

## Thema Nr. 1

### Teilaufgabe Nr. 1

#### Aufgabe 1

- (a) Definieren Sie die Bedeutung der Sichtbarkeiten *private*, *package-private* und *protected* von Feldern in Java-Klassen. Erklären Sie diese kurz (je ein Satz).

**private** Das Feld ist innerhalb einer Klasse zugreifbar.

**package-private** Das Feld ist innerhalb der Klassen eines Packets zugreifbar.

**protected** Das Feld ist in allen Subklassen und in der eigenen Klasse zugreifbar.

- (b) Benennen Sie jeweils einen Grund für den Einsatz der Sichtbarkeiten *private*, *package-private* und *protected* von Feldern in Java-Klassen.

**private** Datenkapselung, Verbergen der internen Implementation. So kann die interne Implementation geändert werden, ohne dass die öffentlichen Schnittstellen sich ändern.

**package-private** Um die Felder zwar innerhalb eines Packets für alle Klassen zugänglich zu machen aber nicht außerhalb, z. B. in einem größeren Projekt.

**protected** Wenn eine Klasse vererbt werden soll, z. B. abstrakte Klassen.

#### 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         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 + "
    ↳ kilometers in " + minutes + " minutes travels at "
33     // + speed + " kilometers per hour");
34     // Ohne Fehler:
35     System.out.println("A bicycle traveling " + car.kilometers +
    ↳ " kilometers in " + car.minutes + " minutes travels at "
36     + speed + " kilometers per hour");
37 }
38 }

```

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
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.

### Aufgabe 3

Wählen Sie bis zu fünf unterschiedliche Elementtypen aus folgendem Diagramm aus und benennen Sie diese Elemente und ihre syntaktische (nicht semantische) Bedeutung.

**Klasse** ConcreteCreator  
**Interface** Produkt  
**Abstrakte Klasse** Creator  
**Kommentar** return new ConcreteProdukt()

## Aufgabe 4

- (a) Schreiben Sie ein Programm in einer objektorientierten Programmiersprache Ihrer Wahl, das den folgenden Anweisungen entspricht.
- (i) Es gibt eine Klasse mit dem Namen jBox.
  - (ii) Alle Zahlen sind Fließkommazahlen.
  - (iii) jBox wird mit einem Argument instanziiert, dessen Wert einer Variable namens jlength zugewiesen wird.
  - (iv) jBox hat eine Methode ohne Argumente namens jsize, welche den Wert von jlength zurückgibt.
  - (v) Eine weitere Methode namens jsize hat genau ein Argument namens jwidth. Diese zweite Methode namens jsize gibt das Produkt aus jwidth und jlength zurück. Eine weitere Methode namens jsize hat genau zwei Argumente, nämlich eine Zahl jnum und einen Faktor jf. Es wird jlength minus das Produkt aus jnum und jf zurückgegeben.
  - (vi) Schreiben Sie eine jmain-Methode, die eine Box namens jexample mit einer Länge von 15 anlegt.
  - (vii) Führen Sie die Methode jsize in der jmain-Methode wie unten angegeben drei Mal aus.
  - (viii) Speichern Sie hierbei das Ergebnis jeweils in einer Variable jmsize. Geben Sie das Ergebnis jeweils in einer eigenen Zeile des Ausgabemediums jSystem.out aus.
    - Mit keinen Argumenten
    - Mit dem Argument j10
    - Mit den beiden Argumenten j5 und j2

```
3 public class Box {  
4     double length;  
5  
6     public Box(double length) {  
7         this.length = length;  
8     }  
9  
10    public double size() {  
11        return length;  
12    }  
13  
14    public double size(double width) {  
15        return this.length * width;  
16    }  
17 }
```

```

18     public double size(double num, double f) {
19         return this.length - num * f;
20     }
21
22     public static void main(String[] args) {
23         Box example = new Box(15);
24
25         double mysize = example.size();
26         System.out.println(mysize);
27
28         mysize = example.size(10);
29         System.out.println(mysize);
30
31         mysize = example.size(5, 2);
32         System.out.println(mysize);
33     }
34 }
35

```

github: raw

(b) Notieren Sie die Ausgabe der jmain-Methode.

