

# Kapitel 11 – Interfaces und abstrakte Klassen: Zusammenfassung + Übungen

---

**In diesem Kapitel lernst du den Unterschied zwischen Interfaces und abstrakten Klassen kennen. Beides sind Werkzeuge zur Strukturierung von Code und zur Definition von gemeinsamem Verhalten über Klassengrenzen hinweg.**

## Themenübersicht

### 1. Interface

Ein Interface ist eine Sammlung abstrakter Methodendefinitionen. Es enthält keine Implementierungen (außer default).

### 2. Interface implementieren

Eine Klasse kann ein Interface mit `implements` umsetzen und muss alle Methoden definieren.

### 3. Mehrfaches Implementieren

Eine Klasse kann mehrere Interfaces gleichzeitig implementieren: `implements A, B`.

### 4. Abstrakte Klasse

`abstract class` kann implementierte und nicht implementierte Methoden enthalten.

### 5. Abstrakte Methoden

Werden in der abstrakten Klasse deklariert, aber nicht implementiert. Subklassen müssen sie überschreiben.

### 6. Unterschied Interface vs. abstrakte Klasse

- Interface: Nur Methodensignaturen, keine Felder, keine Konstruktoren
- Abstrakte Klasse: Kann Felder, Konstruktoren und implementierte Methoden enthalten

### 7. Gemeinsamkeiten

Beides zwingt Unterklassen zur Implementierung bestimmter Methoden.

## Übungsaufgaben

1. Erstelle ein Interface `Beweglich` mit der Methode `void bewegen()`.
2. Implementiere das Interface in einer Klasse `Auto`, die `bewegen()` mit "fährt" umsetzt.
3. Füge eine zweite Klasse `Fahrrad` hinzu, die ebenfalls `Beweglich` implementiert.

4. 4. Erstelle eine abstrakte Klasse `Tier` mit einer abstrakten Methode `geraeusch()`, und einer konkreten Methode `bewegeDich()`.
5. 5. Leite eine Klasse `Hund` von `Tier` ab und implementiere `geraeusch()` mit "Wuff".
6. 6. Erkläre, warum man Interfaces verwenden sollte, obwohl Java Mehrfachvererbung nicht unterstützt.
7. 7. Was passiert, wenn eine Klasse eine Methode aus einem Interface nicht implementiert?
8. 8. Vergleiche eine abstrakte Klasse mit einem Interface anhand eines praktischen Beispiels.