

Aufgabe 5

- (a) Implementieren Sie ein Programm in einer objektorientierten Programmiersprache, z. B. Java, für das folgende UML-Klassendiagramm.

Die `shift`-Methode soll die `x`-Position eines Objektes um `xShift` verändern und die `y`-Position um `yShift`. Die `draw`-Methode soll die Werte der Attribute der Klasse auf der Konsole ausgeben (- dies kann in Java mit `System.out.println` ("...") erfolgen).

```
3 interface Drawable {
4
5     public void draw();
6
7 }

                                     Code-Beispiel auf Github ansehen:
                                     src/main/java/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/object2d/Drawable.java

3 abstract class Object2D implements Drawable {
4     public abstract void shift(int xShift, int yShift);
5 }

                                     Code-Beispiel auf Github ansehen:
                                     src/main/java/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/object2d/Object2D.java

3 public class Point extends Object2D {
4     int xPos;
5     int yPos;
6
7     public Point(int x, int y) {
8         xPos = x;
9         yPos = y;
10    }
11
12    public void shift(int xShift, int yShift) {
13        xPos += xShift;
14        yPos += yShift;
15    }
16
17    public void draw() {
18        System.out.println(String.format("xPos: %s, yPos: %s", xPos,
19        ↪      yPos));
20    }
21 }

                                     Code-Beispiel auf Github ansehen:
                                     src/main/java/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/object2d/Point.java

3 public class Square extends Object2D {
4     Point topLeft;
5     Point bottomRight;
6
7     public Square(int top, int left, int bottom, int right) {
8         topLeft = new Point(left, top);
9         bottomRight = new Point(right, bottom);
10    }
11
12    public void shift(int xShift, int yShift) {
```

```

13     topLeft.shift(xShift, yShift);
14     bottomRight.shift(xShift, yShift);
15 }
16
17 public void draw() {
18     topLeft.draw();
19     bottomRight.draw();
20 }
21
22 }

```

Code-Beispiel auf Github ansehen:
[src/main/java/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/object2d/Square.java](https://github.com/orgs/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/object2d/Square.java)

- (b) Schreiben Sie eine Methode, die ein zweidimensionales Array aus ganzen Zahlen (Datentyp `int`) als Parameter bekommt und ein eindimensionales Array (bestehend aus ganzen Zahlen (Datentyp `int`)) zurückgibt, dessen Elemente jeweils der Summe der Einträge in der entsprechenden Zeile des zweidimensionalen Arrays entsprechen.

Achtung: Die Zeilen des zweidimensionalen Arrays können unterschiedlich lang sein.

Zur Vereinfachung sei die Signatur der Methode gegeben: `public int[] computeSum(int[] [] input)`

```

3 public class ComputeSum {
4     public static int[] computeSum(int[] [] input) {
5         int[] output = new int[input.length];
6         for (int i = 0; i < input.length; i++) {
7             int[] numbers = input[i];
8             int sum = 0;
9             for (int j = 0; j < numbers.length; j++) {
10                 sum += numbers[j];
11             }
12             output[i] = sum;
13         }
14         return output;
15     }
16
17     public static void main(String[] args) {
18         int[] [] input = new int[3] [];
19         input[0] = new int[] { 1, 2, 3 };
20         input[1] = new int[] { 4, 5 };
21         input[2] = new int[] { 6 };
22         int[] output = computeSum(input);
23         for (int i = 0; i < output.length; i++) {
24             System.out.println(output[i]);
25         }
26     }
27 }

```

Code-Beispiel auf Github ansehen:
[src/main/java/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/ComputeSum.java](https://github.com/orgs/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/ComputeSum.java)

- (c) Implementieren Sie eine einfach verkettete Liste in einer Klasse `List` (z. B. in Java), in der in jedem Listenelement ein String gespeichert wird. Die Klasse soll folgende Methoden bereitstellen:

- `void addFirst (String element)`: Diese Methode fügt ein Element am Anfang einer Liste ein.

- `void addLast (String element)`: Diese Methode hängt ein Element an das Ende der Liste an.
- `boolean exists (String element)`: Diese Methode gibt `true` zurück, wenn die Liste ein Element mit dem Inhalt `element` beinhaltet, andernfalls gibt sie `false` zurück.

Hinweis: Zwei `String`-Objekte können mittels der Funktion `equals(...)` verglichen werden.

```

3  public class List {
4
5      Element head;
6
7      class Element {
8          String value;
9          Element next;
10
11         public Element(String value) {
12             this.value = value;
13         }
14     }
15
16     public List() {
17         head = null;
18     }
19
20     void addFirst(String element) {
21         Element newElement = new Element(element);
22         if (head != null) {
23             newElement.next = head;
24         }
25         head = newElement;
26     }
27
28     void addLast(String element) {
29         Element nextElement = head;
30         Element lastElement = head;
31         while (nextElement != null) {
32             lastElement = nextElement;
33             nextElement = nextElement.next;
34         }
35         if (lastElement != null) {
36             lastElement.next = new Element(element);
37         } else {
38             head = new Element(element);
39         }
40     }
41
42     boolean exists(String element) {
43         Element nextElement = head;
44         while (nextElement != null) {
45             if (nextElement.value.equals(element)) {
46                 return true;
47             }
48             nextElement = nextElement.next;
49         }
50         return false;
51     }
52

```

```

53 public static void main(String[] args) {
54     List list = new List();
55     list.addLast("two");
56     list.addFirst("one");
57     list.addLast("three");
58
59     System.out.println(list.exists("one"));
60     System.out.println(list.exists("four"));
61     Element nextElement = list.head;
62     while (nextElement != null) {
63         System.out.println(nextElement.value);
64         nextElement = nextElement.next;
65     }
66 }
67 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/List.java](https://github.com/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/List.java)