

Methoden und Arrays

Wie Methoden und Arrays funktionieren

Inhaltsverzeichnis

Methoden beschreiben.....	1
Methoden überladen.....	2
Datentypen bei Arrays.....	3
Tracing.....	4
Programmieren.....	4

Methoden beschreiben

In folgendem Java-Programm werden verschiedene Methoden aufgerufen. Geben Sie zu jeder Methode den Methodenkopf (Name, Parameter- und Rückgabe-Datentyp(en)) an.

Geben Sie (falls Rückgabe-Datentyp vorhanden) das Ergebnis des jeweiligen Aufrufs an.

Hinweis: `Scanner scanner = new Scanner(System.in);` ist kein Methodenaufruf.

Tipp: Suchen Sie in der Java-API (<https://docs.oracle.com/javase/8/docs/api/>) nach den

Klassen `PrintStream`, `Math`, `System`, `String` und `Scanner`.

```

public class MethodTest {
    public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Es geht los");
        double angle = Math.toDegrees(Math.PI);
        int max = Math.max((int)angle, 180);
        System.out.println(max);
        long time = System.nanoTime();
        String text = scanner.nextLine();
        int length = text.length();
        long min = Math.min(time, length);
        System.out.println(min);
        scanner.close();
    }
}

```

Methoden überladen

```

public class SumMethods {

    public static void main (String[] args) {
        System.out.println(sum(1L,2));
        System.out.println(sum(1,2L));
        System.out.println(sum(1.0f,2.0f));
        System.out.println(sum(1.0,2));
        System.out.println(sum(1L,2.0f));
        System.out.println(sum(1,2,0));
        System.out.println(sum(1L,2,0));
        System.out.println(sum(1,2.0f,0));
        System.out.println(sum(1,2,0.0));
        System.out.println(sum(1.0,2,0L));
    }

    // 1
    public static double sum(long x, int y) {
        return x+y;
    }

    // 2
    public static double sum(int x, long y) {
        return x+y;
    }

    // 3
    public static double sum(long x, long y) {
        return x+y;
    }

    // 4
    public static double sum(int x, float y) {
        return x+y;
    }

    // 5
    public static double sum(float x, float y) {
        return x+y;
    }

    // 6
    public static double sum(double x, float y) {
        return x+y;
    }

    // 7
    public static double sum(double x, long y) {
        return x+y;
    }

    // 8
    public static double sum(int x, int y, int z) {
        return x+y+z;
    }
}

```

```

// 9
public static double sum(long x, int y, int z) {
    return x+y+z;
}

// 10
public static double sum(long x, long y, int z) {
    return x+y+z;
}

// 11
public static double sum(long x, long y, long z) {
    return x+y+z;
}

// 12
public static double sum(double x, long y, float z) {
    return x+y+z;
}

// 13
public static double sum(int x, float y, double z) {
    return x+y+z;
}

// 14
public static double sum(int x, float y, int z) {
    return x+y+z;
}

// 15
public static double sum(int x, double y, int z) {
    return x+y+z;
}
}

```

Geben Sie bei diesem Java-Programm an, welche der Methoden aufgerufen werden.

Beachten Sie, dass an manchen Stellen ein implizites Casting stattfinden kann.

Datentypen bei Arrays

Was gibt das folgende Java-Programm aus? Welche Array-Zugriffe sind zulässig?

Welchen Datentyp haben die Variablen, die ausgegeben werden?

```

public class PrintTest {
    public static void main (String[] args) {
        int[] arrayI = {123,456,789};
        boolean[][] arrayB = {{false},{true},{},new boolean[5]};
        System.out.println(arrayB[3]);
        System.out.println(arrayI[2]);
        System.out.println(arrayB[0][0]);
        System.out.println(arrayI);
        System.out.println(arrayB[2][0]);
    }
}

```

Tracing

Tragen Sie die Belegung der Einträge des Arrays in die nebenstehenden Kästchen ein.

```
public class ArrayTracing {
    public static void main (String[] args) {
        int[] array = {2,4,8,5,9};
        array[1] = 1;
        array[3] *= 2;
        array[1] = -array[4];
        array[4] = array[3]+array[1];
        --array[3];
        array[1] += array[17%3];
        array[array[0]] /= array[0];
        array[0] = 7;
        array[7] = 0;
    }
}
```

2 4 8 5 9

2 1 8 5 9

Programmieren

Schreiben Sie ein Java-Programm *stars*, das vom Benutzer eine ganze Zahl einliest und anschließend diese Anzahl an Zeilen mit Sternen ausgibt, wobei jede Zeile entsprechend der Zeilennummer viele Sterne enthält. Beispiel zur Eingabe 6:

```
*
**
***
****
*****
*****
```