

Allgemeine Informationen zur Übung

Site: [Moodle-Service der Universität Wien](#)
Course: 2022W Software Engineering 1
Book: Allgemeine Informationen zur Übung

Printed by: Boghean Adrian
Date: Friday, 17 February 2023, 8:35 PM

Description

Lesen Sie sich die folgenden Informationen aufmerksam durch. Diese sind teilweise bewertungsrelevant.

Table of contents

Allgemeine Informationen zur Übung

Verpflichtend einzusetzender Technologie Stack

Unterstützung bei der Ausarbeitung

Abgabegespräche

Abgabe der Teilaufgaben über GitLab

- Git: Git&GitLab Tutorial, EGit, Lernplattform, Cheatsheet, Loginprobleme und Zertifikate

Übernahme von Teillösungen aus externen Quellen

Im Rahmen der Übung erreichbare Bonuspunkte

Auswirkung der Übungs-Deadlines

Allgemeine Informationen zur Übung

Hier werden allgemeine Informationen zur Übung zusammengestellt. Diese sind teilweise bewertungsrelevant und definieren, beispielsweise, die zwingend einzusetzenden Technologien und geben Tipps/Hinweise zur Durchführung der Abgaben (z.B. zur Lösung bekannter Probleme). Davon abzuweichen kann dazu führen, dass Ihnen keine oder nicht mehr alle Punkte für eine Teilaufgabe anerkannt werden können. Auch wird eine Vorschau darauf geboten welche Möglichkeiten im Rahmen der Übung bestehen um Bonuspunkte zu sammeln.

Relevanz: Die hier hinterlegten Informationen sind für alle drei Teilaufgaben relevant.

Navigation: Um in den Materialien zu navigieren verwenden Sie entweder die Pfeilbuttons rechts oder das Inhaltsverzeichnis rechts oben (empfohlen).

Verpflichtend einzusetzender Technologie Stack

Für die Implementierung von Teilaufgabe 2 und 3 ist sofern nicht anders angegeben folgender Technologie Stack verpflichtend zu verwenden. Die von der LV bereitgestellten Basisprojekte sind bereits so konfiguriert, dass die passenden Versionen verwendet werden.

Programmiersprache	Java 17 (1.17)
Entwicklungsumgebung	Eclipse 2022-09 for Java Developers (siehe auch den Hinweis unten)
UI-Framework	Client: Voraussetzung CLI für automatisierte Tests. Eine zusätzliche aufwändigere GUI mit freier Technologiewahl (beispielsweise JavaFX oder Swing) ist optional.
Netzwerkkommunikation	Spring Boot 2.5.5, Spring Webflux 2.5.5, JAXB Reference Implementation 2.3.X, iStack 4.0.1
Bereitgestellte Netzwerkprotokoll Implementierung	Binden Sie das bereitgestellte Netzwerkprotokoll (über ein Build Management Tool) in Version 4.0.0 ein.
Logging	Logback 1.2.6 mit SLF4J 1.7.32
Unit Testing	JUnit 5.8.1 und Mockito 3.12.4
Sourcecode Verwaltung	GitLab der Universität (siehe Git Tutorial)
Dokumentation des Quellcodes	JavaDoc Stil - empfohlen, nicht verpflichtend
Build Management Tools	Müssen zum Einbinden aller Bibliotheken verpflichtend verwendet werden. Empfehlung: Gradle
Datenbankeinbindung	OR Mapper für Management und Zugriff (Hibernate via Spring Data JPA 2.5.5), SQLite JDBC via Xerial SQLite JDBC 3.36.0.3

Auswahl der Technologien: Die oben ausgewählten Tools und Bibliotheken wurden aufgrund deren Verbreitung, und durch die von der LV-Leitung geprüften Stabilität, Aktualität und Relevanz ausgewählt. Aufgrund der unüberschaubaren Anzahl an Frameworks und Tools können und werden Tipps und Hilfestellungen nur für diese gegeben.

