

Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2020 / Frühjahr

Thema 2 / Teilaufgabe 1 / Aufgabe 5

(Object2D)

Stichwörter: Implementierung in Java

- (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).

Lösungsvorschlag

```
interface Drawable {  
  
    public void draw();  
  
}
```

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

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

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/object2d/Object2D.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2020/fruehjahr/object2d/Object2D.java)

```
public class Point extends Object2D {  
    int xPos;  
    int yPos;  
  
    public Point(int x, int y) {  
        xPos = x;  
        yPos = y;  
    }  
  
    public void shift(int xShift, int yShift) {  
        xPos += xShift;  
        yPos += yShift;  
    }  
  
    public void draw() {  
        System.out.println(String.format("xPos: %s, yPos: %s", xPos, yPos));  
    }  
  
}
```

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

```
public class Square extends Object2D {
    Point topLeft;
    Point bottomRight;

    public Square(int top, int left, int bottom, int right) {
        topLeft = new Point(left, top);
        bottomRight = new Point(right, bottom);
    }

    public void shift(int xShift, int yShift) {
        topLeft.shift(xShift, yShift);
        bottomRight.shift(xShift, yShift);
    }

    public void draw() {
        topLeft.draw();
        bottomRight.draw();
    }
}
```

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)`

```
public class ComputeSum {
    public static int[] computeSum(int[] [] input) {
        int[] output = new int[input.length];
        for (int i = 0; i < input.length; i++) {
            int[] numbers = input[i];
            int sum = 0;
            for (int j = 0; j < numbers.length; j++) {
                sum += numbers[j];
            }
            output[i] = sum;
        }
        return output;
    }

    public static void main(String[] args) {
        int[] [] input = new int[3] [];
        input[0] = new int[] { 1, 2, 3 };
        input[1] = new int[] { 4, 5 };
        input[2] = new int[] { 6 };
        int[] output = computeSum(input);
        for (int i = 0; i < output.length; i++) {
```

```
        System.out.println(output[i]);
    }
}
}
```

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

Lösungsvorschlag

```
public class List {

    Element head;

    class Element {
        String value;
        Element next;

        public Element(String value) {
            this.value = value;
        }
    }

    public List() {
        head = null;
    }

    void addFirst(String element) {
        Element newElement = new Element(element);
        if (head != null) {
            newElement.next = head;
        }
        head = newElement;
    }

    void addLast(String element) {
        Element nextElement = head;
        Element lastElement = head;
        while (nextElement != null) {
```

```

        lastElement = nextElement;
        nextElement = nextElement.next;
    }
    if (lastElement != null) {
        lastElement.next = new Element(element);
    } else {
        head = new Element(element);
    }
}

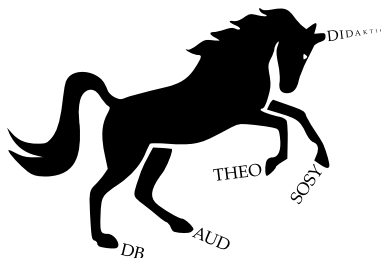
boolean exists(String element) {
    Element nextElement = head;
    while (nextElement != null) {
        if (nextElement.value.equals(element)) {
            return true;
        }
        nextElement = nextElement.next;
    }
    return false;
}

public static void main(String[] args) {
    List list = new List();
    list.addLast("two");
    list.addFirst("one");
    list.addLast("three");

    System.out.println(list.exists("one"));
    System.out.println(list.exists("four"));
    Element nextElement = list.head;
    while (nextElement != null) {
        System.out.println(nextElement.value);
        nextElement = nextElement.next;
    }
}
}

```

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



Die Bschlangaul-Sammlung

Hermine Bschlangauland 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/66116/2020/03/Thema-2/Teilaufgabe-1/Aufgabe-5.tex>