

# Übungsblatt 04

Programmierung und Softwareentwicklung (WS 2024/2025)

Abgabe: Fr. 15.11.2024, 23:59 Uhr — Besprechung: KW 46

- Bitte lösen Sie die Übungsaufgabe in **Gruppen von 2 Studierenden**.
- Dieses Übungsblatt besteht aus zwei Teilen (A, B). Teil A ist in der Präsenzübung zu lösen. Teil B ist in Heimarbeit (Gruppe von 2 Studierenden) zu lösen und rechtzeitig abzugeben. Die Abgabe erfolgt über ILIAS.
- Geben Sie `.java`-Dateien nur im UTF-8 Encoding ab. Ändern Sie ggf. das Textdateiencoding auf UTF-8 ab, bevor Sie die Unterlagen herunterladen. Abhängig von Ihrem Betriebssystem müssen Sie möglicherweise auch nichts tun.
- Geben Sie zu Beginn der Dateien Ihre Namen (Vor- und Nachname), die Matrikelnummern und die E-Mail-Adressen an. Nutzen Sie bei Java-Dateien die korrekte JavaDoc-Syntax.
- Benennen Sie die Dateien nach dem folgenden Schema:
  1. **PSE[ÜB-Nr]-[Nachnamen der Teammitglieder]-[Nachname des Tutors].pdf**  
Beispiel: PSE04-StießSpethKrieger-Becker.pdf
  2. **[Klassenname].java**: Alle von Ihnen bearbeiteten Java-Dateien, die Lösungen für die Aufgaben enthalten.
- Missachtung der formalen Kriterien kann dazu führen, dass einzelne Aufgaben oder die gesamte Abgabe mit 0 Punkten bewertet werden.

**Lernziel:** Auf diesem Übungsblatt werden Sie Ihr Wissen über Klassen, Objekte und Referenzen, Objekterstellung sowie logische Ausdrücke vertiefen. Des Weiteren werden Sie den Umgang mit Fehlertypen und ASTs und das Erstellen von Dokumentation weiter üben.

**Punkte:** Dieses Übungsblatt enthält zwei Teile. In Teil B können Sie bis zu 50 Punkte und mehrere Bonuspunkte erzielen. Zum Bestehen des Blatts benötigen Sie mindestens 25.0 Punkte.

**Style:** Bitte halten Sie die in der Vorlesung vorgestellten Style-Regeln ein. Dazu gehören auch JavaDoc sowie Vor- und Nachbedingungen. Der Style Ihrer Implementierung wird mit bis zu 50% bewertet.

**Vorbereitung:** Bitte erledigen Sie die folgenden Schritte **vor** der Präsenzübung.

- Importieren Sie das zu diesem Übungsblatt gehörende Maven Projekt in Ihre IDE. Sie finden das Maven Projekt in unserem Git-Repository: <https://github.com/SQA-PSE-WS-2024-2025/exercise-sheet-04>
- Stellen Sie sicher, dass Sie Übungsblatt 03 absolviert haben, sowie alle Software installiert und funktionsfähig ist (IDE (Eclipse, IntelliJ, VSCode,...) und Java 21).

**Unterlagen:**

- Git-Repositories: <https://github.com/SQA-PSE-WS-2024-2025/>
- Dokumentation des Hamstersimulators: <https://tinyurl.com/5yx654w8>

**Scheinkriterien:** Durch die Teilnahme am Übungsbetrieb können Sie sich für die Teilnahme an der Klausur qualifizieren:

- Bestehen von min. 80% aller Übungsblätter.
- Ein Übungsblatt gilt als bestanden, wenn 50% der Punkte des abgegebenen Heimarbeitsteils erreicht wurden.
- Aktive Teilnahme an min. 80% der Übungen.

**Viel Erfolg!**

# 1 Teil A - Präsenzaufgaben

## Aufgabe 1 Objekterstellung und -referenzen I

In der folgenden Aufgabe geht es um Ihr Wissen im Bezug auf Objekte, Objekterstellung und Objektreferenzen.

- (a) Erörtern Sie mit ihrem Team, was die folgenden Begriffe bedeuten: *Allokieren*, *Initialisieren* und *Zuweisen*. Schreiben Sie Ihre Ergebnisse auf.
- (b) Erörtern Sie mit ihrem Team, was die folgenden Begriffe bedeuten: *Bezeichner*, *Entität*, *Referenz*, *Objekt*. Schreiben Sie Ihre Ergebnisse auf.
- (c) Betrachten Sie den in Listing 1 gegebenen Code Ausschnitt. Nutzen Sie die Begriffe aus Teilaufgabe (a) und (b), um den Code Ausschnitt zu beschreiben. Was passiert in Zeile 1 des gegebenen Ausschnitts, was passiert in Zeile 2? Worin unterscheiden sich die Zeilen und warum kann dieser Code Probleme bereiten?

```
1 Hamster adam = new Hamster();
2 Hamster eve = new Hamster(game.getTerritory(), new Location(2, 2), Direction.EAST, 42);
```

Listing 1: Codeausschnitt.

