

Java Übungsaufgaben

Diese Übungen decken grundlegende und fortgeschrittene Java-Konzepte ab, einschließlich **Methoden**, **Arrays**, **Listen**, **Sets**, **Maps** und **Kontrollstrukturen**. Sie bieten eine praktische Möglichkeit, das Erstellen und Verwenden von Methoden, die Manipulation von Datensammlungen und die Nutzung von Kontrollstrukturen zu üben.

1. Grundlagen der Methoden

Aufgabe 1: Begrüßungsmethode

- Erstelle eine Methode `begruesseBenutzer`, die den Namen des Benutzers als Parameter entgegennimmt und eine Begrüßungsnachricht in der Konsole ausgibt.

```
public void begruesseBenutzer(String name) {  
    // Ausgabe der Begrüßung  
}
```

Aufgabe 2: Quadrat berechnen

- Erstelle eine Methode `berechneQuadrat`, die eine Zahl als Parameter entgegennimmt und das Quadrat der Zahl zurückgibt.

```
public int berechneQuadrat(int zahl) {  
    // Berechnung des Quadrats  
    return 0; // Ersetze mit der richtigen Berechnung  
}
```

Aufgabe 3: Durchschnitt berechnen

- Schreibe eine Methode `berechneDurchschnitt`, die drei Zahlen als Parameter entgegennimmt und den Durchschnitt der Zahlen berechnet und zurückgibt.

```
public double berechneDurchschnitt(int zahl1, int zahl2, int zahl3) {  
    // Berechne den Durchschnitt  
    return 0; // Ersetze mit der richtigen Berechnung  
}
```

2. Arrays und Listen

Aufgabe 1: Wochentage speichern und ausgeben

- Erstelle ein Array `tage`, das die Namen der sieben Wochentage speichert. Gib die Namen der Wochentage in der Konsole aus.

```
String[] tage = new String[7];  
// Füge die Wochentage zum Array hinzu und gib es aus
```

Aufgabe 2: Temperaturen analysieren

- Erstelle ein Array `temperaturen` mit zehn Beispielttemperaturen. Berechne die durchschnittliche Temperatur und finde die höchste und niedrigste Temperatur im Array.

```
double[] temperaturen = {15.5, 17.3, 19.0, 16.2, 20.5, 18.1, 22.3, 21.1, 23.7,  
19.8};  
// Berechne Durchschnitt, Maximum und Minimum
```

Aufgabe 3: Noten in einer Liste speichern und summieren

- Verwende eine `ArrayList` von `Integer`, um eine Liste von Noten zu speichern. Berechne die Summe und den Durchschnitt aller Noten.

```
ArrayList<Integer> notenListe = new ArrayList<>();  
// Füge Noten zur Liste hinzu, berechne die Summe und den Durchschnitt
```

3. Kontrollstrukturen und Methoden

Aufgabe 1: Gerade und ungerade Zahlen zählen

- Erstelle eine Methode `zaehleGeradeUndUngerade`, die ein Array von Zahlen entgegennimmt und die Anzahl der geraden und ungeraden Zahlen im Array ausgibt.

```
public void zaehleGeradeUndUngerade(int[] zahlen) {  
    // Zähle die geraden und ungeraden Zahlen  
}
```

Aufgabe 2: Prüfe auf Primzahl

- Schreibe eine Methode `istPrimzahl`, die eine Zahl entgegennimmt und prüft, ob diese Zahl eine Primzahl ist.

```
public boolean istPrimzahl(int zahl) {  
    // Logik zur Überprüfung auf Primzahl
```

```
    return false;
}
```

Aufgabe 3: FizzBuzz

- Erstelle eine Methode `fizzBuzz`, die eine Zahl `n` entgegennimmt und die Zahlen von 1 bis `n` durchläuft. Für jede Zahl:
 - Gib "Fizz" aus, wenn die Zahl durch 3 teilbar ist.
 - Gib "Buzz" aus, wenn die Zahl durch 5 teilbar ist.
 - Gib "FizzBuzz" aus, wenn die Zahl sowohl durch 3 als auch 5 teilbar ist.
 - Gib andernfalls die Zahl selbst aus.

```
public void fizzBuzz(int n) {
    // Implementiere FizzBuzz-Logik
}
```

4. Arbeiten mit Sets und Maps

Aufgabe 1: Einzigartige Namen zählen

- Erstelle eine Methode `zaehleEinzigartigeNamen`, die eine `ArrayList` von Namen entgegennimmt und die Anzahl der einzigartigen Namen in einem `HashSet` speichert und ausgibt.

```
public void zaehleEinzigartigeNamen(ArrayList<String> namenListe) {
    // Speichere die Namen in einem HashSet und zähle die einzigartigen Einträge
}
```

