



Übungsblatt 01

Programmierung und Softwareentwicklung (WS 2024/2025)

Abgabe: *Probeabgabe während dem Tutorium* — Besprechung: KW 43

- Bitte lösen Sie die Übungsaufgaben in **Gruppen von 2 Studierenden**.
- Dieses Übungsblatt besteht aus einem Vorbereitungsteil und dem Präsenzteile. In Zukunft werden Übungsblätter aus zwei Teilen (A, B) bestehen. Teil A ist in der Präsenzübung zu lösen. Teil B ist in Heimarbeit (Gruppe von 2 Studierenden) zu lösen und fristgerecht abzugeben. Die Abgabe erfolgt über ILIAS.
- 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: PSE01-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: Dieses Übungsblatt dient dazu, Sie mit dem Übungsbetrieb vertraut zu machen, Ihren Tutor kennen zu lernen, Ihren Rechner oder Ihre Arbeitsumgebung aufzusetzen und Sie an die Integrated Development Environments (IDE) Eclipse bzw. IntelliJ IDEA heranzuführen.

Punkte: Für dieses Übungsblatt können noch keine Punkte erzielt werden. Es dient zum Installieren, Einrichten und Kennenlernen der Werkzeuge, die Sie in den kommenden Wochen nutzen müssen.

Style: Für dieses Übungsblatt gibt es noch keine Style-Regeln. Auf zukünftigen Blättern wird der Style ihrer Implementierung jedoch mit bis zu 50% bewertet.

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

- Dieser Teil wird Ihnen zukünftig die Aufgaben zur Vorbereitung nennen. Für die erste Übung haben wir allerdings einen dedizierten Block. Bearbeiten Sie daher bitte Teil 0 - Vorbereitung.

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 0 - Vorbereitung

Aufgabe 1 Vorbereiten der IDE & Java

Für die Übungen werden wir verschiedene IDEs unterstützen. Sie können sich eine der IDEs aussuchen und diese mit den folgenden Erklärungen einrichten.

- (a) Falls Sie IntelliJ IDEA nutzen wollen, fahren Sie mit den Aufgaben 2 und 3 fort.
- (b) Falls Sie Eclipse nutzen, fahren Sie mit den Aufgaben 4 und 5 fort.
- (c) Falls Sie Visual Studio Code nutzen wollen, finden Sie eine kurze Videoanleitung hier: <https://youtu.be/BGJC0pQsQAo>. Sollte es hierbei Schwierigkeiten geben, empfehlen wir jedoch auf eine der obigen IDEs zu wechseln.
- (d) Falls Ihr Rechner nicht stark genug für eine der IDEs ist, melden Sie sich bitte bei Ihrem Tutor.

Aufgabe 2 Vorbereiten von IntelliJ IDEA

Die folgende Anleitung gibt es auch als Video: <https://youtu.be/mt7smtBQt9o> (Part 1)

- (a) Laden Sie sich die aktuelle Version von IntelliJ IDEA Community herunter: <https://www.jetbrains.com/idea/download>. Achten Sie darauf, die Community Edition herunterzuladen. Eine Installationsanleitung finden Sie hier: <https://www.jetbrains.com/help/idea/installation-guide.html#standalone>
Unter Windows empfehlen wir die Verwendung des Installers (.exe Download Option), hierbei können Sie die Standardeinstellungen des Installers verwenden.
- (b) Installieren Sie Git. Windows-User können diesen Schritt überspringen und git später direkt aus IntelliJ heraus installieren. Eine Installationsanleitung für gängige Betriebssysteme finden Sie unter anderem hier: <https://github.com/git-guides/install-git>.
- (c) Optional: Installieren Sie folgende Plugins für IntelliJ IDEA:
Um ein Plugin für IntelliJ zu installieren, öffnen Sie zuerst das Fenster zum Verwalten von Plugins.

- Windows, Linux: File → Settings → Plugins
- macOS: IntelliJ IDEA → Preferences... → Plugins

Achten Sie darauf, dass oben der Reiter Marketplace ausgewählt ist. Unter Plugins befindet sich ein Suchfeld, mithilfe dessen Sie verfügbare Plugins suchen können, empfohlene Plugins folgen später. Klicken Sie auf Install, um ein Plugin zu installieren. Folgen Sie daraufhin den Anweisungen.