Alternative IDEs: Statt Eclipse können Sie auch eine alternative Entwicklungsumgebung nützen, achten Sie jedoch darauf, dass sich Ihr Projekt wieder problemlos in Eclipse importieren lässt, da dies während der Abgabegespräche und Bewertungen notwendig wird (*hierfür sind Sie alleinig verantwortlich*). Einige LVs setzen derzeit Technologien ein welche nur innerhalb von Eclipse funktionieren, bereits jetzt etwas Erfahrung damit zu sammeln ist hierbei ebenfalls vorteilhaft.

Unterstützung bei der Ausarbeitung

Hilfreiche Unterlagen für den Übungsteil: Dies umfasst [Tipps und Tricks zur Softwarearchitektur](#), [Tipps zum Verständnis von XML und der Implementierung einer Netzwerkkommunikation](#) sowie nützliche Informationen bezüglich [Logging](#), [Unit Testing und Mocking](#). Basierend hierauf sind alle Grundlagen vorgegeben welche Sie zur praktischen Umsetzung der Übung benötigen. Typischerweise lohnt es sich diese Unterlagen vor der jeweils passenden Tutorialeinheit anzusehen um dieser besser folgen zu können.

Tutorials: Die LV bietet mehrere auf die einzelnen Teilaufgaben ausgerichtete Tutorials an (siehe Moodle). Wir empfehlen diese zu besuchen sowie die bereitgestellten Tipps und Tricks zu berücksichtigen.

Vorlesung: Die im Rahmen der Aufgabenblätter anhand von praxisorientierten Beispielen durchgenommenen Themen können auch für die Umsetzung der Teilaufgaben hilfreich sein.

Netzwerkprotokoll: Eine mustergültige Dokumentation der Netzwerkschnittstelle ist [hier](#) oder in der [Angabe zu Teilaufgabe 1](#) zu finden. Weiters erhalten Sie für Teilaufgabe 2 und 3 eine fertige Implementierung aller Nachrichten und die [Dokumentation dieser Nachrichtenimplementierung](#).

Beispielprojekte: Beispielprojekte für alle Themenbereiche (beispielsweise Logging und Unit Testing) sowie auch der Implementierung des Clients bzw. Servers werden nach den jeweiligen Einheiten auf Moodle zur Verfügung gestellt. Wir empfehlen insbesondere die bereitgestellten Client und Server Beispielprojekte als Ausgangsbasis für die eigene Implementierung zu verwenden. Sie erhalten so eine dokumentierte Basis (auch mit Beispielcode) die bereits alle notwendigen Teile des Build Managements richtig konfiguriert beinhaltet und die notwendigen Bibliotheken in der passenden Version einbindet.

Automatische Evaluation: Sie können Ihre Implementierung von Teilaufgabe 2 und 3 automatisch auf Korrektheit überprüfen lassen. Nützen Sie die automatische Evaluierung um die Korrektheit Ihrer Implementierung zu überprüfen. Details hierzu werden in den Übungseinheiten besprochen und auch auf Moodle bekanntgegeben.

Bereitgestellte Server- bzw. Clientimplementierung: Damit Sie Ihren Client bzw. Server problemlos implementieren können wurde eine Server- bzw. Clientimplementierung von der LV-Leitung bereitgestellt. Gegen diese können Sie Ihre Server bzw. Client testen bzw. integrieren. Details hierzu werden jeweils bei der Vorbesprechung von Teilaufgabe 2 und 3 veröffentlicht.

Terminplanung von Teilaufgabe 2: Um Sie zu motivieren rechtzeitig mit der Clientimplementierung zu beginnen wurden Minideadlines unter <http://swe1.wst.univie.ac.at> definiert. Wenn Sie diese einhalten erhalten Sie nicht nur Bonuspunkte, sondern auch eine angenehme Arbeitsverteilung.

Forum: Sollten trotzdem Fragen zu Angaben, Spielidee, Umsetzung etc. auftauchen scheuen Sie sich nicht davor diese auch in den Einheiten oder im [Moodle Forum](#) zu stellen! Wir werden versuchen diese so gut und so schnell wie möglich zu beantworten.

