

Übungsblatt 5: Klassen und UML

Deadline: 13.11.2025 12:00 Uhr

Besprechung: 17.11.–21.11.2025

Hinweise für die Bearbeitung der Aufgaben

- Bitte lesen Sie die Aufgabenstellungen genau durch.
- Testen Sie Ihre Lösungen vor der Abgabe selbst!
- Halten Sie sich an die Java Coding Konventionen (siehe letzte Seite).
- Damit die automatischen Tests korrekte Ergebnisse liefern können, achten Sie bitte auf die folgenden Punkte:
 - Halten Sie sich bitte genau an die Vorgaben/Vorlagen aus der Aufgabenstellung.
 - Wenn Sie Umlaute verwenden, stellen Sie die Kodierung Ihrer Dateien unbedingt auf UTF-8.
 - Achten Sie darauf, dass Ihr Projekt mit Java Version 21 kompiliert und ausgeführt werden kann.
 - Ihre Java-Klassen müssen sich im default package befinden (es darf kein package angegeben werden).



Aufgabe 1 Abgabe über GATE
Konzepte der objektorientierten Programmierung

Ordnen Sie die vier Erklärungen den vier Begriffen zu.

-
1. Verallgemeinerte Sammlung von Eigenschaften in einer Art Bauplan.
 2. Eigenschaften einer Instanz. In Java ist einem/r [Begriff] stets ein eindeutiger Datentyp zugeordnet.
 3. Operationen, die auf einer Instanz einer Klasse ausgeführt werden können.
 4. Eine konkrete Instanz einer Klasse
-

- A. Attribut
 - B. Methode
 - C. Klasse
 - D. Objekt
-

Aufgabe 2 Abgabe über GATE mit Peer-Review
UML Diagramm zu Java-Code

- a) Erstellen Sie ein UML Diagramm zu folgendem Text:

Jedes Quadrat hat eine Seitenlänge. Außerdem kann es eine Linienfarbe und eine Füllfarbe haben. (Hierfür können Sie den Datentyp String benutzen.) Sehen Sie sinnvolle Konstruktoren, sowie die get- und set-Methoden für die Attribute einer Klasse vor. Weiter soll es eine Methode geben, die den Flächeninhalt zurückgibt.

Jeder Kreis hat einen Radius. Außerdem kann er eine Linienfarbe und eine Füllfarbe haben. Abgesehen von den Konstruktoren und den get- und set-Methoden für die Attribute hat die Klasse auch eine Methode, die den Kreisumfang zurück gibt.

Laden Sie Ihr UML-Diagramm als jpg/png/pdf-Datei hoch.

- b) Schreiben Sie zu Ihrem erstellten UML Klassendiagramm zur Klasse für das Quadrat entsprechenden Java-Code. Füllen Sie die Methoden mit sinnvollem Inhalt (Die get- und set-Methoden die Attributwerte zurückliefern bzw. setzen, die Methode für den Flächeninhalt gibt wirklich den Flächeninhalt des Quadrates zurück und auch die Konstruktoren belegen die Attribute mit entsprechenden Werten).

Schreiben Sie außerdem eine Klasse `SquareApplication`, die in der `main`-Methode eine Instanz des Quadrats erzeugt mit den Seitenlänge 3,3, der Linielfarbe "black" und der Füllfarbe "white".

Aufgabe 3 Abgabe über GATE mit Tests

Bruchrechnung

Schreiben Sie eine Java-Klasse `Fraction`, die einen Bruch repräsentiert. Sie soll einen Zähler (`numerator`) und einen Nenner (`denominator`) jeweils vom Datentyp `int` besitzen.

Diese Klasse soll zwei Konstruktoren haben:

- Initialisierung eines Bruchs mit einem Zähler und Nenner:
`Fraction(int numerator, int denominator)`
- Initialisierung eines Bruchs, der eine ganze Zahl repräsentiert:
`Fraction(int n)`

Sollte als Nenner die Zahl 0 übergeben werden, so setzen Sie diesen auf 1.

