2. Juni 2022 (Update: 14.6.2022)

Bewertungskriterien PiS-Projekt, SoSe 2022

Die Grundausrichtung der Bewertung der PiS-Prüfungsleistung steht. Aber beachten Sie: Im Augenblick hat dieses Dokument Entwurfscharakter, rechnen Sie noch mit Änderungen. Spätestens zwei Wochen vor der Abgabe wird es keine Änderungen mehr geben.

Zielsetzung

Aufgabe ist es, eine Animation, eine Simulation oder ein Strategiespiel umzusetzen. Ziel ist es zu zeigen, dass Sie eine frei gewählte Aufgabe alleine und eigenständig gemäß den in der Veranstaltung vermittelten Inhalte umsetzen können und die durch die Bewertungskriterien erfassten Kompetenzen erworben haben.

Rahmenbedingungen

Die Anwendung muss einen algorithmischen, rechenintensiven Anteil haben. Die Anwendung muss zudem interaktiv sein (Beispiel: Spiel) bzw. interaktiv parametrisierbar sein (Beispiel: Simulation). Die Anwendung besteht strukturell aus zwei Anteilen, die sich im Programmcode eindeutig widerspiegeln: Da ist zum einen die Anwendungslogik (AL), die prinzipiell als eigenständige JShell-Anwendung ausführbar wäre. Und da ist zum anderen die Interaktionslogik (IL), die einer Anwenderin bzw. einem Anwender eine graphisch-orientierte Interaktion mit der Anwendung umsetzt und mit Processing realisiert ist. Die Interaktionslogik darf ausschließlich über ein Interface auf die Anwendungslogik zugreifen. Das Interface ist als Teil der Anwendungslogik zu betrachten.

- Es kommt mind. Java 17 zum Einsatz
- Sie verwenden das aktuelle Processing 4 bzw. die extrahierte core.jar-Datei
- Es steht Ihnen frei, verschiedenste Bibliotheken (z.B. JUnit, log4j) oder Werkzeuge (gradle, maven) zu nutzen; sprechen Sie uns im Zweifel an
- Der Einsatz einer Datenbank ist nicht gestattet
- Nim und Tic-Tac-Toe sind als Projektideen nicht zulässig

Beachten Sie auch die Hinweise aus dem Dokument »Rückmeldung zur Projektskizze«.

Abgabe und Abnahme

Da wir Ihren Code nicht auf unseren Rechnern ausführen (dazu sind die Bedingungen nicht standardisiert genug, im Gegenzug haben Sie mehr Freiheiten), kommt zu der Abgabe Ihres Projekt eine Abnahme hinzu, bei der Sie uns Ihre lauffähige Anwendung vorführen. Bereiten Sie diese Demonstration sehr gut vor, da wir zeiteffizient arbeiten müssen.

Abgabekriterien

Anzahl der Lines of Code (LOC)

Die Klassen, Enums, Records und Interfaces der AL sind in einer einzigen, separaten Programmdatei unterzubringen. Die Codezeilen werden mit cloc gezählt. Zur Code-Formatierung gilt im Zweifel der Google Java Style Guide. Test-, Konfigurations-, Medien- und Datenhaltungsdateien werden nicht mitgezählt. Unlautere Maßnahmen zur Erhöhung der Codemenge führen zu 0 Punkten.

<= 80	**> 80, <= 160**	**> 160, <= 240**	**> 240, <= 320**	**> 320, <= 400**	**> 400**
❏ 0.0	❏ 0.5	❏ 1.0	❏ 1.2	❏ 1.0	❏ 0.0

Hinweis: Die Zählung der LOC bezieht sich ausschließlich auf die AL und nicht die IL. Sie dürfen für die IL soviel Code produzieren, wie Sie wollen. Die Maßnahmen, die Sie zum Programmdesign umsetzen, sind Teil der AL.

Programmdesign (PRD)

Das Programmdesign bezieht sich auf das bereitgestellte AL-Interface und auf in den Methoden der AL-Klassen untergebrachten Vorbedingungen [siehe Moodle].

Ein als »sehr gut« bewertetes AL-Interface hat die ausgeführten Eigenschaften. Ein »gut« bewertetes Interface setzt nicht alle, aber viele der Punkte um; ein »mangelhaftes« Interface beachtet vor allem die fett gesetzten Punkte unzureichend.

