

## Aufgabe 1

Aussage	Bewertung
Wird ein Wert eines primitiven Datentyps an eine Methode übergeben, spricht man von „ <i>call by value</i> “. Bei Referenzdatentypen spricht man dagegen von „ <i>call by reference</i> “.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
Die Daten eines Arrays liegen immer auf dem Heap.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
Der Plus-Operator in Java ist für den Datentyp <code>String</code> überladen.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
Funktionsaufrufe können nicht verschachtelt werden.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
In Java wird Heap-Speicher nicht manuell verwaltet – diese Aufgabe übernimmt der Garbage Collector.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
Weist man einer <code>int</code> -Variablen „ <code>null</code> “ zu, dann bedeutet das, dass die Variable keinen Wert enthalten soll.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
Alle Exception-Klassen sind Unterklassen der <code>Throwable</code> -Klasse.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
Selbst geschriebene Exception-Klassen sind immer checked Exceptions.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch
Java kennt keine impliziten Typkonvertierungen.	<input type="checkbox"/> wahr <input type="checkbox"/> falsch

## Aufgabe 2

Was tut die folgende Methode?

```
private static int calculate(int n) {  
    if (n < 0) {  
        throw new IllegalArgumentException();  
    }  
    if (n == 0) {  
        return 0;  
    }  
    return n % 2 + calculate(n / 2);  
}
```

### Aufgabe 3

Was tut die folgende Methode?

```
private static int calculate(String string) {
    if (string == null || string.length() < 1) {
        throw new IllegalArgumentException();
    }
    int a = 1;
    int b = 0;
    for (int i = string.length() - 1; i >= 0; --i) {
        if (string.charAt(i) < '0' || string.charAt(i) > '9') {
            throw new IllegalArgumentException();
        }
        b += a * (string.charAt(i) - '0');
        a *= 10;
    }
    return b;
}
```

---

### Aufgabe 4

Was tut die folgende Methode?

```
private static int[] calculate(int[] array) {
    if (array == null) { throw new IllegalArgumentException(); }
    Random random = new Random();
    boolean[] b = new boolean[array.length];
    int[] result = new int[array.length];
    for (int i = 0; i < result.length; ++i) {
        while (true) {
            int j = random.nextInt(array.length);
            if (!b[j]) {
                result[i] = array[j];
                b[j] = true;
                break;
            }
        }
    }
    return result;
}
```

## Aufgabe 5

Was tut die folgende Methode?

```
private static String transformString(String string, int n) {
    if (string == null || n < 0) {
        throw new IllegalArgumentException();
    }
    n %= 26;
    String result = "";
    for (int i = 0; i < string.length(); ++i) {
        char c = string.charAt(i);
        if (c < 'a' || c > 'z') {
            throw new IllegalArgumentException();
        }
        int o = c - 'a';
        o = (o + n) % 26;
        result += (char)('a' + o);
    }
    return result;
}
```

---

## Aufgabe 6

Was tut die folgende Methode?

```
private static int calculate(int n) {
    int result = 0;
    while (n > 0) {
        result += n % 10;
        n /= 10;
    }
    return result;
}
```

---

## Aufgabe 7

Schreibe eine statische Methode, die ein `double`-Array `array` sowie eine Ganzzahl `n` entgegennimmt. Die Methode soll aus jeweils `n` Werten des Arrays den Durchschnittswert bilden. Sämtliche Durchschnittswerte sollen dann in Form eines Arrays zurückgegeben werden.

Beispiel:

array = { 1.0, 2.0, 3.0, 4.0, 5.0 }  
n = 3

Ergebnis: { 2.0, 3.0, 4.0 }

array = { 1.0, 2.0, 3.0, 4.0, 5.0 }  
n = 3

Ergebnis: { 2.0, 3.0, 4.0 }

array = { 1.0, 2.0, 3.0, 4.0, 5.0 }  
n = 3

Ergebnis: { 2.0, 3.0, 4.0 }

### Aufgabe 8

Schreibe eine statische Methode `isSubset()`, die zwei `int`-Arrays übergeben bekommt und überprüfen soll, ob jedes Element des ersten Arrays auch im zweiten Array enthalten ist.

---

### Aufgabe 9

Schreibe eine statische Methode `containsDuplicates()`, die ein `int`-Array übergeben bekommt und überprüft, ob mindestens ein Array-Element mehrfach im Array vorkommt.

---

### Aufgabe 10

Schreibe eine statische Methode `split()`, die einen `String string` und einen `char separator` entgegennimmt und diesen Anhand des Werts von `separator` in einzelne Strings zerlegt. Die einzelnen Strings sollen als Array zurückgegeben werden.

Beispiel:

Der Aufruf `split("Hello, world! Hello, Alice and Bob!", ',')`

soll die folgenden Werte zurückliefern:

```
{ "Hello", " world! Hello", " Alice and Bob!" }
```