```

- 15
- 150
- 5

```

## Aufgabe 5

Die folgende Abbildung stellt den Entwurf der Implementierung einer verketteten Liste dar, welche Integer-Werte als Elemente enthalten kann.

Die Klasse `MyList` stellt die Methode `getSum()` zur Verfügung, welche die Summe über alle in einer Liste befindlichen Elemente berechnet. Ein Ausschnitt der Implementierung sieht folgendermaßen aus:

Gehen Sie im Folgenden davon aus, dass bereits Methoden existieren, welche Elemente in die Liste einfügen können.

- (a) Implementieren Sie in einer objektorientierten Programmiersprache Ihrer Wahl, z. B. Java, die Methode `calculateSum()` der Klassen `MyEndElement` und `MyRegularElement`, so dass rekursiv die Summe der Elemente der Liste berechnet wird. Als Abbruchbedingung darf hierbei nicht das Feld `MyRegularElement.next` auf den Wert `null` überprüft werden.

Hinweis: Gehen Sie davon aus, die Implementierung von `MyList` garantiert, dass `MyRegularElement.next` niemals den Wert `null` annimmt, sondern das letzte hinzugefügte `MyRegularElement` auf eine Instanz der Klasse `MyEndElement` verweist. Es gibt immer nur eine Instanz der Klasse `MyEndElement` in einer Liste.

Hinweis: Achten Sie auf die Angabe einer korrekten Methodensignatur.

```

3     public class MyElement {
4
5         MyElement next;

```

```

6
7     int value;
8
9     public MyElement(int value) {
10         this.value = value;
11     }
12
13     MyElement add(int value) {
14         next = this.next.add(value);
15         return this;
16     }
17
18     int calculateSum() {
19         return value + next.calculateSum();
20     }
21 }

```

github: raw

```

3     public class MyEndElement extends MyElement {
4
5         public MyEndElement () {
6             super(0);
7         }
8
9         MyElement add(int value) {
10             MyElement element = new MyElement(value);
11             element.next = this;
12             return element;
13         }
14
15         int calculateSum() {
16             return 0;
17         }
18
19     }

```

github: raw

```

3     public class MyList {
4         private MyElement head;
5
6         public MyList() {
7             this.head = new MyEndElement();
8         }
9
10        public int getSum() {
11            return this.head.calculateSum();
12        }
13
14        public void add(int value) {
15            head = head.add(value);
16        }
17
18        public static void main(String[] args) {
19            MyList myList = new MyList();
20            myList.add(1);
21            myList.add(2);
22            myList.add(3);
23            System.out.println(myList.getSum());

```

```
24     }  
25 }
```

github: raw

- (b) Nennen Sie den Namen des Entwurfsmusters, auf welchem das oben gegebene Klassendiagramm basiert, und ordnen Sie dieses in eine der Kategorien von Entwurfsmustern ein.

Hinweis: Es genügt die Angabe eines Musters, falls Sie mehrere Muster identifizieren sollten.

Kompositium (Strukturmuster)

## Aufgabe 6

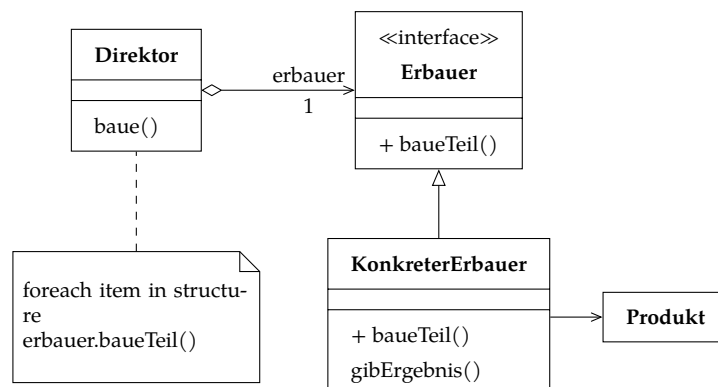
- (a) Erläutern Sie den Zweck (Intent) des Erzeugungsmusters Erbauer in max. drei Sätzen, ohne dabei auf die technische Umsetzung einzugehen.

Die Erzeugung komplexer Objekte wird vereinfacht, indem der Konstruktionsprozess in eine spezielle Klasse verlagert wird. Er wird so von der Repräsentation getrennt und kann sehr unterschiedliche Repräsentationen zurückliefern.

- (b) Erklären Sie, wie das Erzeugungsmuster Erbauer umgesetzt werden kann (Implementierung). Die Angabe von Code ist hierbei NICHT notwendig!
- (c) Nennen Sie jeweils einen Vor- und einen Nachteil des Erzeugungsmusters Erbauer im Vergleich zu einer Implementierung ohne dieses Muster.

**Vorteil** Die Implementierungen der Konstruktion und der Repräsentationen werden isoliert. Die Erbauer verstecken ihre interne Repräsentation vor dem Direktor.

**Nachteil** Es besteht eine enge Kopplung zwischen Produkt, konkretem Erbauer und den am Konstruktionsprozess beteiligten Klassen.



**Erbauer** Der Erbauer spezifiziert eine abstrakte Schnittstelle zur Erzeugung der Teile eines komplexen Objektes.

**KonkreterErbauer** Der konkrete Erbauer erzeugt die Teile des komplexen Objekts durch Implementierung der Schnittstelle. Außerdem definiert und verwaltet er die von ihm erzeugte Repräsentation des Produkts. Er bietet auch eine Schnittstelle zum Auslesen des Produkts.

**Direktor** Der Direktor konstruiert ein komplexes Objekt unter Verwendung der Schnittstelle des Erbauers. Der Direktor arbeitet eng mit dem Erbauer zusammen: Er weiß, welche Baureihenfolge der Erbauer verträgt oder benötigt. Der Direktor entkoppelt somit den Konstruktionsablauf vom Klienten.

**Produkt** Das Produkt repräsentiert das zu konstruierende komplexe Objekt.

## Aufgabe

Lesen Sie die folgenden alternativen Codestücke.