- i. **CodeTogetherLive** (Optional): Mit dem Tool CodeTogether können Sie eine Datei bzw. ein Projekt gleichzeitig bearbeiten. Gehen Sie auf <https://www.codetogether.com>, um weitere Informationen zu erhalten.
- ii. **CheckStyle-IDEA** (Optional): CheckStyle-IDEA ist ein Plugin, das Ihnen definierte Stilfehler markiert.

Aufgabe 3 Importieren eines Git-Maven Projektes in IntelliJ IDEA & Java Setup

Die folgende Anleitung gibt es auch als Video: <https://youtu.be/mt7smtBQt9o> (Part 2)

- (a) Falls noch nicht getan, starten Sie IntelliJ IDEA.
- (b) Haben Sie schon ein Projekt geöffnet, schließen Sie dies unter File → Close Project.
Hinweis: Beim ersten Start ist kein Projekt geöffnet.
- (c) Klicken Sie auf Get from VCS. Es öffnet sich ein neuer Dialog. Achten Sie darauf, dass nun der Abschnitt Repository URL ausgewählt ist, und unter Version control Git.
- (d) Fügen Sie unter URL die URL des gewünschten Projekts ein. Für dieses Blatt lautet diese <https://github.com/SQA-PSE-WS-2024-2025/exercise-sheet-01.git>.
- (e) *Hinweis für Windows-User:* Sollten Sie Git noch nicht installiert haben, erscheint eine Warnmeldung, dass Git noch nicht installiert ist. Um Git zu installieren, klicken Sie auf Download and Install und warten Sie bis IntelliJ die Installation von Git abgeschlossen hat.

- (f) Bestätigen Sie mit `Clone`. Wählen Sie im sich nun öffnenden Dialog `Trust project`.
- (g) Es öffnet sich nun ein neues Fenster mit dem Projekt. Legen Sie wie folgt das JDK fest: Öffnen Sie `File` → `Project Structure...`. Unter `Project Settings` → `Project` → `SDK` befindet sich ein Dropdown, mithilfe dessen Sie das JDK auswählen können.
 - i. **Fall 1: Ein JDK 21 ist schon vorhanden:** Wählen Sie in diesem Fall aus der Liste das gewünschte JDK aus.
 - ii. **Fall 2: Ein JDK 21 ist in der Liste noch nicht vorhanden, jedoch ist auf dem System schon ein JDK 21 installiert:** Wählen Sie `Add SDK` → `JDK...` aus, und finden Sie im sich nun öffnenden Dialog das JDK in Ihrem Dateisystem.
 - iii. **Fall 3: Ein JDK 21 ist in der Liste noch nicht vorhanden, und auch nicht auf dem System installiert:** Wählen Sie `Add SDK` → `Download JDK...` aus, und wählen Sie Ihr gewünschtes JDK aus. Wir empfehlen `Version`: 21 sowie `Vendor`: Eclipse Temurin. Bestätigen Sie mittels `Download`.
- (h) Stellen Sie sicher, dass unter `Language level` „21 [...]“ ausgewählt ist, und bestätigen Sie mittels `OK`.

Aufgabe 4 Vorbereiten von Eclipse & Java

- (a) Installieren Sie Java 21 (64 bit) auf Ihrem Gerät. Sie finden beispielsweise Open JDK unter <https://jdk.java.net/archive>. Nutzen Sie eine Suchmaschine Ihrer Wahl und recherchieren Sie nach einer zu Ihrem Betriebssystem passenden Installationsanleitung.
- (b) Laden Sie sich die aktuelle Version von Eclipse herunter: <https://www.eclipse.org/downloads>. Achten Sie darauf, die richtige Eclipse-Version herunterzuladen oder im Installations-Wizard auszuwählen (Eclipse IDE for Java Developers).
- (c) Installieren Sie nun folgende Plugins in Ihrem Eclipse-System:
 - i. **Maven-M2E:** Maven ist ein Build-Management-Tool für Java-basierte Software. Unsere Projekte werden immer auf Maven basieren. Falls Sie Eclipse nicht mit dem Installer installiert haben oder Ihr Eclipse nicht schon das Maven-M2E Plugin mitbringt, müssen Sie noch das Plugin **M2E** manuell installieren, um Maven auch in Eclipse nutzen zu können. Folgen Sie dafür folgender Anleitung: `Help` → `Install New Software`, fügen Sie im Textfeld `Work with` die URL <https://download.eclipse.org/technology/m2e/releases/latest> ein. Öffnen Sie das Untermenü `Maven Integration for Eclipse` und wählen Sie in der Liste das Plugin `M2E - Complete Development Kit` aus. Klicken Sie dann auf `Next >` und folgen Sie den weiteren Anweisungen.
 - ii. **M2E-EGit:** EGit erlaubt Ihnen den Import von Maven-Projekten aus GitHub. Öffnen Sie `Help` → `Install New Software`, fügen Sie im Textfeld `Work with` die URL <https://github.com/tesla/m2eclipse-egit/releases/download/0.17.0/> ein. Öffnen Sie das Untermenü `m2e extensions` und wählen Sie in der Liste das Plugin `Maven SCM Handler for EGit` aus. Klicken Sie dann auf `Next >` und folgen Sie den weiteren Anweisungen.
Hinweis: Sie werden aufgefordert, Eclipse neu zu starten. Tun Sie dies.
 - iii. **CodeTogether** (Optional): Mit dem Tool CodeTogether können Sie eine Datei bzw. ein Projekt gleichzeitig bearbeiten. Gehen Sie auf <https://www.codetogether.com> um weitere Informationen zu erhalten. Sie können das Tool über den Eclipse Marketplace installieren.
 - iv. **CheckStyle** (Optional): CheckStyle ist ein Plugin, das Ihnen definierte Stilfehler markiert. Um dieses zu installieren, gehen Sie ähnlich wie bei Maven-M2 vor:
`Help` → `Install New Software`, fügen Sie im Textfeld `Work with` die URL <https://checkstyle.org/eclipse-cs-update-site>, klicken Sie auf `Eclipse Checkstyle Plug-in`. Klicken Sie dann auf `Next` und folgen Sie den Anweisungen.

