TU Ilmenau, Fakultät IA
FG Telematik/Rechnernetze
Prof. Dr.-Ing. G. Schäfer
http://www.tu-ilmenau.de/telematik/pua

TECHNISCHE UNIVERSITÄT
ILMENAU

Programmierung und Algorithmen WS 25/26
Übungsblatt 2

Die Lösungen der Aufgaben sind bis zum 02.11.25, 23:59 Uhr abzugeben.

Die Besprechung der Aufgaben erfolgt in KW 45.

Aufgabe 1 (Java Einmaleins) — 7 Punkte

Das untenstehende Java-Programm gibt sieben Zeilen auf der Konsole aus. Geben Sie den Inhalt jeder Zeile an und begründen Sie Ihre Antwort kurz. Ohne Begründung kann Ihre Antwort nicht gewertet werden!

```
public static void main(String[] args) {
    int i = 8;
    double d = --i;
    System.out.println(i + ", " + d);
    switch(i) {
        case 7: i = 4;
        default: i = 0;
    }
    System.out.println(i);
    short s = 32767;
    System.out.println(++s);
    d = d + --d;
    System.out.println(d);
    System.out.println(d > 0 ? "+" : "-");
    d = 28 / 10;
    System.out.println(d);
    int[] a = {21, 22, 23};
    int[] b = a;
    b[0] = 0;
    System.out.println(a[0]);
}
```

Aufgabe 2 (Datentypen in Java) — 5 Punkte

Entscheiden Sie für die folgenden Werte (!), ob Java *primitive Datentypen* für ihre Repräsentation zur Verfügung stellt. Falls ja, geben Sie *alle* passenden an.

(a) `true`	(b) -32768	(c) {1, 4, 2}	(d) "Hello Java"
(e) $\sqrt{2}$	(f) $\frac{3}{2}$	(g) 'Z'	(h) 9223372036854775807

Aufgabe 3 (Schleifenkonstrukte) — 6 Punkte

Die Schleifenkonstrukte `for`, `while` und `do…while` besitzen dieselbe Ausdruckskraft. Implementieren Sie in Java *drei unterschiedliche Varianten* zur Berechnung der Summe aller Elemente eines gegebenen Arrays `int[] a`. Verwenden Sie dabei jeweils ein anderes Schleifenkonstrukt (zum Beispiel implementiert in drei Methoden).

Hinweis: Die Summe eines leeren Arrays sei als `0` definiert.

Bitte wenden!

Aufgabe 4 (Maximale Teilsumme) **1 + 6 Punkte**

Gegeben sei eine Folge von n ganzen Zahlen in einem ungeordneten Array `int[] a`. Gesucht ist diejenige *zusammenhängende* Teilfolge, welche unter allen möglichen Teilfolgen die maximale Summe aufweist. Diese Summe wird häufig auch als *maximale Teilsumme* bezeichnet.
Beispiel: Für das Array `int[] a = {31, -41, 59, 26, -53, 58, 97, -93, -23, 84}` ist die maximale Teilsumme `187`, welche sich aus der Teilfolge `59, 26, -53, 58, 97` ergibt.

(a) Wie viele verschiedene zusammenhängende Teilfolgen gibt es in einer Folge der Länge n?

(b) Entwerfen Sie einen Algorithmus in Java zur Bestimmung der maximalen Teilfolge sowie deren Summe. Die Position der Teilfolge soll dabei durch die beiden Indizes `i, j` des ersten und letzten Elements (im Beispiel: `i = 2` und `j = 6`) als Ausgabe charakterisiert werden.

Hinweis: Auch eine leere Teilfolge ist erlaubt. Die Summe der leeren Teilfolge ist 0 und ihre Position kann mit den Indizes `i = -1` und `j = -1` charakterisiert werden.

Aufgabe 5 (Palindromprüfung) **5 Punkte**

Schreiben Sie ein Java-Programm, welches testet, ob ein `String` ein Palindrom[1] ist. Groß- und Kleinschreibung sowie Leer- und Satzzeichen sollen dabei nicht beachtet werden. Beispiele für Palindrome sind:

- Reliefpfeiler
- Lagerregal
- Na, Fakir, Paprikafan?

[1] Ein Palindrom ergibt vorwärts und rückwärts gelesen die selbe Zeichenkette