- ☐ Es werden für die Aufruf- und Rückgabewerte immutable Datentypen verwendet
- ☐ Es gibt keine “Referenz-Lecks”, die Objekte der AL von außen manipulieren lassen
- ☐ Das Interface ist als fluides Interface angelegt
- ☐ Das Interface macht von `default`-Implementierungen Gebrauch
- ☐ Das Interface bietet **keine redundanten Methoden** an
- ☐ Es werden **nicht** alle Klassenmethoden einfach ans Interface “**durchgereicht**”
- ☐ Das Interface ist zweckmäßig **schlank** bzgl. der AL und zugleich **praktikabel** bzgl. der IL
- ☐ Unabdingbar: Die **IL programmiert gegen das AL-Interface** statt gegen die AL-Klassen

Vorbedingungen sind mit `assert`-Anweisungen in den Methoden der AL-Klassen umzusetzen. Die `assert`-Anweisungen legen fest, welche Bedingungen die Methoden-Parameter erfüllen müssen, damit die Implementierung der Methode die Voraussetzungen vorfindet, um ihre Funktion bzw. Aufgabe erfüllen zu können. Eine »gute« Umsetzung liegt vor, wenn die Kriterienliste weitgehend erfüllt ist.

- ☐ Die Zusicherungen überprüfen umfassend die notwendigen Voraussetzungen, die mögliche Argumentwerte erfüllen müssen
- ☐ Mögliche Abhängigkeiten von internen Objektzuständen sind berücksichtigt
- ☐ Es gibt keine redundanten Überprüfungen (Beispiel: Methode profitiert von `assert`-Anweisungen in Hilfsmethode)
- ☐ Umfangreichere Überprüfungen sind gegebenenfalls in Hilfsmethoden ausgelagert
- ☐ Aufzählungstypen (`enums`) für Parameter sind `assert`-Anweisungen vorzuziehen, wenn sich damit die Anzahl der Vorbedingungen reduzieren lässt
- ☐ Die Intention einer assert-Anweisung ist angegeben; Beispiel:
  `assert !board.isEmpty() : "Spielbrett muss noch mind. eine Figur haben";`

AL-Interface	Vorbedingungen
❏ 0.5 (sehr gut) ❏ 0.3 (gut) ❏ 0.0 (mangelhaft)	❏ 0.5 (»gute« Umsetzung) ❏ 0.0 (unzureichende Umsetzung)

Validierung (VAL)

Die Validierung umfasst das Testen und das Monitoring [siehe Moodle]. Es liegen in einer separaten Datei einerseits 5 Tests vor, die die AL über das Interface testen, und andererseits 5 weitere Tests, die die AL auf Klassenebene testen. Die Tests sind entweder durchgängig als `assert`-Anweisungen zu formulieren oder mittels JUnit umzusetzen. Das Monitoring muss ein-/ausschaltbar sein und macht nachvollziehbar, dass der algorithmisch-rechenintensive Anteil für ein zwar einfaches aber hinreichend belegkräftiges Anwendungsszenario arbeitet wie gedacht. Für das Logging können einfache bedingte `print`-Anweisungen oder Log4j verwendet werden.

Tests gelten als »gut«, wenn sie folgende Kriterien erfüllen:

- ☐ Die Tests testen nicht nur triviale Verhältnisse ab
- ☐ Die Tests sind variantenreich, d.h. sie decken sehr unterschiedliche Szenarien ab
- ☐ Es wird nichts getestet, was durch geeignete Vorbedingungen abgedeckt werden könnte
- ☐ Mit jedem Test wird die Testabsicht im Code dokumentiert (bei `assert`, siehe oben)

Ein Monitoring gilt als »gut«, wenn folgende Kriterien erfüllt sind:

- ☐ Das Logging beschränkt sich auf das absolut Notwendige an Ausgaben
- ☐ Das Logging ist so aufbereitet, dass die Ausgabe verständlich und übersichtlich ist
- ☐ Die Logging-Ausgabe erlaubt es, den Algorithmus in seiner Arbeitsweise nachzuvollziehen
- ☐ Die Logging-Ausgabe macht plausibel, dass der Algorithmus korrekt arbeitet

Tests			Monitoring		
gut	ausreichend	mangelhaft	gut	ausreichend	mangelhaft
❏ 0.6 (JUnit) ❏ 0.5 (assert)	❏ 0.3 (JUnit) ❏ 0.25 (assert)	❏ 0.0	❏ 0.6 (log4j) ❏ 0.5 (print)	❏ 0.3 (log4j) ❏ 0.25 (print)	❏ 0.0

Dokumentation (DOK)

Mit Ihrer Anwendung ist eine Dokumentation als README abzugeben, die die nachstehenden Inhalte hat. Sie bekommen ein Template für das README zur Verfügung gestellt. Formulieren Sie grundsätzlich in vollständigen Sätzen und schreiben Sie den Text für Lesende, die kein Vorwissen zu Ihrem Projekt haben.