Aufgabe 5 Importieren eines Git-Maven Projektes in Eclipse

Um ein Maven Projekt von GitHub in Eclipse zu importieren, führen Sie bitte folgende Schritte aus (auch als Video: <https://youtu.be/7c9JEk783ac>):

1. Vergewissern Sie sich, dass Sie die Plug-Ins **Maven-M2E** und **M2E-EGit** installiert haben.
2. Klicken Sie nun auf **File** → **Import...** und wählen Sie **Check Out Maven Projects from SCM** unter dem Reiter **Maven**. Klicken Sie dann auf **Next**.
3. Wählen Sie unter **SCM URL** im ersten Drop-Down Menü **git** aus.
4. Kopieren Sie die URL des gewünschten Git-Projekts in das Textfeld bei **SCM URL**. Für dieses Blatt lautet die URL <https://github.com/SQA-PSE-WS-2024-2025/exercise-sheet-01.git>. Klicken Sie dann auf **Finish**.
Sollte ein Fehler „Can't rename ...“ auftreten, begeben Sie sich in Ihren Workspace (den Ort haben Sie beim Start von Eclipse definiert). Löschen Sie dort den neuen Ordner, dessen Namen mit „maven.“ beginnt und versuchen Sie es erneut. Alternativ können Sie diesen Ordner auch als Maven Projekt importieren: **File** → **Import** → **Maven** → **Existing Maven Projects**.
5. Danach wird Ihnen eventuell eine Liste an **pom.xml** Dateien angezeigt. Bei uns sollte immer nur eine angezeigt werden. Wählen Sie diese aus und klicken Sie auf **Finish**. Sollten mehr als eine **pom.xml** Datei angezeigt werden, wenden Sie sich an Ihren Tutor.
6. Wechseln Sie im Anschluss in die Java-Perspektive: **Window** → **Perspective** → **Open Perspective** → (evtl **Other...** →) **Java** (default).
7. Im linken Panel finden Sie nun den Package Explorer, in dem die Dateien des Projekts angezeigt werden. Klappen Sie das Projekt auf, finden Sie den Ordner **src/main/java**. Dort befindet sich der Code. Im Paket **de.unistuttgart.iste.sqa.pse.sheet01.presence** befindet sich die Klasse **FirstHamsterGameApp**. Sie können nun das Programm ausführen, indem Sie die Klasse öffnen und anschließend auf den grünen Pfeil in der Kopfleiste klicken. Es sollte dann in einem neuen Fenster das Spielfeld erscheinen. Alternativ können Sie das Spielfeld auch wie folgt öffnen: **Rechtsklick auf FirstHamsterGameApp** → **Run As** → **Java Application**.
8. Problembehandlung: Beachten Sie, dass für einen reibungslosen Ablauf Eclipse mit Java 21 gestartet und konfiguriert sein muss. Wenn Sie mehrere Java Versionen installiert haben, kann es zu Problemen kommen, sollte die falsche Version in Eclipse eingestellt sein.
Zudem kann es vorkommen, dass die **vm** Variable nicht gesetzt ist. In dem Fall beachten Sie bitte die Hinweise aus dem Eclipse-Wiki: <https://wiki.eclipse.org/Eclipse.ini>. Falls Sie jedoch den Eclipse Installer genutzt haben, sollte die VM gleich gesetzt worden sein.
Ein Anzeichen dafür, dass die Java Version in Eclipse falsch eingestellt wurde, ist, dass in Ihren Projekten beispielsweise „JavaSE-17“ statt „JavaSE-21“ angezeigt wird, obwohl Sie alle vorigen Schritte befolgt haben. Gehen Sie dann wie im Folgenden (alternativ ähnlich zum Video für JDK 15 vorgehen: <https://www.youtube.com/watch?v=KC6aM7DYQJw>) erklärt vor:
 - (a) Stellen Sie sicher, dass die Eclipse Version „2024-09 (4.33.0)“ (oder aktueller) installiert ist. Sie finden die Version folgendermaßen:
 - Windows, Linux: **Help** → **About Eclipse IDE**
 - macOS : **Eclipse** → **About Eclipse**
 - (b) Setzen Sie in den Einstellungen (**Eclipse** → **Preferences** → **Java** → **Compiler**) (Hinweis: Optionen können bei verschiedenen Betriebssystemen unterschiedlich benannt sein) den „Compiler Compliance Level“ auf 21.
 - (c) Eventuell müssen Sie Ihre Projekte aktualisieren, z.B. über Rechtsklick auf das Projekt, dann **Maven** → **Update Project...** → **OK**.

