

I - Klausur Fortgeschrittene Programmierungsmethoden

26 Jan 2024

Kandidat/in:

Name: _____

Allgemeine Hinweise

- Neben Papier und Schreibutensilien sind keine weiteren Hilfsmittel erlaubt.
- Verwenden Sie keine roten Stifte und keine Bleistifte.
- Sie dürfen alle Aufgaben in beliebiger Reihenfolge lösen, aber konzentrieren Sie sich jeweils auf eine Aufgabe, aber teilen Sie sich Ihre Zeit ein.
- Alle Mobiltelefone müssen vollständig ausgeschaltet sein.
- Vergessen Sie nicht, Ihren Namen auf jedes Blatt zu schreiben.
- Blätter ohne diese Angaben werden nicht bewertet.
- Bitte schreiben Sie in Ihrem eigenen Interesse deutlich. Unverständliche oder nicht begründete Antworten werden nicht bewertet.
- In den letzten 15 Minuten der Prüfung kann der Raum nicht mehr verlassen werden.
- Im Fall von Täuschungsversuchen wird die Klausur sofort mit 0 Punkten bewertet. Eine Vorwarnung erfolgt nicht.

Aufgabe	max. Punkte	erreicht
1-A	15 Punkte	
1-B	10 Punkte	
2-A	10 Punkte	
2-B	20 Punkte	
3-A	10 Punkte	
3-B	25 Punkte	
Summe	90 + 10 e.o. = 100 Punkte	

Aufgabe 1

- A. Beschreiben Sie die zentrale Idee des **Interface-Segregation-Prinzip**. Klären Sie mit einem Code-Beispiel aus der realen Welt ab. **(15 Punkte)**
- B. Was ist eine **Annotation**? Geben Sie **zwei** Beispiele aus der Standard-Java-Bibliothek. Wie kann man eine **Custom-Annotation** in **Java** umsetzen? Gibt es einen Link zwischen Annotations und Reflection? Klären Sie mit Theorie und Code-Beispielen ab. **(10 Punkte)**

Aufgabe 2

- A. Erläutern Sie anhand von konkreten Code-Beispielen den Unterschied zwischen **Komposition** und **Vererbung** in Java. **(10 Punkte)**
- B. Eine Buch-Klasse besitzt die folgenden Attributen:
- private String name;
 - private String editor;
 - private int preis;
 - private List<String> autoren;

1. Erstellen Sie die Buch-Klasse und eine Liste von Büchern. **(5 Punkte)**
2. Implementieren Sie eine Java 8 Stream Funktionalität, die
 - eine MwSt von 15% addiert, zu jedem Buch das genau vier Autoren hat;
 - den Mittelpreis für die ersten 10 Bücher berechnet;
 - das Ergebnis auf dem Bildschirm ausgibt. **(15 Punkte)**

For/While-Schleife und Rekursion sind nicht erlaubt!

Aufgabe 3

- A. Was gibt der folgende Code aus? **(10 Punkte)**

Code-Fragment

```
class S{
    public static void main(String [] args){
        String s = "Java";
        s.concat(" SE 6");
        s.replace('6','7');
        System.out.print(s);
    }
}
```

- B. Ein Online-Shop möchte eine Liste der zum Verkauf verfügbaren Fahrzeuge anzeigen. Alle Fahrzeuge sind in einer Liste vom Typ "ArrayList<Fahrzeug>" gespeichert und haben als Attribute eine *Marke*, ein *Modell* und einen *Kilometerstand*. *Marke* und *Modell* dürfen nicht geändert werden. Der *Kilometerstand* kann nur um 1 inkrementiert werden.

Es gibt nur zwei aktuelle Fahrzeugtypen: **Verbrennungsfahrzeug** und **Elektrofahrzeug**. Ein **Verbrennungsfahrzeug** hat ein veränderbares "HorsePower"-Attribut und ein **Elektrofahrzeug** hat ein veränderbares Attribut "Kilowatt". Eine Methode "display()" soll bereitgestellt sein, die alle Fahrzeuge der Liste anzeigt. Jedes Fahrzeug "weiß", wie es auf dem Bildschirm angezeigt werden soll. **Verbrennungsfahrzeug** wird anzeigen: "Thermal: <Marke> <Modell> <Km> <HorsePower>", wobei z. B. "<Marke>" ein Platzhalter für die Marke des Fahrzeugs ist. **Elektrofahrzeug** wird anzeigen: "EV: <Marke> <Modell> <Kilowatt>".

1. Implementieren Sie alle benötigten Klassen. **(10 Punkte)**
2. Verwenden Sie das Singleton-Muster, um sicherzustellen, dass es nur eine Instanz von "Fahrzeug" mit demselben Marke und Modell in der Liste gibt. **(10 Punkte)**
3. Stellen Sie die Endlösung in Form eines UML-Diagramms grafisch dar. **(5 Punkte)**