

## **Java Basics**

### **Aufgabe 1 – Java-Programmaufbau**

Skizziere die grundlegende Struktur eines Java-Programms. Erkläre jede Komponente jeweils mit einem Satz.

### Aufgabe 2 – Einfaches Kompilieren von Java-Programmen

Schreibe, kompiliere und führe ein einfaches Java-Programm aus, das "Hallo Welt!" ausgibt. Befolge die untenstehenden Schritte und beantworte die zugehörigen Fragen.

#### 1. Schreibe das Programm:

- Erstelle eine Java-Klasse namens *HelloWorld*.
- Füge eine main-Methode hinzu, die den Text "Hallo Welt!" auf der Konsole ausgibt.

#### 2. Kompiliere das Programm:

- Verwende den javac-Befehl, um das Programm zu kompilieren.
- Beschreibe den Prozess und notiere die Kompilierungsbefehle, die du verwendet hast.

#### 3. Führe das Programm aus:

- Verwende den java-Befehl, um die kompilierte .class-Datei auszuführen.
- Beschreibe den Prozess und notiere die Ausführungsbefehle, die du verwendet hast.

### Aufgabe 3 – Klassen und Objekte

Fülle die Lücken passend:

\_\_\_\_\_

Eine \_\_\_\_\_ beschreibt die Struktur und das Verhalten einer Menge gleichartiger \_\_\_\_\_.

\_\_\_\_\_

Ein \_\_\_\_\_ ist ein spezielles, individuelles Exemplar einer \_\_\_\_\_, das durch spezifische Werte seiner \_\_\_\_\_ gekennzeichnet ist.

Lösung unter: [https://www.mosbach.dhbw.de/fileadmin/user\\_upload/dhbw/redaktion/mitarbeiter/neuendorf/Java1\\_OO\\_Grundlagen.pdf](https://www.mosbach.dhbw.de/fileadmin/user_upload/dhbw/redaktion/mitarbeiter/neuendorf/Java1_OO_Grundlagen.pdf) Folie 8

## Aufgabensammlung: Wiederholung 811

### Aufgabe 4 - OOP

Erkläre das Konzept der **Objektorientierten Programmierung** (OOP):

### Aufgabe 5 – Prinzipien der OOP

Erkläre in ein bis zwei Sätzen, was die nachfolgenden Prinzipien der OOP bedeuten. Gib jeweils ein Beispiel an.

	Erklärung	Beispiel
Abstraktion (Abstraction)		
Kapselung (En- capsulation)		
Vererbung (Inhe- ritance)		
Polymorphismus (Polymorphism)		

Lösung unter: [https://www.mi.fu-berlin.de/wiki/pub/ABI/AlDaBiWS13Praktikum/02\\_WS13\\_OOP\\_Templates.pdf](https://www.mi.fu-berlin.de/wiki/pub/ABI/AlDaBiWS13Praktikum/02_WS13_OOP_Templates.pdf) Folie 6

### Aufgabe 6 – Anwendung der Prinzipien der OOP

Erstelle ein Java-Programm, das die vier grundlegenden Prinzipien der Objektorientierten Programmierung (OOP) demonstriert: Kapselung, Vererbung, Polymorphismus und Abstraktion. Verwende dafür das Beispiel eines Fahrzeug-Systems.

#### Anforderungen:

##### 1. Kapselung (Encapsulation):

Implementiere eine Klasse *Fahrzeug* mit privaten Attributen *marke* und *geschwindigkeit*. Stelle öffentliche Methoden (getter und setter) bereit, um auf diese Attribute zuzugreifen und sie zu ändern.

##### 2. Vererbung (Inheritance):

Erstelle eine Klasse *Auto*, die von *Fahrzeug* erbt und ein zusätzliches Attribut *anzahlTueren* hat.

##### 3. Polymorphismus (Polymorphism):

Erstelle eine Methode *start()* in der Klasse *Fahrzeug* und überschreibe diese Methode in der Klasse *Auto*, um spezifische Informationen anzuzeigen.

##### 4. Abstraktion (Abstraction):

Erstelle eine abstrakte Klasse *Mitarbeiter* mit einer abstrakten Methode *berechneGehalt()*. Erstelle zwei Unterklassen *VollzeitMitarbeiter* und *TeilzeitMitarbeiter*, die die Methode *berechneGehalt()* implementieren.

**Aufgabe 7 – Unterscheidung zwischen static und instance**

a) Worin besteht der Unterschied zwischen static und instance?

b) Implementiere folgendes Java-Programm:

1. Implementiere eine Klasse *Rechner*, die eine statische Variable *pi* und eine statische Methode *berechneKreisumfang(double radius)* enthält.
2. Implementiere in der Klasse *Rechner* eine Instanz-Variable *ergebnis* und eine Instanz-Methode *addiere(double a, double b)*.
3. Implementiere eine statische innere Klasse *MathUtils* in der Klasse *Rechner*, die eine statische Methode *multiplikation(double a, double b)* enthält.
4. Erstelle in der *main*-Methode ein Objekt der Klasse *Rechner* und demonstriere den Zugriff auf die statischen und Instanz-Mitglieder sowie die statische Klasse.