2 Teil A - Präsenzaufgaben

Aufgabe 1 ILIAS und Abgabesystem

Lassen Sie sich von Ihrem Tutor zeigen, wie Sie ILIAS nutzen und Abgaben darin tätigen können. Hierfür wird Ihr Tutor Ihnen die folgenden Inhalte mit einer Live-Demo vorführen:

- (a) Vorstellung des ILIAS Kurses, der ILIAS Übungsgruppe, und der ILIAS Foren (Gruppe und allgemein).
- (b) Vorstellung des Abgabesystems und -prozesses.

Aufgabe 2 Abgabegruppenerstellung und erste Datei hochladen

In dieser Aufgabe werden Sie sich in Gruppen einteilen und eine erste Datei ins ILIAS hochladen. Dies dient zur Erfassung Ihrer Gruppeneinteilung, sowie Ihrer Matrikelnummern.

Hinweis: Wenden Sie sich bei Fragen oder Problemen an Ihren Tutor.

- (a) Teilen Sie sich möglichst selbstständig in 2er Abgabegruppen ein (nur wenn das nicht aufgeht, ist auch eine 3er-Gruppe möglich).
- (b) Erstellen Sie eine PDF, die die vollständigen Namen aller Teammitglieder Ihrer Abgabegruppe, sowie deren Matrikelnummern und studentische E-Mail-Adressen enthält. Optional können Sie noch die Studiengänge aller Teammitglieder Ihrer Abgabegruppe angeben.

Hinweis: Ihr Tutor wird diese Daten verwenden, um Sie in unserem Managementsystem einzutragen. Achten Sie daher bitte auf Korrektheit.

- (c) Erstellen Sie auf ILIAS, wie von Ihrem Tutor vorgestellt, eine Abgabegruppe und laden Sie die PDF hoch. Geben Sie Ihre Datei in Abgabe 1 ab.

Aufgabe 3 Einführung in den Hamstersimulator

In dieser Aufgabe lernen Sie unsere Mini Programming World, den Hamstersimulator, kennen. Um diese Aufgabe lösen zu können, sollten Sie bereits die Aufgaben aus Teil 0 - Vorbereitung erledigt haben. Im Hamstersimulator schlüpfen Sie in die Rolle des kleinen und verfressenen Hamsters **Paule** und seinen Freunden. Sie steuern dabei die Hamster durch verschiedene Szenarien. Dabei erleben **Paule** und seine Freunde diverse Abenteuer und sammeln auf ihren Wegen Körner ein. Anfangs lösen Sie die Aufgaben noch spielerisch, im Laufe des Semesters werden Sie dies zunehmend mit von Ihnen verfasstem Quellcode tun und dabei immer mehr neue Konzepte anwenden. Bevor es jedoch richtig ans Werk geht, müssen Sie sich erst mit **Paule** und seiner Welt vertraut machen. Lösen Sie hierfür die folgenden Teilaufgaben.