## Aufgabe 2 Objekte erstellen

In dieser Aufgabe erstellen Sie eigene Objekte. Öffnen Sie dazu die Klasse `PaulesFriendsGame` und lösen Sie die folgenden Teilaufgaben in der Operation `createPaulesFriends`.

- (a) Erzeugen Sie zwei neue Hamster-Objekte für die Hamster *Ronnie* und *Paula*. Erzeugen Sie das Hamster-Objekt für Ronnie in der rechten unteren Ecke des Territoriums, und das für Paula in der linken unteren Ecke des Territoriums. Wählen Sie jeweils unterschiedliche Blickrichtungen und unterschiedliche Anzahlen an Körnern aus. Speichern Sie die Werte für die benötigten Argumente zunächst in den im Code vorgegebenen lokalen Variablen (siehe Abschnitt „Variablen: Deklaration und Zuweisung“ im Foliensatz zu Referenzen und Objekterzeugung).

Nutzen Sie die Dokumentation des Hamstersimulators, um die benötigten Schnittstellen nachzuschlagen, z.B. wie Sie Zugriff auf das Territoriums-Objekt erhalten.

*Hinweis:* Das vordefinierte Objekt `game` repräsentiert das aktuelle Spiel. Sie können auf diesem Objekt alle Operationen der Klasse `HamsterGame` aufrufen, unter anderem eine Abfrage, die Ihnen das aktuelle Territorium zurück gibt.

*Hinweis:* Der Typ `Direction` ist ein Aufzählungstyp und repräsentiert somit eine endliche, abgeschlossenen Menge von Konstanten. Sie können z.B. mit `Direction.SOUTH` auf die Konstante, die den Süden repräsentiert zugreifen.

## Aufgabe 3 Objekterstellung und -referenzen II

In dieser Aufgabe geht es um Objekte, Objekterstellung und Objektreferenzen. Schauen Sie sich den Code in Listing 2 an. Nach der Ausführung der ersten fünf Codezeilen befinden sich Paula und Ronnie, wie in Abbildung 1 gezeigt, auf dem Territorium.

```
1 Location locationLeftTop = new Location(0,0);
2 Location locationLeftBottom = new Location(7,0);
3
4 Hamster paula = new Hamster(game.getTerritory(), locationLeftBottom, Direction.EAST, 2);
5 Hamster ronnie = new Hamster(game.getTerritory(), locationLeftTop, Direction.EAST, 2);
6
7 paula.putGrain();
8 ronnie.putGrain();
9 ronnie.putGrain();
10
11 Hamster paule = paula;
12 paule.move();
13 paule.move();
14
15 Hamster ronja = ronnie;
16 ronja.pickGrain();
```

Listing 2: Objektreferenzen

- (a) Besprechen Sie mit Ihrem Teampartner, was passiert, wenn der gesamte Codeschnipsel ausgeführt wird.

- (b) Beantworten Sie die folgenden Fragen bezüglich des Zustands der Hamster nach der Ausführung des Codes aus Listing 2:
- Auf welchen Kacheln befinden sich die Hamster, auf die die Objektreferenzen **paula**, **paule**, **ronnie** und **ronja** zeigen?
  - Wie viele Körner haben die Hamster im Mund, auf die die Objektreferenzen **paula**, **paule**, **ronnie** und **ronja** zeigen?
  - Wie viele Hamster befinden sich auf dem Territorium?
  - Welche Nachteile hat es, zwei Referenzen im selben Sichtbarkeitsbereich auf dasselbe Objekt zu setzen?

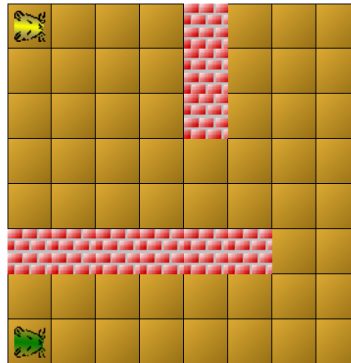


Abbildung 1: Paula und Ronnie

#### Aufgabe 4 Logik und Kontrakte

Die folgende Aufgabe soll die Grundlagen der Logik vertiefen. Außerdem wird die Definition von Verträgen geübt.

