

Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2021 / Frühjahr

## Thema 2 / Teilaufgabe 1 / Aufgabe 5

(Webshop)

**Stichwörter:** Implementierung in Java, Sequenzdiagramm

(a) Nennen Sie vier Programmierparadigmen.

Lösungsvorschlag

- Imperative Programmierung
- Prozedurale Programmierung
- Funktionale Programmierung
- Objektorientierte Programmierung

<sup>a</sup>

<sup>a</sup>[https://de.wikipedia.org/wiki/Programmierparadigma#Strukturierte\\_Programmierung](https://de.wikipedia.org/wiki/Programmierparadigma#Strukturierte_Programmierung)

(b) Erläutern Sie die Begriffe *Overloading* und *Overriding*, sowie deren Unterschiede.

Lösungsvorschlag

- Beim Überladen muss die Methode eine andere Signatur haben, beim Überschreiben dieselbe Signatur.
- Die Intention beim Überladen ist, Methode zu erweitern, beim Überschreiben die Methode vollständig zu ersetzen.
- Überladen der Methode wird verwendet, um den Polymorphismus der Kompilierzeit zu erreichen. Das Überschreiben der Methode wird verwendet, um einen Laufzeit-Polymorphismus zu erreichen.  
Bei der Methodenüberladung weiß der Compiler, welches Objekt welcher Klasse zum Zeitpunkt der Kompilierung zugewiesen wurde. In der Methodenüberschreibung sind diese Informationen jedoch erst zur Laufzeit bekannt.
- Das Überladen von Methoden findet in derselben Klasse statt, während das Überschreiben in einer von einer Basisklasse abgeleiteten Klasse stattfindet.

<sup>a</sup>

<sup>a</sup><https://gadget-info.com/difference-between-method-overloading>

(c) Erläutern Sie, wie sich zentrale und dezentrale Versionsverwaltung unterscheiden.

Lösungsvorschlag

Beim der dezentralen Versionsverwaltung hat jede/r EntwicklerIn das komplette Repository mit seiner kompletten History lokal gespeichert und kann diese dann mit anderen Repositories abgleichen.

Bei der zentralen Versionsverwaltung gibt es einen zentralen Server, der die kom-

plette History vorhält.

(d) Erstellen Sie ein Sequenzdiagramm zur Methode `main` der Klasse `Webshop`.

Hinweise:

- Arithmetische Operationen müssen nicht weiter aufgelöst werden.
- Listenoperationen müssen nicht explizit dargestellt werden.
- Auf das Zeichnen einer passiven Lebenslinie muss nicht geachtet werden.
- Übertragen Sie das untenstehende Diagramm als Ausgangspunkt in Ihren Bearbeitungsbogen.

```
public class Webshop {
    public static void main(String[] args) {
        Bestellung b1 = new Bestellung();

        // ab hier soll modelliert werden
        Artikel a1 = new Artikel();
        a1.setName("Taschenrechner");
        a1.setPrice(10);

        b1.addArticle(a1);

        Bestellung b2 = new Bestellung();
        Artikel a2 = new Artikel();
        a2.setName("Lineal");
        a2.setPrice(2.5);

        Artikel a3 = new Artikel();
        a3.setName("Bleistift");
        a3.setPrice(0.7);

        b2.addArticle(a3);
        b1.addArticle(a2);

        b1.getSize();

        b2.getPrice();
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2021/fruehjahr/webshop/Webshop.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/webshop/Webshop.java)

```
@SuppressWarnings({"unused"})
public class Artikel {
    private String name;

    private double price;