```

(a) public class MyParser {
    2     private InputStream input;
    3
    4     public MyParser(String filePath) {
    5
    6     }
    7 }

(b) public class MyParser {
    2     private InputStream input;
    3
    4     public MyParser(InputStream stream) {
    5     }
    6 }

```

Beide Codestücke zeigen die Initialisierung einer Klasse namens `MyParser`. Das zweite Codestück nutzt jedoch hierfür eine Technik namens Abhängigkeits-Injektion (Dependency Injection).

- (a) Erklären Sie den Unterschied zwischen beiden Initialisierungen. Hinweis: Sie können diese Aufgabe auch lösen, falls Sie die Technik nicht kennen.



Die Abhängigkeit von einer Instanz der Klasse InputStream wird erst bei der Initialisierung des Objekts übergeben.

(b) Benennen Sie einen Vorteil dieser Technik.

Die Kopplung zwischen einer Klasse und ihrer Abhängigkeit wird verringert.

## Aufgabe 8

Das Client-Server-Modell ist ein Architekturmuster. Nennen Sie zwei Vorteile einer nach diesem Muster gestalteten Architektur.

- (a) Einfache Integration weiterer Clients
- (b) Prinzipiell uneingeschränkte Anzahl der Clients <sup>a</sup>
- (c) Es muss nur ein Server gewartet werden. Dies gilt z. B. für Updates, die einmalig und zentral auf dem Server durchgeführt werden und danach für alle Clients verfügbar sind. <sup>b</sup>

<sup>a</sup><https://www.karteikarte.com/card/164928/vorteile-und-nachteile-des-client-server-modells>

<sup>b</sup><https://www.eoda.de/wissen/blog/client-server-architekturen-performance-und-agilitaet-fuer-data-science/>

## Aufgabe 9

Betrachten Sie die folgende Liste von Technologien:

- Nodejs
- PHP
- CSS
- AJAX
- Python
- Java

Welche dieser Technologien laufen in einem Client-Server-System üblicherweise auf der Seite des Klienten und welche auf der Seite des Servers? Nehmen Sie hierzu an, dass der Client ein Browser ist.

Übertragen Sie die im Folgenden gegebene Tabelle in Ihren Bearbeitungsbogen und ordnen Sie die aufgelisteten Technologien anhand der Buchstaben in die Tabelle ein. Fortsetzung nächste Seite!

Hinweis: Mehrfachzuordnungen sind möglich.

Client-seitige Technologien	Server-seitige Technologien
CSS	Nodejs
AJAX	PHP
Python	Python
Java	Java

## Aufgabe 10

- (a) Was bedeutet die Abkürzung AJAX?

Asynchronous JavaScript and XML

- (b) Erklären Sie in max. drei Sätzen die grundlegende Funktion von AJAX.

Konzept der asynchronen Datenübertragung zwischen einem Browser und dem Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, und die Seite zu verändern, ohne sie komplett neu zu laden. <sup>a</sup>

<sup>a</sup>[https://de.wikipedia.org/wiki/Ajax\\_\(Programmierung\)](https://de.wikipedia.org/wiki/Ajax_(Programmierung))

## Aufgabe 11

- (a) Was ist das Hypertext Transfer Protocol (HTTP) und wozu dient es?