- (a) Es sei die folgende logische Beschreibung gegeben:

- paule** schaut nach Norden
- Auf der Kachel von **paule** gibt es kein Korn
- paule** kann einen Schritt nach vorne machen und hat keine Körner im Mund
- Auf dem Territorium gibt es weniger Körner als Hamster

Nehmen Sie an, dass es einen Hamster **paule** und ein passendes Territorium **territory** gibt. Geben Sie zu jeder Aussage einen passenden Ausdruck in Java an. Schreiben Sie den Ausdruck auf!

- (b) Nehmen Sie an, dass der folgende Ausdruck angibt, ob Paule eifersüchtig ist oder nicht.

```

1 (
2     !paule.frontIsClear() && (
3         territory.getNumberOfGrainsAt(ronnie.getLocation())
4             > territory.getNumberOfGrainsAt(paule.getLocation())
5     )
6 ) || paula.getLocation().getRow() == ronnie.getLocation().getRow()
    
```

- Diskutieren Sie mit Ihrem Team, wann Paule eifersüchtig ist. Halten Sie das Ergebnis der Diskussion in natürlich-sprachlichen Stichpunkten fest.  
*Hinweis 1:* Sie können annehmen, dass die Hamster **paule**, **paula** und **ronnie** auf dem selben Territorium existieren.
  - Erstellen Sie eine Wahrheitstabelle für den obigen Ausdruck. Zerlegen Sie den Ausdruck hierfür zunächst in seine einzelnen booleschen Ausdrücke. In der Tabelle soll am Ende **jeder** Ausdruck in einer eigenen Spalte erscheinen und mit jeder Kombination von **true** und **false** ausgewertet werden.
- (c) Definieren Sie für die Operation **pickGrain** unseres Hamsters eine Vor- und eine Nachbedingung in natürlicher Sprache.

## 2 Teil B - Heimarbeit

### Aufgabe 1 Abstrakter Syntaxbaum (PDF)

- (a) (14 Punkte) Gegeben ist die einfache Java Klasse aus Listing 3. Die Objekte `characterInfo` und `gui` sind in `PersonalCharacter` vordefiniert. Zeichnen Sie den abstrakten Syntaxbaum zu Listing 3 analog zur Präsenzteile-Aufgabe aus Aufgabenblatt 3.

```

1  class Barbarian extends PersonalCharacter {
2
3      Integer getHitpoints() {
4          return characterSheet.getHitpoints();
5      }
6
7      void castAbility() {
8          characterSheet.reduceRage(50);
9          gui.update();
10     }
11 }
    
```

Listing 3: Java AST Codebeispiel.

- (b) (10 Punkte) Abbildung 2 zeigt einen Auszug des ASTs der Klasse `PersonalCharacter`. Überführen Sie den Auszug zurück in Java-Code.

