

# Aufgaben – Methoden in Java

## Allgemeine Hinweise

- Erstelle für jede Aufgabe eine eigene Java-Datei mit einer passenden Klasse.
- Jede Klasse soll eine `public static void main(String[] args)` -Methode enthalten.
- Achte darauf, Methoden zu schreiben, die wiederverwendbar sind und keinen doppelten Code enthalten.

### 1) PrintName

**Aufgabe:** Erstelle eine Methode `printName(String name)` , die den übergebenen Namen in einem vollständigen Satz ausgibt.

**Beispielausgabe für den Namen** `Maria` :

`Hallo Maria, schön dich zu sehen!`

**Anforderungen:**

- Die Methode gibt nichts zurück ( `void` ).
- Die Methode nimmt genau einen Parameter vom Typ `String` entgegen.
- Verwende die Methode in `main()` .

### 2) anzahlBuchstaben

**Aufgabe:** Schreibe eine Methode `anzahlBuchstaben(String text, char buchstabe)` , die zählt, wie oft der übergebene Buchstabe im Text vorkommt.

**Beispiel:**

- Eingabe: "Banane" , 'a' → Rückgabewert: 3

**Anforderungen:**

- Rückgabetyp: `int` .
- Wird wie im Beispiel ein Buchstabe in Kleinschreibung übergeben, sollen nur Kleinbuchstaben im Wort gezählt werden. Wird ein Großbuchstabe übergeben, sollen nur Großbuchstaben im Wort gezählt werden.
- Teste die Methode in `main()` .

## 3) findeMax

**Aufgabe:** Schreibe eine Methode `findeMax(int[] zahlen)` , die das größte Element eines ganzzahligen Arrays zurückgibt.

**Anforderungen:**

- Rückgabetyp: `int` .
- Teste die Methode mit unterschiedlichen Arrays (inkl. negatives, positives, gemischt).

## 4) kombiniereArrays

**Aufgabe:** Schreibe eine Methode `kombiniereArrays(int[] a, int[] b)` , die ein **neues** Array zurückgibt, das zuerst alle Elemente von `a` und danach alle Elemente von `b` enthält.

**Beispiel:**

- `a = {1,2,3}` , `b = {4,5}` → Rückgabe: `{1,2,3,4,5}`

**Anforderungen:**

- Rückgabetyp: `int[]` .
- Erstelle ein neues Array mit der passenden Länge und fülle es.
- Verändere die Eingangsarrays nicht.
- Teste die Methode in `main()` .

## 5) Bankautomat (Bonusaufgabe)

**Aufgabe:** Erstelle einen einfachen Bankautomaten mit den Methoden:

- `showBalance(double balance)` – zeigt den aktuellen Kontostand an.
- `deposit(Scanner scanner)` – liest einen Betrag ein, prüft, dass er nicht negativ ist, gibt den Betrag zurück.
- `withdraw(Scanner scanner, double balance)` – liest einen Betrag ein, prüft, dass genug Guthaben vorhanden ist, gibt den Auszahlungsbetrag zurück.

**Anforderungen:**

1. Nutze eine `while` -Schleife in `main()` , um ein Menü anzubieten:
  - i. Kontostand anzeigen
  - ii. Einzahlung
  - iii. Auszahlung
  - iv. Beenden
2. Aktualisiere den Kontostand basierend auf Ein- und Auszahlungen.
3. Scanner wird in `main()` erstellt und an die Methoden übergeben.

4. Achte darauf, Methoden wiederzuverwenden und keinen doppelten Code zu schreiben.

5. Bonus: Verwende `printf` für eine schöne Formatierung des Kontostands.

**Beispielausgabe:**

```
# # # # # # # #  
# BANK AUTOMAT #  
# # # # # # # #
```

1. Kontostand anzeigen
2. Einzahlung
3. Auszahlung
4. Exit

Was möchten Sie tun?: 1

Hier ist ihr aktueller Kontostand:

```
# # # # # # # #  
# 10.00 € #  
# # # # # # # #
```