eine Primzahl ist, sonst false.
8  */
9  boolean isPrim(int n) {
10     ...
11 }

```

31

### 0.31. Ist der Kommentar ausreichend?

```

1  /**
2   * Computes the absolute value of the argument.
3   *
4   * @param a - the argument whose absolute value is to be
5   *           determined
6   * @return the absolute value of the argument.
7   */
8  double abs(double a) { ... }

```

32

### 0.32. Ist die Kommentierung hier ausreichend?

```

1  /**
2   * Returns the absolute value of an int value.
3   * If the argument is not negative, the argument is returned.
4   * If the argument is negative, the negation of the argument
5   * is returned.
6   *
7   * @param a - the argument whose absolute value is to be
8   *           determined
9   * @return the absolute value of the argument.
10  */
11 long abs(long a) { ... }

```

33

### 0.33. Sind Java Assertions (`assert`) in Java immer aktiv?

34

### 0.34. Wofür sollten Assertions verwendet werden?

1. Zur Validierung von Eingabeparametern?
2. Zur Validierung von Rückgabewerten?
3. Zur Validierung von internen Invarianten?

35

### 0.35. Beschreiben Sie die Ausgabe des Programms

```

1  int width = 20;
2  int height = 10;
3  for (int i = 0; i < width; i++) print("-");
4  println("");
5  for (int i = 0; i < height - 2; i++) {
6      print("|");
7      for (int j = 0; j < width - 2; j++) print(" ");
8      println("|");
9  }
10 for (int i = 0; i < width; i++) {
11     print("-");
12 }
13 println("");

```