

## Übungsblatt 4: Kontrollstrukturen und Arrays

Deadline: 06.11.2025 12:00 Uhr

Besprechung: 10.11.–14.11.2025

### 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 in GATE Fehler finden

Finden Sie in dem folgenden Programmcode, welches der Inhalt einer Datei namens `FehlerSuche.java` sein soll, mindestens 5 der 8 Fehler!

Beschreiben Sie jeden Fehler eindeutig.

```
10 public class programm_fehlerhaft
20 { public static main(String[] args)
30 int i = 7;
40 if ((i = 25) || (i > 6 && < 10)) {
50 /* Hier findet gleich eine Ausgabe statt */
60 system.out.print(Hello World!)
70 /* Hier ist die Ausgabe schon
80 abgeschlossen */
90 }
100 }
```

Hinweise: Die Zeilennummern gehören nicht zum Inhalt der Datei, sondern sollen Ihnen lediglich bei der Fehlerbeschreibung behilfreich sein. Es wird empfohlen keine Entwicklungsumgebung (Eclipse, NetBeans, IntelliJ, . . .) zu verwenden, da Ihnen eine solche in der Klausur nicht zur Verfügung stehen wird, sondern nur Stift und Papier.

### Aufgabe 2      Abgabe in GATE mit automatischen Tests Verschachtelte Schleifen

Benutzen Sie die hinterlegte Datei `TriangleOutput.java` als Vorlage und schreiben Sie an die vorgegebene Stelle ein Programm, das ein Dreieck beliebiger Größe ausgibt. Hierbei soll der Anwender nach der Größe gefragt werden – dies haben wir für Sie in der Vorlage `TriangleOutput.java` bereits vorbereitet. Verwenden Sie zur Ausgabe des Dreiecks ausschließlich die folgenden zwei Befehle:

```
System.out.print("*");
System.out.println();
```

Außerdem sollen Sie zwei ineinander geschachtelte **for**-Schleifen verwenden.

Die Ausgaben Ihres Programms sollten z.B. für die Eingaben 8 und 3 folgendermaßen aussehen:

In der Programmcode-Vorlage sehen Sie zudem, wie Integer-Werte mithilfe eines sog. Scanners von der Konsole gelesen werden können.

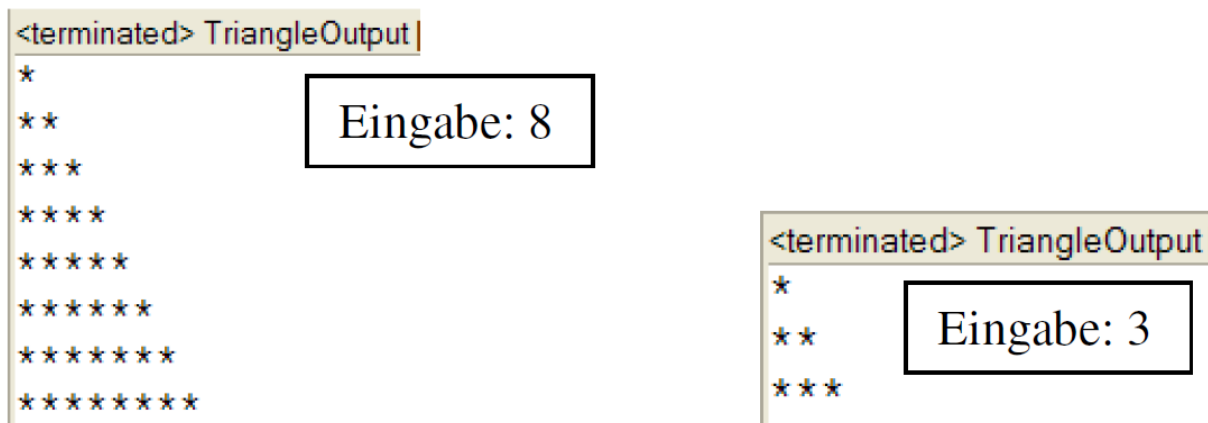


Abbildung 1: Beispielausgabe

Listing 1: TriangleOutput.java

```

1 import java.util.Scanner; /* notwendig, damit der Scanner
   funktioniert */
2
3 public class TriangleOutput {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in); /* oeffnet die
   Konsole */
6         System.out.println("Bitte geben Sie die Hoehe des Dreiecks ein:
   ");
7         int triangleSize = scanner.nextInt(); /* liest einen Integer
   von der Konsole */
8         drawTriangle(triangleSize);
9         scanner.close(); /* schliesst die Konsole */
10    }
11
12    public static void drawTriangle(int sizeOfTriangle) {
13        // Bitte hier Loesung einfuegen.
14    }
15 }

```

### Aufgabe 3 Abgabe über GATE mit Peer-Review Buchstaben zählen

Schreiben Sie ein Java-Programm `ZaehleTEL` mit der Methode

```
public static void countTEL(char[] zeichen),
```

die in einer Eingabe vom Benutzer die Häufigkeit der Buchstaben 't', 'e' und 'l' zählt. Es sollen sowohl die Kleinbuchstaben als auch die entsprechenden Großbuchstaben gezählt werden. Dann soll eine entsprechende Meldung ausgegeben werden.

Beispiel: auf die Eingabe `HaLlo Welt` sollte ihr Programm *genau* die folgende Ausgabe produzieren (Beachten Sie die doppelten Anführungszeichen):

```
Der Buchstabe "t" kommt 1 mal vor.  
Der Buchstabe "e" kommt 1 mal vor.  
Der Buchstabe "l" kommt 3 mal vor.
```

Gegeben ist die folgende Vorlage, die vom Benutzer eine Zeichenfolge abfragt und diese als Array von Zeichen (`char`) in der Variable `zeichen` speichert.

ZaehleTEL.java

---

```
import java.util.Scanner;  
  
public class ZaehleTEL {  
    public static void main(String args[]) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("Bitte geben Sie eine Zeichenfolge ein  
            : ");  
        String eingabe = scanner.nextLine();  
        char[] zeichen = eingabe.toCharArray();  
        countTEL(zeichen);  
        scanner.close();  
    }  
}
```

---

#### Aufgabe 4      Abgabe mit automatischen Tests in GATE                  Reverse an Array

Programmieren Sie ein Java-Programm mit dem Namen `ReverseArray` mit einer Methode

```
public static int[] reverse(int[] a),
```

das ein Integer-Feld entgegennimmt und ein neues zurück gibt, in dem sich die Einträge in entgegengesetzter Reihenfolge befinden.

Achtung: Modifizieren Sie das original Array nicht und achten sie darauf, dass Ihr Programm nicht abstürzt, falls **null** übergeben wird!

## **Abgabe**

Die Abgabe erfolgt bis zum 06.11.2025 12:00 Uhr auf GATE, <https://gate.ifi.lmu.de>.

*Zuletzt aktualisiert: 24. 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>