Abgabegespräche

Nützen Sie die Abgabegespräche um direkt Feedback und eine Fachdiskussion zu Ihrer Lösung zu erhalten. Erläutern Sie dabei Ihre Ideen und Umsetzung und erhalten Sie Tipps und Hinweise um Ihr individuelles Verbesserungspotential auszunützen. Diese Chance wird Ihnen im universitären Alltag aufgrund des großen Aufwandes für die Lehrenden selten geboten.

Hierzu wird jede der drei Teilaufgaben von der Möglichkeit zu einem individuellen Abgabegespräch begleitet bei denen Sie Tipps und Feedback zu Ihrer Abgabe erhalten können. Nebenbei findet hier auch die Bewertung statt. Sie müssen sich selbstständig zu einem der auf Moodle bereitgestellten Termine anmelden. Es wird ausreichend Termine geben damit alle einen Termin erhalten könnten – jedoch ist nicht garantiert, dass Sie einen für sich optimalen Termin erhalten. Je früher Sie daher einen Termin aus den möglichen Terminen auswählen desto mehr Auswahl haben Sie.

Vergessen Sie nach der Anmeldung nicht auf die Abgabegespräche. Sobald Sie sich für einen Termin entschieden haben müssen Sie diesen wahrnehmen um Punkte auf Ihre Abgabe zu erhalten. Falls Sie aus einem wichtigen Grund nicht an Ihrem Termin teilnehmen können oder Sie sich zu dem von Ihnen ausgewählten Termin verspäten informieren Sie uns bitte **vorab** darüber. In jedem Fall vor dem Ende der Anmeldefrist bzw. vor dem Termin. Genauere Details erhalten Sie sobald die jeweilige Anmeldung auf Moodle freigegeben wird.

Abgabe der Teilaufgaben über GitLab

Wir verwenden für alle Abgaben das GitLab der Fakultät, dieses ist erreichbar unter: <https://git01lab.cs.univie.ac.at> (Nicht mit dem GitLab der SWA-Gruppe verwechseln!).

Stellen Sie sicher, dass sich **vor** dem Ende der Abgabefrist Ihre aktuelle Ausarbeitung (Dokumentation bzw. Source Code und JAR) im "**master**" Branch Ihres privaten GitLab Repositories des **aktuellen** Semesters befindet. Dessen Zustand, wird automatisch kopiert, eingefroren und für die kommende Bewertung herangezogen. Nützen Sie zum Ablegen der Ausarbeitungen die vorgegebene Ordnerstruktur. Andere Branches oder nur im Status "committed" befindliche aber nicht gepushte Lösungen können nicht für die Bewertung herangezogen werden.

Regelmäßige Commits: Pushen sie so oft wie möglich Ihre Ausarbeitung in Ihr GitLab Repository. Das stellt zum einen sicher, dass Sie die aktuelle Version für die Abgabe gepusht haben und zum andere können Sie bei Fehlern ältere Zustände einfach wiederherstellen. Es kann außerdem zu Punkteabzügen kommen, wenn Sie GIT unsachgemäß verwenden, daher falls beispielsweise nur ein einzelner Commit kurz vor dem Ende der Deadline mit der kompletten Ausarbeitung erfolgt. Machen Sie daher, wie in der Praxis üblich, mehrere Commits [> 10] und verwenden Sie aussagekräftige Commit-Messages. Dies ist Bewertungsrelevant.

Vor der Deadline: Kontrollieren Sie den Zustand Ihres Repositories auch über die GitLab Website. Relevante Fragen dabei sind: Habe ich alles gepusht? Lässt sich die Software nach einem Clone auch in Eclipse importieren und bauen? Funktioniert die automatische Evaluierung? Pushe ich in den richtigen Branch (`master`) und in das richtige zum Semester passende Repository?

Um auf Nummer sicher zu gehen (empfohlen): Überprüfen Sie die Vollständigkeit Ihrer Abgabe indem Sie Ihre Abgabe downloaden und in einen neuen Eclipse Workspace importieren und bauen/testen.

