

## Übungsblatt 1: Test prozedural & Semesterspiel

180 Punkte möglich

### Aufgabe 1: Der schriftliche Test in Einzelarbeit

[90 Punkte] Sie sind pünktlich zum Test-Termin anwesend (s. Kalender) und nehmen an einem schriftlichen Test teil, der alle Inhalte bis zu diesem Tag gemäß Kalender umfasst. Insbesondere werden die PR1-Vorkenntnisse in prozeduraler Programmierung geprüft. Ein schlechtes Abschneiden in diesem Test hat zur Folge, dass Ihre individuellen Leistungen im Semesterspiel bis zum nächsten Test stärker geprüft werden! Bis zum ersten Test werden stets die Teilnehmer zuerst gefragt, die formell vorhandene Kompetenzen noch nicht nachweisen konnten, insbesondere aus PR1.

### Aufgaben 2 bis 5: Das Semesterspiel als Teamarbeit

In diesem Semester werden Sie das Spiel „Wizard“ als Java-Anwendung für einen Kunden Ihres Unternehmens (=Teams) zu realisieren. Im Moodlekurs finden Sie die Spielanleitung zu dem Spiel als Ausgangsbasis, die in enger Absprache mit dem Kunden umzusetzen ist. Varianten sind (noch) nicht zu implementieren.

Die Rolle des Vertrauten übernimmt Ihre Java-Anwendung. Zunächst finden sich die Spieler ein. Ab 3 Spielern kann das Spiel gestartet werden, mehr als 6 Spieler dürfen es nicht sein. Ist das Spiel einmal gestartet, darf kein weiterer Spieler hinzukommen oder das Spiel verlassen.

Die Reihenfolge der Spieler beim Vorhersagen und beim Ausspielen ergibt sich in der ersten Runde aufsteigend anhand ihrer ID. Jede Runde beginnt mit dem Austeilen der Karten für diese Runde vom Kartenstapel. Danach schaut sich jeder Spieler seine Karten an und trifft seine Vorhersage. Nun beginnt der Kampf um den Stich dieser Runde. Nach der Beendigung der Runde behält der Spieler den Stich, der ihn gewonnen hat und das Spiel trägt die Blockzeile ein.

Es sind keine weiteren oder anderen Regeln ohne dokumentierte Absprachen mit dem Kunden zu implementieren. Erfragen Sie beim Kunden, wie viele JUnit-Tests er bei welcher Klasse wünscht.

Sollten Sie noch Fragen zu den Regeln haben, stellen Sie diese bitte *rechtzeitig* im Kundenforum! Nur diese Antworten des Kunden sind dann für alle Teams verbindlich. Interpretieren Sie Regeln anders als der Kunde a la „wir haben das aber so verstanden“, so erhalten Sie entsprechende Punktabzüge. Generell gilt:

*Der Kunde hat immer Recht! Ihr Team ist Dienstleister des Kunden, er bezahlt Sie!*

*Die anderen Teams sind Ihre Konkurrenten:  
Jede Weitergabe von Codes oder von Informationen  
führt zum sofortigen Ausschluß aller beteiligten Teams von dieser Studienleistung!*

Die Sprache in allen Quellcodes und in allen Dokumentationen der gesamten Veranstaltung ist Deutsch.

Im Folgenden sehen Sie ein UML-Klassendiagramm, welches bereits vom Kunden erstellt wurde. Diese Klassen, Methoden und Beziehungen sind genauso zu verwenden! Erweiterungen und Änderungen erfolgen nur nach rechtzeitiger vorheriger Absprache und Genehmigung durch den Kunden, wobei die Absprache schriftlich zu dokumentieren ist.

Die überschriebene toString-Methode des Spiels gibt den vollständigen aktuellen Zustand des Spiels zurück, der über einen CLI -Menüeintrag jederzeit abrufbar ist.

Sowohl das gesamte Konsolenmenü, als auch jede Ausgabe auf der Konsole hat zentral durch die Klasse SpielCLI zu erfolgen. Der Zugang zum Spiel hat ausschließlich zentral über die Spiel-Klasse als Fassade<sup>1</sup> zu erfolgen, die SpielCLI darf keine weiteren internen Klassen des Spiels kennen.

---

<sup>1</sup> [https://de.wikipedia.org/wiki/Fassade\\_\(Entwurfsmuster\)](https://de.wikipedia.org/wiki/Fassade_(Entwurfsmuster))



## Aufgabe 2: Das Semesterspiel als Teamarbeit - Ein Projekt verteilt entwickeln

[2 Punkte] Loggen Sie sich zuerst einmalig in Gitty ein unter <https://gitty.informatik.hs-mannheim.de/user/login>. Der Benutzername ist ihre Matrikelnummer und das Passwort ist dasselbe wie beim Zugriff auf Moodle. Schreiben Sie danach eine Mail an [m.koehler@hs-mannheim.de](mailto:m.koehler@hs-mannheim.de), dass Sie eingeloggt waren. Herr Köhler weist Sie dann Ihrem Team-Repo zu.

