

## Aufgabe 2

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 {
4     private int width;
5     private int length;
6
7     Rectangle(int w, int l) {
8         width = w;
9         length = l;
10    }
11
12    public int getWidth() {
13        return this.width;
14    }
15
16    public int getLength() {
17        return this.length;
18    }
19 }
20
21 class RectangleDemo {
22     public static void main(String args[]) {
23         Rectangle rectangle1 = new Rectangle(10, 20);
24         Rectangle rectangle2 = new Rectangle(3, 90);
25         Rectangle example = new Rectangle(1, 2);
26         int area;
27
28         // Compute area of first box
29         area = rectangle1.getWidth() * rectangle1.getLength();
30         System.out.println("Area is " + area);
31
32         // Compute area for second box
33         area = rectangle2.getWidth() * rectangle2.getLength();
34         System.out.println("Area is " + area);
35
36         // Compute area for third box
37         area = example.getWidth() * example.getLength();
38         System.out.println("Area is " + area);
39     }
40 }
```

github: raw

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

```
(b) public class CalculateSpeed {
4     private double kilometers;
5     private double minutes;
6
7     public CalculateSpeed(double k, double m) {
8         this.kilometers = k;
9     }
10 }
```

```

9         this.minutes = m;
10    }
11
12    // Display the speed
13    void speed() {
14        double speed;
15        speed = kilometers / (minutes / 60);
16        System.out.println("A car traveling " + kilometers + " kilometers in "
17            ↳ + minutes + " minutes travels at " + speed
18            ↳ + " kilometers per hour");
19    }
20
21    public static void main(String args[]) {
22        CalculateSpeed car = new CalculateSpeed(110.0, 120.0);
23
24        // Display car speed
25        car.speed();
26
27        // Display bicycle speed
28        double speed;
29
30        speed = 20.0 / (80.0 / 60);
31        // So steht es in der Angabe
32        // System.out.println("A bicycle traveling " + kilometers + "
33            ↳ kilometers in " + minutes + " minutes travels at "
34            ↳ + speed + " kilometers per hour");
35        // Ohne Fehler:
36        System.out.println("A bicycle traveling " + car.kilometers +
37            ↳ " kilometers in " + car.minutes + " minutes travels at "
38            ↳ + speed + " kilometers per hour");
39    }
40 }

```

github: raw

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

```

(c) class Stack {
4
5     int stck[] = new int[3];
6     public int top;
7
8     // Initialize top of stack
9     Stack() {
10         top = -1;
11     }
12
13     // Push an item on the stack
14     void push(int item) {
15         if (top == 2) {
16             System.out.println("Stack is full.");
17         } else {
18             stck[++top] = item;
19         }
20     }
21 }

```

```
21
22     // Pop an item from the stack
23     int pop() {
24         if (top < 0) {
25             throw new IllegalStateException("Stack is empty.");
26         } else {
27             return stck[top--];
28         }
29     }
30 }
```

[github: raw](#)

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