Grundlegende Erwartungen an Ihre Abgabe: Ihre Implementierung am GitLab Server muss zum Zeitpunkt der Abgabedeadline **in der vorgegebenen Eclipse Version problemlos importierbar, kompilierbar und lauffähig** sein sowie sich in ein ausführbares JAR übersetzen lassen!

Automatische Evaluation Teilaufgabe 2&3: Beachten und berücksichtigen Sie für Teilaufgabe 2 auch die Security Policies am Testserver. Wenn Ihr Code gegen diese verstößt kann dies dazu führen, dass die automatische Evaluierung am Testserver der LV fehlschlägt. Probieren Sie es also davor aus. Vergleichbar hierzu wird auch für Teilaufgabe 3 ein Testclient zur Verfügung gestellt. Die notwendigen Informationen erhalten Sie sobald die jeweilige Angabe verfügbar wird.

Qualität Ihrer Ausarbeitung: Die abgegebenen Implementierungen müssen kompilierbar (ausführbar) sein und die Funktionalität bzw. Aspekte funktionsfähig und korrekt beinhalten. Sollte dies nicht der Fall sein, können nicht mehr alle Punkte angerechnet werden. Auch für fehlerhafte (nicht der Angabe – Spielidee entsprechend, Abstürze, unvollständige Umsetzung bzw. Implementierung etc.) oder schlechte (Performance, Wartbarkeit etc.) Implementierungen kann es Abzüge bis hin zu Bewertung mit 0 Punkten geben.

Git: Git&GitLab Tutorial, EGit, Lernplattform, Cheatsheet, Loginprobleme und Zertifikate

Git&GitLab Tutorial, Lernplattform & Cheatsheet

Wir haben ein [Lehrvideo](#) sowie eine [Zusammenfassung aller relevanten Informationen](#) bezüglich Git & GitLab für Sie erstellt. Außerdem können Sie den Umgang mit Git online in Ihrem Webbrowser schnell und einfach erlernen unter: [TryGit](#)
Für den Umgang mit Ihrer lokalen Git-Installation ist voraussichtlich ein [Git Cheatsheet](#) hilfreich. Dieses listet die wichtigsten Befehle auf und gibt außerdem Beispiele zu deren Verwendung. Ein Cheatsheet welches wir empfehlen können wurde verlinkt. Für die Belange dieser LV sollten die Befehle aus den Bereichen "Git Basics" und "Remote Repositories" ausreichen.

Login am Gitserver schlägt fehl

Problem: Während des git clone Vorgangs meldet der Git Server, dass meine Logindaten nicht korrekt wären. Trotzdem werde ich vom Git-Client auch nicht nach meinem Usernamen und Passwort gefragt. Was kann ich tun?

Lösung: Fügen Sie Ihren u:account Usernamen zur Url hinzu (zwischen https:// und git01lab.cs.univie.ac.at) während des Aufrufs von git clone, typischerweise wird Git Sie dann nach dem passenden Passwort fragen. Vergessen Sie nicht auf das @ zwischen Ihrem u:account und git01lab.cs.univie.ac.at.

Ein Beispiel:

Ursprünglicher Befehl: `git clone https://git01lab.cs.univie.ac.at/VU_Software_Engineeri...`

Befehl welcher Ihren u:account Usernamen in die URL inkludiert: `https://IhrUAccountUser@git01lab.cs.univie.ac.at/VU_Software_Engineeri...`

Probleme mit den Zertifikaten

Falls Sie beim clonen Ihres Git Repositories Probleme gemeldet bekommen welche mit der Prüfung der Zertifikate in Verbindung stehen ist es eine schnelle Lösung diese abzuschalten. Hierzu kann folgender Befehl verwendet werden: `git config --global http.sslVerify false`

Grafischer Git Client (in Eclipse integriert)

Der im Rahmen des Git-Tutorials gezeigte grafische Git-Client trägt den Namen EGit und ist direkt in Eclipse integriert. Diesen müssen Sie nicht verwenden, daher der ebenfalls gezeigte Konsolencient für Git und die Befehle `clone`, `add`, `commit` und `push` sind ausreichend. Sollten Sie jedoch die Informationen zum EGit gerne nachlesen wollen können Sie diese hier finden: [Dokumentation für EGit](#) und [Tutorials für EGit](#)