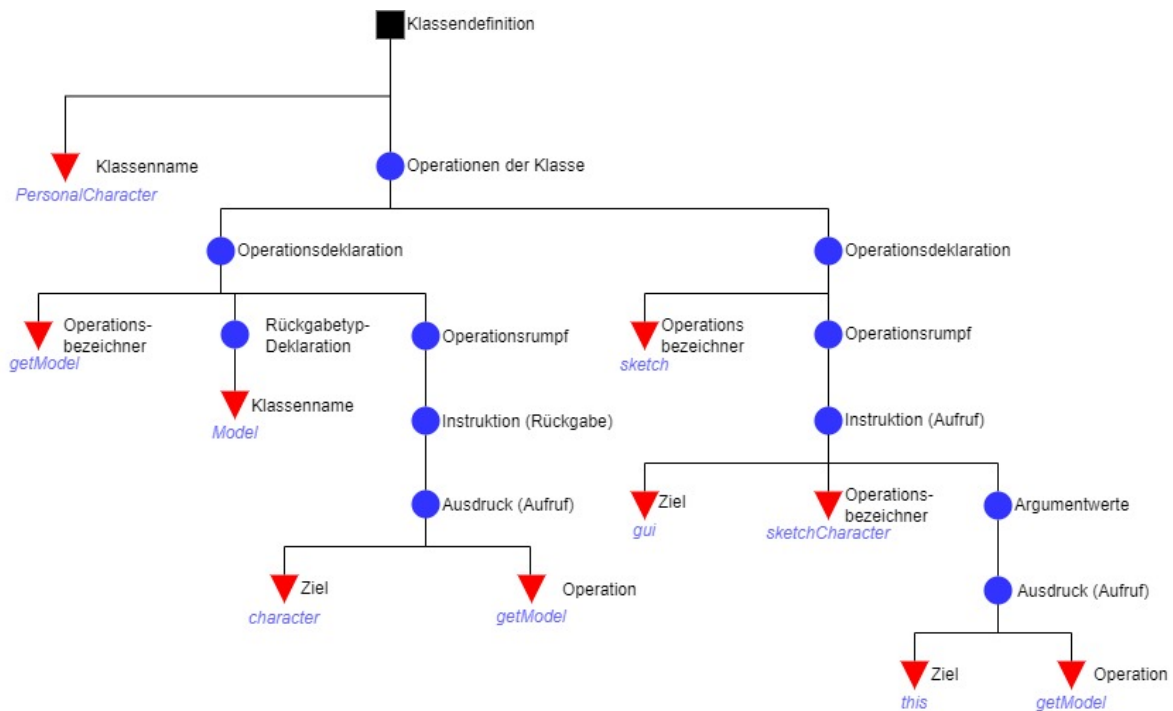


Abbildung 2: Auszug des abstrakten Syntaxbaums der Klasse `PlayableCharacter`.

- (c) Betrachten Sie den Code in Listing 5. In diesem Code haben sich einige Fehler eingeschlichen. Im Folgenden müssen Sie verschiedene Fehler finden. Benennen und klassifizieren Sie jeden Fehler und begründen Sie kurz, warum es sich um einen Fehler dieser Art handelt. Schlagen Sie außerdem eine Verbesserung vor.

*Hinweis:* Ignorieren Sie die fehlende Dokumentation. Antworten, die die fehlende Dokumentation betreffen, werden nicht bepunktet.

- i. (2 Punkte) Finden und beschreiben Sie einen lexikalischen Fehler.
- ii. (4 Punkte) Finden und beschreiben Sie zwei syntaktische Fehler.
- iii. (4 Punkte) Finden und beschreiben Sie zwei statisch semantische Fehler.
- iv. (4 Punkte) Finden und beschreiben Sie zwei (mögliche) dynamisch semantische Fehler.
- v. (4 Punkte) Finden und beschreiben Sie zwei Stylefehler.
- vi. (? Bonuspunkte) Finden und beschreiben Sie weitere Fehler. Für jeden korrekt erläuterten Fehler erhalten Sie 1 Bonuspunkt. Für falsch benannte, falsch klassifizierte, falsch begründete oder falsch verbesserte Fehler gibt es keine Bonuspunkte.

```

1  class Cats extends 4LeggedAnimal {
2      Boolean Move() {
3          frontFeet.pushOff();
4          backFeet.pushOff(;;
5      }
6
7      stop() {
8          backFeet .stop();
9          backFeet.stop();}
10
11     void meowmessage(message, Integer name) {
12         meowName(name);
13         System.out.format("and i meow this : %e \m", name);
14     }
15
16     Integer meowName(String name) {
17         System.out.format("Mewn Name is %s", name);
18         return name;
19     }
20 }
    
```

Listing 5: Java Code einer Klasse.

## Aufgabe 2 APIs lesen und nutzen (PDF)

In dieser Aufgabe betrachten Sie Operationen aus der Klasse `java.lang.String`. Die Klasse verwaltet Zeichenketten in Java und wird häufig benutzt. Sie finden die Dokumentation der Klasse über folgenden Link: <https://tinyurl.com/ye2722zk>

Gegeben ist der folgende Code:

```

1  String statement = "Paule is Ronnies's friend";
2  String result = statement.substring(9,15);
3  System.out.println(result);
4  System.out.println(statement.contains("RONnie"));
    
```

Listing 6: Codeausschnitt

- (a) (3 Punkte) Beschreiben Sie in eigenen Worten, was die Operation `substring(int beginIndex, int endIndex)` bewirkt. Was wird hier durch Zeile 3 auf der Konsole ausgegeben? Welche Vor- und Nachbedingungen hat die Operation?
- (b) (3 Punkte) Beschreiben Sie in eigenen Worten, was die Operation `contains(...)` bewirkt und was durch Zeile 4 auf der Konsole ausgegeben wird. Begründen Sie Ihre Antwort. Geben Sie mindestens eine Vorbedingung für diese Operation an.
- (c) (2 Punkte) Geben Sie einen Funktionsaufruf an, mit dem Sie die Position des ersten 'i's im String `statement` finden können. In diesem Beispiel sollte also die Position des siebten Buchstabens gefunden werden.
- (d) (1 Bonuspunkt) In Zeile 4 wurde der Operation `contains` ein Objekt vom Typ `String` als Argument übergeben, obwohl die Dokumentation eine `CharSequence` als Typ des zu übergebenden Objekt definiert. Erklären Sie kurz, wieso dies möglich ist.