Aufgabe 2: Telefonbuch mit HashMap

- Erstelle eine `HashMap`, die Namen (Schlüssel) und Telefonnummern (Werte) speichert. Füge drei Kontakte hinzu und gib das Telefonbuch aus.

```
HashMap<String, String> telefonbuch = new HashMap<>();
// Füge Einträge zum Telefonbuch hinzu und gib sie aus
```

Aufgabe 3: Häufigkeit von Wörtern zählen

- Erstelle eine Methode `zaehleWoerter`, die eine Liste von Wörtern entgegennimmt und eine `HashMap` zurückgibt, die die Häufigkeit jedes Wortes speichert.

```
public HashMap<String, Integer> zaehleWoerter(List<String> woerterListe) {
    // Implementiere Logik zur Zählung der Wort-Häufigkeit
}
```

```
    return null; // Ersetze null durch die richtige HashMap
}
```

5. Mehrdimensionale Arrays und Schleifen

Aufgabe 1: Matrix ausgeben

- Erstelle ein zweidimensionales Array `matrix` (3x3) und fülle es mit Zahlen. Gib die Matrix in einer gut lesbaren Form in der Konsole aus.

```
int[][] matrix = new int[3][3];
// Befülle das Array und gib es in Matrixform aus
```

Aufgabe 2: Summe der Elemente in einer Matrix berechnen

- Schreibe eine Methode `berechneMatrixSumme`, die eine 3x3-Matrix entgegennimmt und die Summe aller Elemente berechnet und zurückgibt.

```
public int berechneMatrixSumme(int[][] matrix) {
    // Berechne die Summe der Elemente in der Matrix
    return 0; // Ersetze 0 durch die richtige Summe
}
```

Aufgabe 3: Diagonale einer Matrix ausgeben

- Erstelle eine Methode `gibDiagonaleAus`, die eine quadratische Matrix entgegennimmt und die Elemente der Hauptdiagonale ausgibt.

```
public void gibDiagonaleAus(int[][] matrix) {
    // Gib die Elemente der Hauptdiagonale aus
}
```

6. Fortgeschrittene Übungen

Aufgabe 1: Array umdrehen

- Schreibe eine Methode `dreheArrayUm`, die ein Array entgegennimmt und die Reihenfolge der Elemente umkehrt.

```
public int[] dreheArrayUm(int[] array) {
    // Kehre die Reihenfolge der Elemente um
}
```

```
    return array;
}
```

Aufgabe 2: Menge der gemeinsamen Elemente finden

- Erstelle eine Methode `findeGemeinsameElemente`, die zwei `HashSet`-Mengen entgegennimmt und ein neues Set mit den gemeinsamen Elementen zurückgibt.

```
public HashSet<Integer> findeGemeinsameElemente(HashSet<Integer> set1,
HashSet<Integer> set2) {
    // Finde die gemeinsamen Elemente beider Mengen
    return null; // Ersetze null durch die richtige Menge
}
```

Aufgabe 3: Wörter mit Länge filtern

- Schreibe eine Methode `filtereWoerterMitLaenge`, die eine Liste von Wörtern und eine gewünschte Länge entgegennimmt und eine neue Liste mit Wörtern dieser Länge zurückgibt.

```
public List<String> filtereWoerterMitLaenge(List<String> woerterListe, int laenge)
{
    // Filtere die Wörter und gib eine neue Liste zurück
    return null; // Ersetze null durch die gefilterte Liste
}
```

Zusammenfassung der Themen

Thema	Beschreibung
Methoden	Definieren und Verwenden von Methoden mit Parametern und Rückgabewerten
Arrays	Initialisieren und Verarbeiten von Arrays, Zugriff auf Elemente, Summen und Durchschnitte
Listen	Arbeiten mit <code>ArrayList</code> zur Speicherung von Sammlungen und Berechnung von Werten
Sets	Verwenden von <code>HashSet</code> , um einzigartige Werte zu speichern und zu filtern
Maps	Erstellen und Verwenden von <code>HashMap</code> , um Schlüssel-Wert-Paare zu speichern und zu verwalten

Diese Übungen decken wichtige Programmierkonzepte in Java ab und bieten eine umfassende Grundlage für das Arbeiten mit Datenstrukturen und Kontrollstrukturen.