Git-Repository für Einzel- und Gruppenarbeiten

Einzel- und Gruppenarbeiten erhalten unterschiedliche Git-Repositories zugeteilt mit unterschiedlichen an die jeweilige Situation angepassten Funktionen. Achten Sie entsprechend darauf, dass jeweils für die Aufgabe (und je nachdem ob es ein Einzelnen oder als Gruppe zu bearbeitende Aufgabe ist) passende Repository zu nützen. Beispielsweise, zählt im Rahmen des freiwilligen Gruppenprojektes nur die `commits` im `master` Branch des für Ihre Gruppe erstellten Repositories. Abgaben (`commits`) in dem für Sie erstellten Repository für die Einzelarbeiten sind dann nicht relevant für diesen Teil der Übung (und umgekehrt). Gruppenarbeiten sind nur möglich wenn diese individuell von der LV-Leitung genehmigt und abgesprochen wurden.

Übernahme von Teillösungen aus externen Quellen

Implementieren, konzipieren und erstellen Sie Algorithmen, Architekturen etc. (z.B. zur Wegfindung sowie zur Kartengenerierung bzw. Validierung) selbst um einen maximalen Lernerfolg zu erzielen. Für einfache „Kopien“ vorgegebener Algorithmen, Architekturen, Ideen etc. aus anderen Quellen können je nach Eigenleistung keine/kaum/weniger Punkte vergeben werden.

Immer wenn Source Code, Architekturen, Konzepte, Ideen etc. aus diversen Quellen (inklusive Ihrer Kommilitonen aus allen Semestern) ganz oder in Teilen kopiert/übernommen wird ist die Quelle anzugeben. Erfolgt dies nicht wird es zu einer „X“ Bewertung führen. Beachten Sie, dass Ihre Abgaben automatisch auf Plagiate geprüft werden! Passend hierzu wurde von der Universität Wien folgendes festgelegt: *"Alle fremden Gedanken, die in die eigene Arbeit einfließen, müssen durch Quellenangaben belegt werden."*

Entsprechend gilt: Falls Sie Teile oder komplette Algorithmen, Architekturen, Ideen etc. kopieren/übernehmen (auch bei Pseudocode und Konzepten) wollen, geben Sie immer die Quelle als Kommentar im Dokument bzw. Source Code an. Sprechen Sie diesen Umstand selbst beim Abgabegespräch aktiv an. Sie können dann trotzdem noch Teilpunkte für die Integration der übernommenen Lösungen erhalten.

Die Zitierregeln für Textdokumente sollten bekannt sein. Die Angabe der Quellen im Source Code hat wie folgt zu erfolgen:

1. Am Beginn der Java Datei direkt nach der Klassendefinition mit folgendem Text: **„TAKEN FROM <X>“**, dann 1-2 Sätze die übernommene Teile beschreiben und die Quelle (z.B. als Link auf eine Webseite) anführen.
2. Vor Beginn und nach Ende des übernommenen Source Codes, Konzeptes, Teils etc.

Beginn: Einleitender Text **„TAKEN FROM START <X>“**

Ende: Einleitender Text **„TAKEN FROM END <X>“**

Nummerierung: <X> bitte mit einer eindeutigen Nummer/Anmerkung ersetzen damit ersichtlich ist wo welcher kopierte Teil anfängt und aufhört und zu welcher Beschreibung am Beginn der Datei dieser Teil gehört.

Im Rahmen der Übung erreichbare Bonuspunkte

Bonuspunkte werden für Leistungen vergeben welche über das geforderte Minimum hinaus erbracht werden. Auch bei den Bonuspunkten wird die Qualität und Sinnhaftigkeit der Umsetzung berücksichtigt sodass unter Umständen nicht alle Bonuspunkte angerechnet werden können. Die Bonuspunkte aus dem Vorlesungsteil sind von der Übung unabhängig. Die LV bietet Ihnen bei folgenden Abgaben die Möglichkeit Bonuspunkte zu erreichen:

