

Rahmenbedingungen

Übersicht

Ziel des Leistungsnachweises des Praktikums ist die Entwicklung einer Steuerungsapplikation für eine gegebene Spielsimulation entlang mehrteiliger Aufgabenstellungen. Die Simulation erhalten Sie als Abhängigkeit für Ihre Implementierung (.jar-Datei) und binden diese in Ihr Projekt mit ein. Weiterhin erhalten Sie die Dokumentation der Implementation als JavaDoc (HTML).

Bearbeiten Sie die Aufgaben in fixen Gruppen von 3 - 5 StudentInnen, die Sie der DozentIn mit Erhalt der ersten Aufgabe bekanntgeben.

Grundlage für die Bewertung ist Gesamtergebnis der Umsetzung, die Abgabe findet als Projekt-Export in ILIAS statt, der konkrete Abgabetermin wird separat bekanntgegeben. Der Leistungsnachweis wird als Gruppennote benotet und fließt zu 30% in die Modulnote ein. Bewertet wird entlang folgender Kriterien:

- Ausführbarkeit der Applikation
- Dokumentation / Einsatz von JavaDoc
- Code-Organisation und Lösungsansätze

Insgesamt gibt es fünf Aufgabenteile, die jeweils wie folgt aufgebaut sind:

Vorbereitende Aufgaben

Der erste Teil beinhaltet generelle Aufgaben, die Java- und Programmiertechniken unabhängig ob der Spielsimulation wiederholen bzw. vertiefen. Die Bearbeitung dieser ist optional, fließt folglich nicht mit in die Bewertung ein und ist nicht abzugeben.

Empfehlung: Bearbeiten Sie diese Aufgaben einem separaten Java-Projekt.

Prüfungsrelevante Aufgaben

Der zweite Teil beinhaltet prüfungsrelevante Aufgaben, deren Umsetzung bewertet wird. Diese Aufgaben bauen teilweise auf vorherigen Aufgaben auf oder führen neue Konzepte / Konstrukte ein. Ziel ist die Entwickler einer Applikation, folglich ist als Leistungsnachweis auch nur ein Projekt pro Gruppe abzugeben.

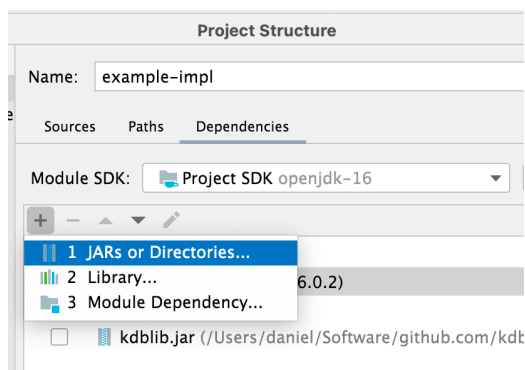
Technische Voraussetzungen

Sie benötigen eine aktuelle Entwicklungsumgebung für Java (bspw. JetBrains IntelliJ, Eclipse IDE, Netbeans) sowie ein Java Development Kit (JDK) in der Version ≥ 16 , nach Möglichkeit jedoch 17. Sie können dies separat oder durch die Entwicklungsumgebung installieren lassen.

Um die Arbeit in Gruppen zu erleichtern, wird der Einsatz eines Versionskontrollsystems (bspw. Git mit einer Plattform wie GitHub, GitLab oder Bitbucket) empfohlen.

Projektkonfiguration mit Spielsimulation

- Erstellen Sie ein neues Java-Projekt
- Laden Sie die .jar-Datei der Simulation in Absprache mit der DozentIn herunter und platzieren Sie diese in Ihrem Projekt
- Richten Sie die .jar-Datei als Abhängigkeit im Classpath ein:
 - **Hier exemplarisch mit IntelliJ:**
 - Öffnen Sie die Projekteinstellungen (*Project Structure*)
 - Wählen Sie den Tab *Modules* und dort das Modul Ihrer Applikation (meist ist dort nur eines vorhanden)
 - Wählen Sie den Tab *Dependencies*, klicken Sie auf das Plus-Symbol, wählen Sie *JARs or Directories ...* und wählen Sie die .jar-Datei aus



- Nach Bestätigung der Einstellungen können Sie die Klassen aus der .jar-Datei in Ihrem Projekt benutzen.

Szenario

Sie entwickeln ein Spiel, bei dem ein imaginäres Softwareentwicklungsunternehmen gesteuert wird. Auf der nächsten Seite finden Sie das UML-Klassendiagramm der Domäne. Das Spiel funktioniert entlang der folgenden Regeln:

- Das Spiel ist rundenbasiert, Sie entwickeln einen *Handler*, der pro Runde aufgerufen wird, den Spielstatus analysiert und Operationen zur Manipulation anbietet
- Die Spielsteuerung ist textbasiert und interaktiv
- Beim Starten des Spiels werden zufällige Daten (Entwickler, Büro, Vermögen, ...) generiert
- Pro Runde werden Kosten (bspw. Miete und Löhne) vom Vermögen des Unternehmens abgezogen. Sollte Ihr Unternehmen in einen negativen Kontostand laufen, wird das Spiel beendet
- Pro Runde werden ggf. anstellbare EntwicklerInnen und Projektangebote generiert
- Es können Projekte zur Bearbeitung akzeptiert werden und EntwicklerInnen im selben Büro zugewiesen werden.

Pro Runde wird vom Aufwand eines Projekts die Skills der zugewiesenen EntwicklerInnen abgezogen. Sobald alle Aufwände auf 0 reduziert wurden, wird das Projekt als erfolgreich abgeschlossen und der Erlös dem Vermögen gutgeschrieben. Sofern das Projekt zur Deadline nicht abgeschlossen wurde, wird es entfernt. Projektabschlüsse werden am Anfang jeder Runde vor dem Bezahlen von laufenden Kosten durchgeführt

- Pro Runde können Sie drei manipulative Aktionen durchführen. Diese sind jeweils im *GameDevStudio* implementiert und umfassen aktuell das Anstellen einer EntwicklerIn und Akzeptieren eines verfügbaren Projekts.

Bei der vierten Aktion wird mit einem Fehler abgebrochen.