Daraufhin erhalten Sie eine Mail mit einer URL. Diese URL ist in Eclipse zu integrieren. Die Authentifikation in Eclipse erfolgt dann wieder mit Ihrer Matrikelnummer und Ihrem Moodle-Passwort.

Richten Sie das Projekt „Semesterspiel“ in Eclipse so ein, dass es über Git mit allen Teilnehmern Ihres Teams synchronisiert wird.

Verwenden andere Teams dieselbe Code-Basis ohne ausdrückliche Genehmigung von Prof. Dopatka, so werden alle Mitglieder der beteiligten Teams ohne weitere Diskussion für dieses Semester ausgeschlossen.

Weisen Sie nun dem Kunden nach, dass Sie Gitty im Team beherrschen incl. Merging. Es ist lediglich ein einziger Master-Branch im Repository erlaubt, in dem alle Mitglieder des Teams arbeiten. Vor allen folgenden Abgaben muss die aktuelle Version Ihrer Software in Gitty vorhanden sein, der Kunde schaut sich Ihren Code ab jetzt nur noch via Gitty an! Jede Verteilung von Code außerhalb von Gitty gilt als Täuschungsversuch!

Nach der Deadline jedes Übungsblatts wird die Gitty-Aktivität jedes Teammitglieds geprüft. Sind Personen passiv oder deutet die Aktivität auf Betrugsversuche hin, so kann eine Nachprüfung, ein individueller Punktabzug und/oder ein Ausschluss von dieser Studienleistung erfolgen.

**Solange Sie diese Aufgabe nicht vollständig abgeben, können Sie in den folgenden Aufgaben keine Punkte im Semesterspiel erhalten! Erledigen Sie diese Aufgabe also möglichst schnell!**

## Aufgabe 3: Das Semesterspiel als Teamarbeit - Der Kern der Fachlogik

3a) [18 Punkte] Erstellen Sie alle Klassen des UML-Klassendiagramms testgetrieben unter Verwendung von JUnit4-Tests, welche Sie zur Dokumentation abgeben.

3b) [50 Punkte] Implementieren Sie die volle Funktionalität des Spiels gemäß der Regeln. Weisen Sie dem Kunden durch Erstellung von mehreren realen Szenarien nach, dass Ihr Spiel gemäß der Regeln funktioniert. Erstellen Sie dazu eine passende Menüführung. Die gesamte Interaktion mit dem Benutzer incl. Debug-Ausgaben zum aktuellen Stand des Spiels und der Zustände der Spieler, z.B. wer hat noch welche Karten, ist zu präsentieren.

## Aufgabe 4: Das Semesterspiel als Teamarbeit - Fehlermanagement

[10 Punkte] In allen Fehlerfällen ist ab jetzt ein Objekt der eigenen Klasse SpielException zu werfen, die von Exception erbt und die neben der Fehlermeldung eine Fehler-ID besitzt. Eine Liste der Fehler-IDs ist Gegenstand der Dokumentation.

Leiten Sie davon sinnvolle Unterklassen ab, wie beispielsweise eine SpielEingabeException.

## Aufgabe 5: Das Semesterspiel als Teamarbeit - Dokumentation

[10 Punkte] Kommentieren und dokumentieren Sie alle Quellcodes aus Aufgaben 3 und 4 vollständig und ausführlich in JavaDoc. Integrieren Sie die JavaDoc in Ihren Workspace und in Gitty.

Ab jetzt ist für die weiteren Übungsblätter jede Klasse vollständig und ausführlich in JavaDoc zu dokumentieren, ansonsten werden die Punkte der jeweiligen Teilaufgabe um bis zu 50% reduziert!

### **WICHTIG:**

Die Verwendung von Lösungen von anderen Personen sowie die Absprache mit anderen Personen außerhalb des Teams ist nicht gestattet und gilt als Täuschungsversuch! Es wird nochmals darauf hingewiesen, dass jedes Teammitglied die Lösung vollständig verstanden haben muss! Das individuelle Testen dieser Kenntnis wird vorbehalten!

Es ist notwendig, das bereitgestellte Vorlesungsskript vorzuarbeiten, um bei den Live-Terminen rechtzeitig offene Fragen klären zu können. Ansonsten wird die vollständige Fertigstellung der Aufgaben unter Umständen sehr schwierig! Arbeiten Sie ab jetzt ständig, eigenhändig und konzentriert!

Die Aufgaben können als Grundlage für weitere Übungsblätter dienen! Arbeiten Sie daher besonders sorgfältig, denn Sie müssen den Rest des Semesters damit leben! Wenn Sie in diesem Blatt nicht alle Anforderungen realisieren und eine Aufgabe im nächsten Übungsblatt baut darauf auf, so haben Sie diese Anforderungen bis zur Abgabe des nächsten Übungsblattes nachzuliefern! Fehlt dann diese Anforderung immer noch, so gibt es erneut Punktabzug!