Teilaufgabe 1

- Keine

Teilaufgabe 2

- **Selbstorganisation:** Einhaltung der Minideadlines (Implementierungsfortschritt), einsehbar unter <http://swe1.wst.univie.ac.at/>. Der Fortschritt wird automatisch vom bereitgestellten Server erhoben sobald Ihr Client mit diesem kommuniziert - egal wie diese Kommunikation zustande kommt. Maximal 1 Bonuspunkt basierend auf den rechtzeitig fertiggestellten Features (Anteilsmäßig, erst ab und inkl. des 3. Features/Smileys je 0,25 Bonuspunkte).
- **Interface:** Aufwändigere *grafische* Oberfläche des Clients. Hierzu muss die automatische Evaluierung auf <http://swe1.wst.univie.ac.at/> während des Abgabegesprächs erfolgreich durchlaufen werden und die grafische Visualisierung des Spielgeschehens während des Abgabegesprächs funktionsfähig sein. Einfache grafische Oberflächen (z.B. farbige Kartendarstellungen) erhalten maximal einen Bonuspunkte. Aufwändigere (z.B. auch Icons zur Kartendarstellung) maximal zwei Bonuspunkte.
- **Tournament:** Wenn Sie mir Ihrer eigenen Clientimplementierung am Tournament aktiv teilnehmen erhalten Sie bis zu einem Bonuspunkte.

Teilaufgabe 3

- **TDD:** Für die sinnvolle und nachvollziehbare Umsetzung von Test Driven Development (TDD) kann maximal bis zu einem Bonuspunkt erreicht werden.
- **Vollständigkeit:** Für die Umsetzung zusätzlicher Funktionen können Abhängig von der Umsetzungsvollständigkeit maximal zwei Bonuspunkte erreicht werden.
 - Falls Teilaufgabe 3 als Einzelprojekt umgesetzt wird: Hinzufügen des Bewegungsendpoints.
 - Falls Teilaufgabe 3 als Gruppenprojekt umgesetzt wird: Webbasierte Visualisierung aktuell laufender Spiele (Karte, Bewegungen, Farben, aufgedeckte Bereiche, etc.), automatische Umschaltung auf das nächste Spiel, etc.
- **Business Rules:** Für die korrekte Umsetzung zusätzlicher valider und in die Ausführungen eingebundenen Businessrules welcher über das verlangte Minimum hinausgehen können jeweils 0,2 Bonuspunkte erreicht werden (bis zu einem Maximum von 1,2 für sechs oder mehr).

Auswirkung der Übungs-Deadlines

Für die Übungsabgaben werden *mehrere Deadlines* angeboten. Wir empfehlen direkt immer den *früheren Termin anzupeilen*. Alle nachfolgenden Deadlines sind dafür gedacht Sie im Falle eines *kurzfristigen unerwarteten* Ereignisses kurz vor der Endabgabe zu *unterstützen*. Beispielsweise, wenn Sie in *letzter Minute* einen kleinen, aber kritischen, Fehler entdecken oder unerwartete Schwierigkeiten beim Upload in GitLab haben.

Um Sie zusätzlich zu motivieren früher mit der Bearbeitung zu beginnen wirkt sich die gewählte Deadline auch auf die Beurteilung aus. Zumeist indem die Anzahl der maximal zu erreichenden Punkte bei einer früheren Deadline höher ausfällt als bei einer späteren Deadline.

Die finalen Punkte errechnen sich hierbei wie folgt: Annahme die frühere Deadline ergibt maximal 10 Punkte und die spätere Deadline maximal 9 Punkte. Wenn eine Ausarbeitung 80% des geforderten Minimums hinsichtlich Qualität und Umfang erreicht dann ergibt das: Frühere Deadline $10 \cdot 0,8 = 8$ Punkte. Spätere Deadline $9 \cdot 0,8 = 7,2$ Punkte. Die Auswirkung der Deadline ist also *relativ zur erbrachten Leistung*.

Sie können durch Ihr Nutzungsverhalten von Git/GitLab selbst steuern welche Deadline für Ihre Abgaben herangezogen wird. Details dazu finden Sie in den Angaben.