Lösungsvorschlag

Thema: Objektorientierte Programmierung (Java) – Einführung & Recherche

Bezug: Aufgabenblatt 2 (AB2-OOP.pdf).

Aufgabe 1: Was bildet Programmieren ab? Was sind eigentlich Programme? [4 BE]

Ein **Computerprogramm** ist eine *folgerichtige, endliche Folge von Anweisungen*, die ein Computer ausführt, um ein bestimmtes Problem zu lösen. Programme *modellieren reale Abläufe/Daten* in einer formalen Sprache: Eingaben werden verarbeitet (Rechnen, Vergleichen, Entscheiden), Ausgaben werden erzeugt (z. B. Text, Grafik, Dateien). Kurz: *Programme sind präzise Rezepte*, die die Maschine ohne Deutungsschritte befolgen kann.

Aufgabe 2: Prozedural vs. Objektorientiert [8 BE]

Prozedurale Programmierung	**Objektorientierte Programmierung (OOP)**
Denkt in *Schritten/Prozeduren/Funktionen*. Daten werden an Funktionen übergeben.	Denkt in *Objekten*: *Daten (Attribute) + Verhalten (Methoden)* gehören zusammen.
Zustand (Daten) und Logik (Funktionen) sind getrennt; globale Daten sind üblich(er).	**Kapselung**: interne Daten werden geschützt; Zugriff nur über Methoden.
Gut für lineare Abläufe, kleine bis mittlere Programme.	Gut für komplexe Modelle (z. B. GUI, Spiele, Simulationswelten). **Wiederverwendung** über Klassen-/Vererbung/Komposition.
Beispiele: C, Pascal, prozeduraler Stil in Python.	Beispiele: Java, C#, Python (OOP-Stil), C++ (multi-paradigm).

Kernaussage: Prozedural = „*Wie* wird etwas getan?“ (Schrittfolgen). OOP = „*Wer* tut etwas?“ (Objekte mit Verantwortung).

Aufgabe 3: Begriffsklärung [16 BE]

Jeweils kurz erklären (1–3 Sätze) + Beispiel.

1. **Objekt:** *konkretes Exemplar* mit eigenem Zustand und Verhalten.
   Beispiel: `meinRad` mit Gangzahl = 3, Geschwindigkeit = 12 km/h; Methoden `schalteGang(…)`.

2. **Klasse:** *Bauplan/Schablone* für Objekte: definiert Attribute und Methoden.
   Beispiel: `class Fahrrad { int gaenge; void bremsen() {...}}`.

3. **Attribut (Feld):** *Eigenschaft* eines Objekts (speichert Daten/Zustand).
   Beispiel: `private int gaenge;` oder `private String farbe;`.

4. **Methode:** *Fähigkeit/Verhalten* eines Objekts (Prozedur/Funktion in der Klasse).
   Beispiel: `public void beschleunigen(double delta) {...}`.

5. **Konstruktor:** Spezial-Methode zum *Erzeugen/Initialisieren* eines Objekts (gleichnamig wie Klasse, ohne Rückgabetyp).
   Beispiel: `public Fahrrad(int gaenge, String farbe) { this.gaenge=gaenge; this.farbe=farbe; }`.

6. **Instanz:** *konkretes, erzeugtes Objekt* einer Klasse.
   Beispiel: `Fahrrad meinRad = new Fahrrad(7, "blau");`

Aufgabe 4: Von Alltagsobjekt zur Java-Klasse [12 BE]

Beispielobjekt: **Fahrrad**.

a) Attribute und Methoden

Mögliche Attribute (mind. 4):

- `marke: String`, `gaenge: int`, `aktuelleGeschwindigkeit: double`, `lichtAn: boolean`

Mögliche Methoden (mind. 4):

- `beschleunigen(double delta)`, `bremsen(double delta)`, `schalteGang(int neu)`, `klingeln()`

b) Java-Klasse (Rohentwurf: Felder, Konstruktor, 2–3 Methoden genügen)

```java
// Klassenname: Fahrrad
class Fahrrad {

  // Felder (Attribute)
  private String marke;
  private int gaenge;
  private double aktuelleGeschwindigkeit; // km/h
  private boolean lichtAn;

  // Konstruktor
  public Fahrrad(String marke, int gaenge) {
    this.marke = marke;
    this.gaenge = gaenge;
    this.aktuelleGeschwindigkeit = 0.0;
    this.lichtAn = false;
  }

  // Methoden (Signaturen + einfache Logik)
  public void beschleunigen(double delta) {
    if (delta > 0) {
      aktuelleGeschwindigkeit += delta;
    }
  }

  public void bremsen(double delta) {
    if (delta > 0) {
      aktuelleGeschwindigkeit -= delta;
      if (aktuelleGeschwindigkeit < 0) {
        aktuelleGeschwindigkeit = 0;
      }
    }
  }

  public boolean schalteGang(int neu) {
    if (neu >= 1 && neu <= gaenge) {
      // hier könnte man den aktuellen Gang speichern (extra Feld)
      return true;
    }
    return false;
  }

  public void klingeln() {
    System.out.println("Klingeling!");
  }

  // (Optional) Getter als Beispiel für Kapselung
  public double getAktuelleGeschwindigkeit() {
    return aktuelleGeschwindigkeit;
  }
```

```
50      }
```

Hinweise zur Bewertung (transparent für SuS).

- **A1 (4 BE):** korrekte, knappe Definition (≈ 2–3 Sätze), Zweck/Funktion klar.
- **A2 (8 BE):** Unterschiede klar benannt (Denkrichtung, Kapselung, Wiederverwendung) + passende Beispiele.
- **A3 (16 BE):** alle 6 Begriffe sauber erklärt; je ein Beispiel (Java-nah).
- **A4 (12 BE):** mind. 4 Attribute & 4 Methoden sinnvoll; Klasse mit Feldern, Konstruktor, 2–3 Methoden korrekt (Syntax/Benennung).

Quelle der Aufgabenstellung: AB2-OOP.pdf (Aufgabenblatt 2 – OOP).