Die Klasse soll außerdem folgende Methoden anbieten:

- Den Bruch mit einem anderen Bruch multiplizieren:
`void multiply(Fraction factor)`
- Den Bruch mit einem anderen Bruch addieren:
`void add(Fraction summand)`
- Den Bruch auszugeben:
`String toString()`
z. B. "3/6" (ohne Anführungszeichen)
- Den Wert des Bruchs als `double` zurückgeben:
`double asDouble()`

Das Implementieren des Kürzens ist nicht notwendig. Es gilt:

$$\frac{m_1}{n_1} + \frac{m_2}{n_2} = \frac{m_1 n_2 + m_2 n_1}{n_1 n_2}, \quad \frac{m_1}{n_1} \times \frac{m_2}{n_2} = \frac{m_1 m_2}{n_1 n_2}$$

Aufgabe 4 Abgabe in GATE Quiz

Welche der folgenden Aussagen sind für Java korrekt?

1. Jedes Java-Programm benötigt eine **public void** `main(String args)`-Methode
2. Jede Java-Klasse benötigt eine `main`-Methode.
3. Man kann den Inhalt von zwei Arrays `a` und `b` mit "`a == b`" vergleichen.
4. Arrays können aus Verbünden gebildet werden.
5. Arrays und Verbünde sind Referenzdatentypen.
6. Die Länge eines Arrays kann in Java zur Laufzeit beliebig verändert werden.
7. Jede Methode muss etwas auf der Konsole ausgeben.

Aufgabe 5 Abgabe in GATE Code lesen und verstehen

In einem Start-Up Unternehmen für wiederaufbereitete Hardwareprodukte (Smartphones, Tablets und Laptops) wird ein Programm zur Verwaltung der Umsatzzahlen entwickelt. Es werden die Umsätze für die ersten Monate gespeichert. Beschreiben Sie in eigenen Worten, was das folgende Programm macht.

Abgabe

Die Abgabe erfolgt bis zum 13.11.2025 12:00 Uhr auf GATE, <https://gate.ifi.lmu.de>.

Zuletzt aktualisiert: 27. Oktober 2025

```
public class VerkaufsAnalyse {
    public static void main(String[] args) {
        String[] month = { "Januar", "Februar", "März" };
        String[] product = {"Smartphone", "Tablet", "Laptop"};
        double[][] numbers = {
            {1200, 950, 800},
            {1500, 1050, 950},
            {1300, 900, 1100}};
        double result = 0;
        for (int i = 0; i < numbers.length; i++) {
            double temp = 0;
            for (int j = 0; j < numbers[i].length; j++) {
                temp += numbers[i][j];
            }
            if (temp > result) {
                result = temp;
                index = i;
            }
        }
        System.out.println(month[index] + ":" + result);
    }
}
```

Abbildung 1: Programmcode für VerkaufsAnalyse.java

Java Coding Konventionen

Zur Erstellung eines lesbaren und wartbaren Codes sind Coding Conventions sehr hilfreich. Im Folgenden sind einige der wichtigsten Konventionen für Java aufgeführt:

- Achten Sie auf gute Lesbarkeit Ihrer Programme und rücken Sie Anweisungen immer entsprechend ein (üblich sind 2 oder 4 Leerzeichen bzw. ein Tabulator pro Ebene).
- Kommentare sollten verwendet werden, um den Code zu erklären an den Stellen, an denen Ihre Algorithmus-Idee nicht offensichtlich ist und nicht um ihn zu wiederholen.
- Leerzeichen sollten verwendet werden, um den Code lesbarer zu machen (z. B. nach Kommas, um Operatoren herum). Nutzen Sie diese auch vor und nach Klammern (z. B. `if (condition) { ... }`).

- Java ist eine Case-sensitive Sprache. `Variable`, `variable` und `VARIABLE` sind drei verschiedene Bezeichner.
- Klassen beginnen mit einem Großbuchstaben, z. B. `HelloWorld`.
- Methoden und Variablen beginnen mit einem Kleinbuchstaben, z. B. `main`, `firstName`.
- Mehrteilige Bezeichner werden im camelCase-Stil geschrieben, z. B. `firstName`, `calculateSum`.
- Konstanten werden in Großbuchstaben mit Unterstrichen geschrieben, z. B. `MAX_VALUE`.
- Verwenden Sie verständliche bzw. sprechende Bezeichner für Variablen und Methoden.
- Verwenden Sie geschweifte Klammern auch für einzeilige Blöcke.
- Leerzeilen verbessern die Lesbarkeit, indem sie logisch zusammenhängende Abschnitte des Codes visuell trennen, z. B. zwischen Methoden, Definition von Klassen und Interface, lokalen Variablen in einer Methode und der ersten Anweisung.

Weitere Informationen finden Sie in den offiziellen Java Coding Conventions: <https://www.oracle.com/java/technologies/javase/codeconventions-contents.html>