

Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2021 / Frühjahr

Thema 1 / Teilaufgabe 1 / Aufgabe 2 (Verstoß gegen die Prinzipien guter objektorientierter Programmierung)

Stichwörter: Objektorientierung

Lesen Sie die folgenden Beispielcodes gründlich. Identifizieren Sie für jeden Beispielcode den jeweiligen wesentlichen Verstoß gegen die Prinzipien guter objektorientierter Programmierung. Benennen und erklären Sie jeweils den Verstoß (Fehler) in einem Satz und erläutern Sie für jeden Beispielcode, welche Probleme aus dem jeweiligen Fehler resultieren können, ebenfalls in einem Satz.

```
(a) class Rectangle {
    private int width;
    private int length;

    Rectangle(int w, int l) {
        width = w;
        length = l;
    }

    public int getWidth() {
        return this.width;
    }

    public int getLength() {
        return this.length;
    }
}

class RectangleDemo {
    public static void main(String args[]) {
        Rectangle rectangle1 = new Rectangle(10, 20);
        Rectangle rectangle2 = new Rectangle(3, 90);
        Rectangle example = new Rectangle(1, 2);
        int area;

        // Compute area of first box
        area = rectangle1.getWidth() * rectangle1.getLength();
        System.out.println("Area is " + area);

        // Compute area for second box
        area = rectangle2.getWidth() * rectangle2.getLength();
        System.out.println("Area is " + area);

        // Compute area for third box
        area = example.getWidth() * example.getLength();
        System.out.println("Area is " + area);
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/Rectangle.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/Rectangle.java)

Lösungsvorschlag

Es sollte die Methode `computeArea` in der Klasse `Rectangel` implementiert werden.

```
(b) public class CalculateSpeed {
    private double kilometers;
    private double minutes;

    public CalculateSpeed(double k, double m) {
        this.kilometers = k;
        this.minutes = m;
    }

    // Display the speed
    void speed() {
        double speed;
        speed = kilometers / (minutes / 60);
        System.out.println("A car traveling " + kilometers + " kilometers in " + minutes +
            ↪ " minutes travels at " + speed
            + " kilometers per hour");
    }

    public static void main(String args[]) {
        CalculateSpeed car = new CalculateSpeed(110.0, 120.0);

        // Display car speed
        car.speed();

        // Display bicycle speed
        double speed;

        speed = 20.0 / (80.0 / 60);
        // So steht es in der Angabe
        // System.out.println("A bicycle traveling " + kilometers + " kilometers in " +
        ↪ minutes + " minutes travels at "
        // + speed + " kilometers per hour");
        // Ohne Fehler:
        System.out.println("A bicycle traveling " + car.kilometers + " kilometers in " +
        ↪ car.minutes + " minutes travels at "
        + speed + " kilometers per hour");
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/CalculateSpeed.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/CalculateSpeed.java)

Lösungsvorschlag

Klassen sollten nach Objekten modelliert werden und nicht nach Tätigkeiten (berechne Geschwindigkeit). Besser wäre der Name `SpeedCalculator` gewesen. Außerdem sind die beiden Attribute `kilometer` und `miutes` in der `Main` Methode so nicht ansprechbar, weil sie nicht statisch sind.

```
(c) class Stack {
```

```
int stck[] = new int[3];
public int top;

// Initialize top of stack
Stack() {
    top = -1;
}

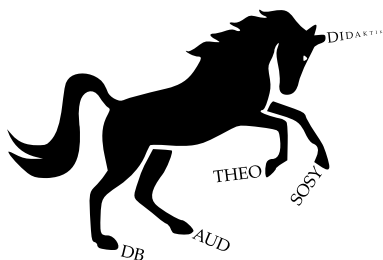
// Push an item on the stack
void push(int item) {
    if (top == 2) {
        System.out.println("Stack is full.");
    } else {
        stck[++top] = item;
    }
}

// Pop an item from the stack
int pop() {
    if (top < 0) {
        throw new IllegalStateException("Stack is empty.");
    } else {
        return stck[top--];
    }
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/Stack.java](https://github.com/org/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/Stack.java)

Lösungsvorschlag

Das Feld stck sollte private sein. So wird die interne Implementetation verborgen.



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66116/2021/03/Thema-1/Teilaufgabe-1/Aufgabe-2.tex>