Hinweis: Manchmal müssen Sie den Simulator zwischen den Teilaufgaben neustarten, z.B. um das Spielfeld auf den Ausgangszustand zurückzusetzen oder falls das Spiel in einen fehlerhaften Zustand gerät.

- (a) Wählen Sie die Datei `FirstHamsterGameApp.java`. Sie finden diese unter `src → main → java` im Package `de.unistuttgart.iste.sqa.pse.sheet01.presence`. Starten Sie das Programm in Ihrer IDE. Sie können das Programm in Eclipse oder IntelliJ IDEA über den grünen Run-Knopf kompilieren und ausführen. Bei erfolgreicher Ausführung sollte sich ein zweites Fenster mit einer UI für das Spielfeld öffnen.
- (b) Machen Sie sich mit der UI vertraut. Links befindet sich das Spielfeld (**territory**), auf der sich der Hamster (**paule**), sowie einige Körner (**grains**) befinden. Rechts davon befinden sich das Log und die UI Elemente, um Klassen und Objekte zu inspizieren. Bspw. können Sie über die UI Elemente rechts Paules Fähigkeiten (*Operationen*) aufrufen. Dadurch können Sie Paule in den ersten Übungen steuern.
- (c) Verschaffen Sie sich einen groben Überblick über die Fähigkeiten Paules.
- (d) Bewegen Sie Paule in Blickrichtung nach vorne, bis er eine Wand erreicht. Bleiben Sie vor der Wand stehen.
- (e) Auf Paules Startfeld liegt bereits ein Korn. Lassen Sie Paule das Korn aufheben.
- (f) Bewegen Sie Paule entlang der Linie mit Körnern von links unten nach rechts oben.
- (g) Heben Sie 5 Körner auf dem Spielfeld auf. Lassen Sie Paule anschließend alle aufgesammelten 5 Körner auf ein Feld ablegen.

- (h) Benutzen Sie die Operation `write()` und lassen Sie *Paule* einen Satz ausgeben, z.B. „My name is Paule!“.
- (i) Lassen Sie *Paule* gegen eine Wand laufen oder versuchen Sie, ein Korn auf einem leeren Feld aufzuheben. Was passiert und wie könnten Sie solche Fehler verhindern? Diskutieren Sie Ihre Beobachtungen und Ideen mit Ihrem Nachbarn.

Hinweis: Fehler werden in einer späteren Vorlesung behandelt.

Hinweis: Wenn das Spiel in einem fehlerhaften Zustand ist, müssen Sie es neu starten.

Aufgabe 4 Dokumentation des Hamstersimulators

Ziel dieser Aufgabe ist es, den Hamstersimulator – und Ihre Kommilitonen – eigenständig besser kennen zu lernen. Öffnen Sie dazu die Dokumentation des Hamstersimulators. Den Link zur Dokumentation des Hamstersimulators finden Sie auf der ersten Seite jedes Aufgabenblattes – und so auch auf diesem – unter *Unterlagen*.

Hinweis: Sie werden in den nächsten Aufgabenblättern immer wieder auf die Dokumentation des Simulators zurückgreifen. Zu Beginn werden die Aufgabenblätter Sie noch explizit dazu auffordern, später müssen Sie eigenständig einschätzen, wann Sie etwas nachschlagen sollten.

- (a) Nutzen Sie die Suchfunktion, um die Dokumentation für Hamster¹ zu finden. Lesen Sie die Dokumentation. Bearbeiten Sie dann die weiteren Teilaufgaben.
- (b) Welche Operation des Hamsters haben Sie schon verwendet, welche noch nicht?
- (c) Betrachten Sie die Dokumentation für andere Bestandteile unserer Hamsterwelt, z.B. für Territorien (*Territory*), Positionen (*Location*), Blickrichtungen (*Direction*) oder für das Spiel an sich (*SimpleHamsterGame*).
- (d) Welche Bestandteile der Dokumentation verstehen Sie, welche sind Ihnen noch unbekannt? Diskutieren Sie mit Ihrem Partner und mit anderen Gruppen.
- (e) Starten Sie den Hamstersimulator wie in Aufgabe 3 beschrieben. Probieren Sie weitere Operationen des Hamsters `paule` und der anderen Objekte aus. Ziehen Sie die Dokumentation zu Rate, falls Sie eine Operation nicht verstehen.

¹<https://javadoc.io/doc/de.hamstersimulator.objectsfirst/main/latest/de.hamstersimulator.objectsfirst.main/de/hamstersimulator/objectsfirst/external/model/Hamster.html>