Das Hypertext Transfer Protocol ist ein zustandsloses Protokoll zur Übertragung von Daten auf der Anwendungsschicht über ein Rechnernetz. Es wird hauptsächlich eingesetzt, um Webseiten (Hypertext-Dokumente) aus dem World Wide Web (WWW) in einen Webbrowser zu laden. Es ist jedoch nicht prinzipiell darauf beschränkt und auch als allgemeines Dateiübertragungsprotokoll sehr verbreitet. <sup>a</sup>

<sup>a</sup>[https://de.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

- (b) Betrachten Sie die folgende Zeile Text. Um welche Art von Text handelt es sich?

`https://developer.mozilla.org/en-US/search?q=client+servertooverview`

Es handelt sich um eine HTTP-URL (Uniform Resource Locator). Die URL lokalisiert eine Ressource, beispielsweise eine Webseite, über die zu verwendende Zugriffsmethode (zum Beispiel das verwendete Netzwerkprotokoll wie HTTP oder FTP) und den Ort (engl. location) der Ressource in Computernetzwerken. <sup>a</sup>

<sup>a</sup>[https://de.wikipedia.org/wiki/Uniform\\_Resource\\_Locator](https://de.wikipedia.org/wiki/Uniform_Resource_Locator)

- (c) Was sind die vier wesentlichen Bestandteile des Texts aus der vorigen Teilaufgabe?

**Schema** https://  
**Host** developer.mozilla.org  
**Pfad** /en-US/search  
**Query** ?q=client+servertooverview

<sup>a</sup>

<sup>a</sup>[https://de.wikipedia.org/wiki/Uniform\\_Resource\\_Locator](https://de.wikipedia.org/wiki/Uniform_Resource_Locator)

## Aufgabe 12

Es gibt Softwaresysteme, welche auf peer-to-peer (P2P) Kommunikation basieren und eine entsprechende Architektur aufweisen.

- (a) Bewerten Sie die folgenden Aussagen als entweder richtig oder falsch.

- (i) Mithilfe des Befehls “lookup” können Peers sich gegenseitig identifizieren.

richtig

- (ii) In einem P2P-System, wie auch bei Client-Server, sind alle Netzwerkteilnehmer gleichberechtigt.

falsch. Im Client-Servermodell sind nicht alle Netzwerkteilnehmer gleichberechtigt. Der Server hat mehr Privilegien wie der Client.

- (iii) Alle P2P-Systeme funktionieren grundsätzlich ohne einen zentralen Verwaltungs-Peer.

falsch. Es gibt zentralisierte P2P-Systeme (Beispiel: Napster), welche einen zentralen Server zur Verwaltung benötigen, um zu funktionieren. <sup>a</sup>

<sup>a</sup><https://de.wikipedia.org/wiki/Peer-to-Peer>

- (iv) P2P kann auch für eine Rechner-Rechner-Verbindung stehen.

richtig

- (v) Es gibt strukturierte und unstrukturierte P2P-Systeme. In unstrukturierten P2P-Systemen wird zum Auffinden von Peers eine verteilte Hashtabelle verwendet (DHT).

richtig

- (vi) In einem P2P-System sind theoretisch alle Peers gleichberechtigt, praktisch gibt es jedoch leistungsabhängige Gruppierungen.

richtig

- (vii) Ein Peer kann sowohl ein Client wie auch ein Server für einen anderen Peer sein.

richtig

- (b) Wählen Sie zwei falsche Aussagen aus der vorherigen Tabelle aus und berichtigen Sie diese in jeweils einem Satz.

Sie oben.