

Übungsblatt 3: Erste Programme und Zustand

Deadline: 30.10.2025 12:00 Uhr

Besprechung: 03.11.–07.11.2025

Hinweise für die Bearbeitung der Aufgaben

- Bitte lesen Sie die Aufgabenstellungen genau durch.
- Testen Sie Ihre Lösungen vor der Abgabe selbst!
- Halten Sie sich an die Java Coding Konventionen (siehe letzte Seite).
- Damit die automatischen Tests korrekte Ergebnisse liefern können, achten Sie bitte auf die folgenden Punkte:
 - Halten Sie sich bitte genau an die Vorgaben/Vorlagen aus der Aufgabenstellung.
 - Wenn Sie Umlaute verwenden, stellen Sie die Kodierung Ihrer Dateien unbedingt auf UTF-8.
 - Achten Sie darauf, dass Ihr Projekt mit Java Version 21 kompiliert und ausgeführt werden kann.
 - Ihre Java-Klassen müssen sich im default package befinden (es darf kein package angegeben werden).

Aufgabe 1 Abgabe mit automatischen Tests in GATE
for-Schleife

Schreiben Sie eine Java-Anwendung mit dem Namen `SimpleForLoop` welche in einer FOR-Schleife die Zahlen von 10 bis 1 herunter zählt (je eine Zahl pro Zeile) und danach „START!“ ausgibt (ohne Anführungszeichen).

```
public class SimpleForLoop {  
    public static void main(String[] args) {  
        // Tragen Sie bitte hier Ihre Loesung ein  
    }  
}
```

Aufgabe 2 Abgabe mit automatischen Tests in GATE
while-Schleife

Schreiben Sie eine Java-Anwendung mit dem Namen `SimpleWhileLoop` welche in einer WHILE-Schleife alle ungeraden Zahlen von 1 bis 10 hochzählt und (je eine Zahl pro Zeile) ausgibt und danach „Boom!“ ausgibt (ohne Anführungszeichen).

Aufgabe 3 Abgabe in GATE mit Peer-Review
Programmmzustände

In dieser Aufgabe sollen die Werte (Zustände) der lokalen Variablen während der Ausführung eines Programms betrachten werden. Gegeben sei das folgende Java-Programm:

Listing 1: Zustand.java

```

1 public class Zustand {
2     public static void main(String args[]) {
3         int m = 3;
4         double d = 2.2;
5         d = m++ * d;
6         int n = 15 * (int)d;
7         if (d == m || true) {
8             int k = 8;
9             d = d + k;
10        }
11        d = n / 7;
12        System.out.println("Ergebnis: " + d);
13    }
14 }

```

Geben Sie die Zustände des Programms nach Ausführung der einzelnen Anweisungen (identifiziert durch ihre Zeilennummer) als Liste von Variablenname/Wert-Paaren an. Beispiel: die Liste $[(x, 2), (y, \text{true})]$ zeigt einen Programmzustand an, in dem die Variable x (vom Typ `int`) den Wert 2 hat und y (vom Typ `boolean`) den Wert `true` hat.

Ein Beispiel für das Programm BeispielZustand:

Listing 2: BeispielZustand.java

```

1 public class BeispielZustand {
2     public static void main(String args[]) {
3         int i = 4;
4         int j = 2 * i;
5         i = i*i;
6         System.out.println(i);
7     }
8 }

```

Hat die folgenden Zustände:

- 3: $[(i, 4)]$
- 4: $[(i, 4), (j, 8)]$
- 5: $[(i, 16), (j, 8)]$
- 6: $[(i, 16), (j, 8)]$

Aufgabe 4 Abgabe in GATE mit automatischen Tests

Quersumme berechnen

Laden Sie die Datei `CrossSum.java` von GATE herunter und ergänzen sie den Code an gegebener Stelle, sodass die Quersumme der natürlichen Zahl n berechnet wird. Testen Sie anschließend ihr Programm, indem sie an der angegebenen Stelle verschiedene Zahlen ausprobieren! Die hochzuladende Datei muss "CrossSum.java" heißen und sich im default-Package befinden.

```
public class CrossSum {  
  
    public static int calculateCrossSum (int n) {  
        int result = 0;  
  
        // ergaenzen Sie das Programm so, dass result die  
        // Quersumme von n berechnet  
  
        return result;  
    }  
  
    public static void main(String[] args) {  
        // Testen Sie Ihr Programm hier  
        System.out.println(calculateCrossSum(125));  
    }  
}
```

Aufgabe 5 Fehler kennenlernen (keine Abgabe)

Bei dieser Aufgabe geht es darum, Fehler zu machen und mit Fehlermeldungen des Compilers vertraut zu werden. Was passiert, wenn Sie in der Datei `HelloWorld.java` z.B.

1. eine Semikolon entfernen?
2. ein doppeltes Anführungszeichen entfernen?
3. statt "System" "Sytem" schreiben?
4. "class" entfernen?
5. eine geschweifte Klammer entfernen?
6. eine runde Klammer entfernen?

Abgabe

Die Abgabe erfolgt bis zum 30.10.2025 12:00 Uhr auf GATE, <https://gate.ifi.lmu.de>.

Zuletzt aktualisiert: 21. Oktober 2025

Java Coding Konventionen

Zur Erstellung eines lesbaren und wartbaren Codes sind Coding Conventions sehr hilfreich. Im Folgenden sind einige der wichtigsten Konventionen für Java aufgeführt:

- Achten Sie auf gute Lesbarkeit Ihrer Programme und rücken Sie Anweisungen immer entsprechend ein (üblich sind 2 oder 4 Leerzeichen bzw. ein Tabulator pro Ebene).
- Kommentare sollten verwendet werden, um den Code zu erklären an den Stellen, an denen Ihre Algorithmus-Idee nicht offensichtlich ist und nicht um ihn zu wiederholen.
- Leerzeichen sollten verwendet werden, um den Code lesbarer zu machen (z. B. nach Kommas, um Operatoren herum). Nutzen Sie diese auch vor und nach Klammern (z. B. `if (condition) { ... }`).
- Java ist eine Case-sensitive Sprache. `Variable`, `variable` und `VARIABLE` sind drei verschiedene Bezeichner.
- Mehrteilige Bezeichner werden im camelCase-Stil geschrieben, z. B. `firstName`, `calculateSum`.
- Konstanten werden in Großbuchstaben mit Unterstrichen geschrieben, z. B. `MAX_VALUE`.
- Verwenden Sie verständliche bzw. sprechende Bezeichner für Variablen und Methoden.
- Verwenden Sie geschweifte Klammern auch für einzeilige Blöcke.

Weitere Informationen finden Sie in den offiziellen Java Coding Conventions: <https://www.oracle.com/java/technologies/javase/codeconventions-contents.html>