    public void setName(String name) {
        this.name = name;
    }
}
```

```
public void setPrice(double price) {
    this.price = price;
}

public double getPrice() {
    return price;
}
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2021/fruehjahr/webshop/Artikel.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/webshop/Artikel.java)

```
import java.util.ArrayList;
import java.util.List;

public class Bestellung {
    private List<Artikel> articles;
    // Anzahl an Artikeln
    private int size = 0;
    // Gesamtpreis der Bestellung
    private double price = 0;

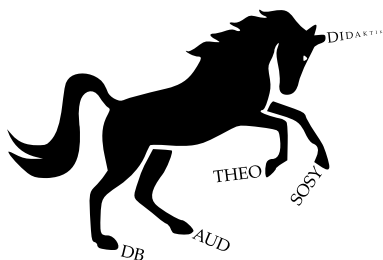
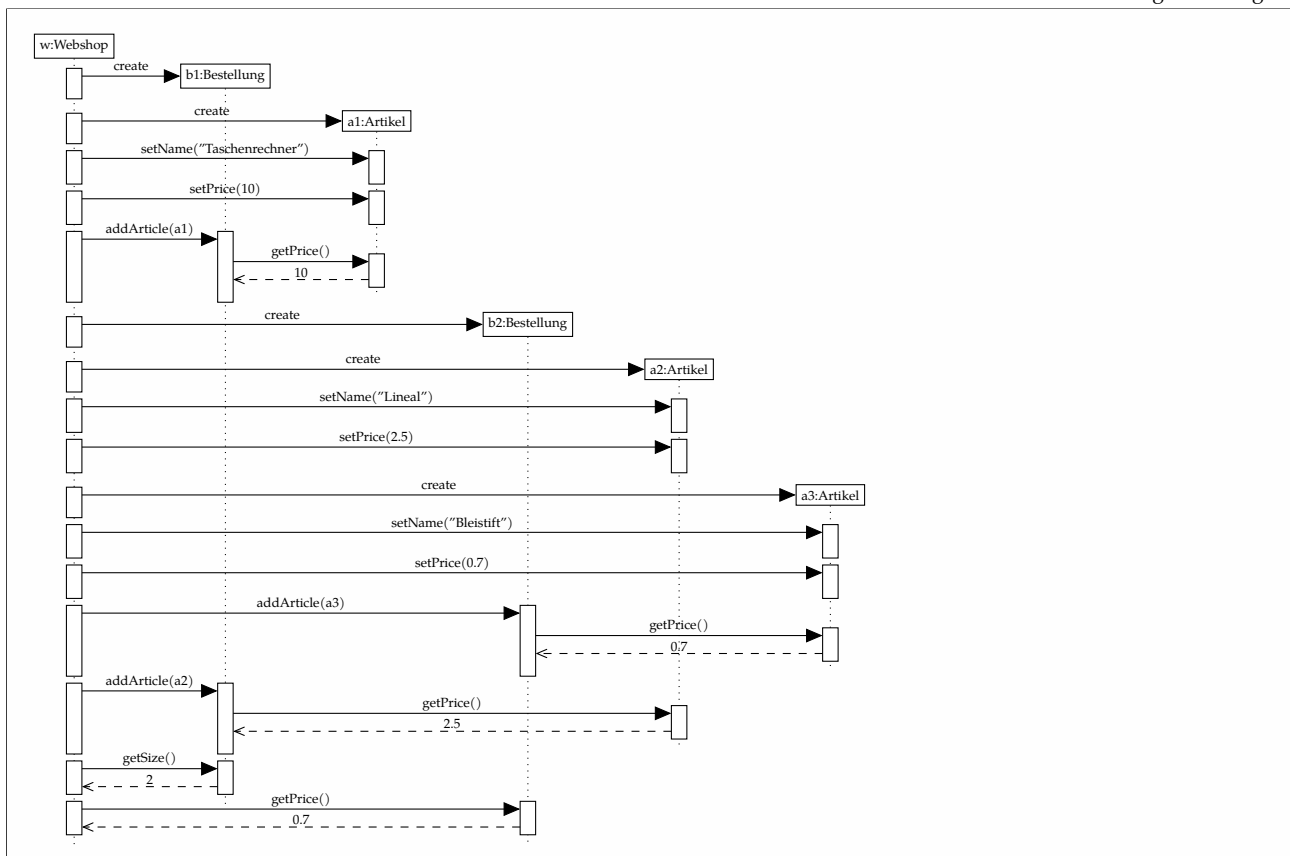
    public Bestellung() {
        articles = new ArrayList<>();
    }

    public void addArticle(Artikel article) {
        // muss nicht weiter aufgelöst werden, siehe Hinweise
        articles.add(article);
        size++;
        // muss nicht weiter aufgelöst werden, siehe Hinweise
        price = article.getPrice() + price;
    }

    public int getSize() {
        return size;
    }

    public double getPrice() {
        return price;
    }
}
```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/examen/examen\\_66116/jahr\\_2021/fruehjahr/webshop/Bestellung.java](https://github.com/bschlangaul/examen/examen_66116/jahr_2021/fruehjahr/webshop/Bestellung.java)



## 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](mailto: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-2/Teilaufgabe-1/Aufgabe-5.tex>