

## Selbsttest 5

### 1. Umwandlung von Schleifen

Geben Sie die Ausgabe bei der Ausführung der folgenden Code-Stücke an. Geben Sie eine Assertion nach der Schleife an und wandeln Sie anschließend die Schleife in eine do-while- und in eine for-Schleife um, die dieselben Werte berechnen.

```
int i, j, k;

i = 0;
j = 10;
k = 0;
while (i++ != j) {
    k = i + j;
    if (i % 3 == 0)
        j--;
    out.write("i: " + i + ", j: " + j + ", k: " + k);
}
// ???
out.write("i: " + i + ", j: " + j + ", k: " + k);
```

### 2. Wissensfragen – Parameter

- 2.1 Formale Parameter stehen \_\_\_\_\_ (im Methodenkopf / beim Aufruf) .
- 2.2 Formale Parameter können \_\_\_\_\_ (nicht / schon) Ausdrücke sein.
- 2.3 Formale Parameter müssen an aktuelle Parameter \_\_\_\_\_ (nicht / schon) zuweisbar sein.
- 2.4 Aktuelle Parameter werden \_\_\_\_\_ (nach / vor) dem Aufruf ausgewertet.

### 3. Wissensfragen – lokale und globale Variablen

- 3.1 Lokale Variablen werden \_\_\_\_\_ (nach dem Methodenaufruf / am Programmende) wieder freigegeben.
- 3.2 Globale Variablen stehen \_\_\_\_\_ (nach / vor / während) einem Methodenaufruf zur Verfügung (wenn man mögliche Überdeckung unbeachtet lässt).
- 3.3 Lokale Variablen stehen \_\_\_\_\_ (nach / vor / während) eines Methodenaufrufs zur Verfügung.
- 3.4 Auf lokale Variablen der Methode X kann die Methode Y nur zugreifen, wenn die Methode X gerade aktiv ist. (richtig / falsch)

### 4. Wissensfragen – Definitionen

- 4.1 Der Sichtbarkeitsbereich einer Variable ist \_\_\_\_\_.
- 4.2 Der Gültigkeitsbereich einer Variable ist \_\_\_\_\_.
- 4.3 Die Lebensdauer einer Variable ist \_\_\_\_\_.
- 4.4 Ein Name überdeckt einen anderen, wenn \_\_\_\_\_.