- ❏ Kurzbeschreibung der Anwendung (50-150 Wörter) samt Screenshot zur Anwendung. Dokumentiere Sie zudem verwendete Quellen, die Sie zur Umsetzung herangezogen haben.
- ❏ Beschreibung: Welcher Berechnungsalgorithmus kommt zum Einsatz? Dazu eine kurze Erklärung, wie der Algorithmus funktioniert.
- ❏ Erklären Sie anhand eines Szenarios, inwiefern das beispielhaft beigefügte Logging-Protokoll die korrekte Umsetzung des Berechnungsalgorithmus' belegt

Diese einzelnen Anteile sind in der Tabelle aufgeführt und werden addiert. Sollte der betreffende Dokumentationsaspekt von unzureichender Qualität sein (unvollständig oder unverständlich) oder gar fehlen, dann gibt es keine Teilpunkte dafür.

Kurzdarstellung + Screenshot + Quellenbelege	Beschreibung des Algorithmus'	Erklärung des Logging-Protokolls
❏ 0.2	❏ 0.4	❏ 0.4

Wenn Sie Quellen verwendet aber nicht dokumentiert haben, so gilt das als Täuschungsversuch und zieht, wenn wir das belegen können, entsprechende Konsequenzen nach sich.

Abnahmekriterien

Interaktionsdesign (IND)

Das Interaktionsdesign gilt nach den folgenden Kriterien als »gut«. Werden diese Erwartungen übererfüllt, ist es »sehr gut«. Erfüllt das Design die Kriterien nur bedingt, gibt es ein »ausreichend«.

- ☐ Die graphische Oberfläche ist ansprechend gestaltet und funktional bzgl. der Anwendung
- ☐ Die Bedienung der Anwendung ist gleichsam selbsterklärend
- ☐ Es kommt zu keinen Fehlern oder Abstürzen, wenn die Bedienung absichtlich Fehler zu produzieren versucht

sehr gut	gut	ausreichend	mangelhaft
❏ 1.2	❏ 1.0	❏ 0.5	❏ 0.0

Hinweis: Der Programmcode zur IL fließt nicht in die Bewertung ein; es wird bei der Abnahme lediglich überprüft, ob die IL tatsächlich gegen das AL-Interface programmiert ist.

Gesamteindruck (GES)

All die aufgeführten Punkte können nicht den Gesamteindruck ersetzen, den Ihre Umsetzung vermittelt:

- ❏ Löst die Anwendung ihr Funktionsversprechen ein? Das Funktionsversprechen ist durch Ihre Dokumentation (siehe DOK) definiert.
- ❏ Wie ist die Qualität der Arbeit insgesamt einzuschätzen? Das heißt u.a.: Klassen und Interfaces werden genutzt; kein unbegründetes `static`, kein `print` (außer ev. Logging); keine Code-Duplikationen; Lambda-Ausdrücke und Streams kommen gelegentlich zum Einsatz; geeignete Datenstrukturen werden verwendet.
  - ❏ viele Fehler, wenig Bezug zu Veranstaltungsinhalten, sachlich/inhaltlich deutlich verbesserungswürdig: »schwach«
  - ❏ es wurde sauber gearbeitet, Bezug zu Veranstaltungsinhalten ist erkennbar, Code und Doku sind verständlich: »gut«
  - ❏ Code und Doku sind exzellent aufgearbeitet, deutlicher Bezug zu Veranstaltungsinhalten, problemlose Orientierung: »sehr gut«

Funktionsversprechen (F)	Qualität (Q)	GES = (F + Q) / 2
❏ 1.0 (ja) ❏ 0.5 (so lala) ❏ 0.3 (nein)	❏ 1.0 (sehr gut) ❏ 0.75 (gut) ❏ 0.35 (schwach)	(mind. 0.5)

Der Gesamteindruck ergibt sich als Mittelwert der Teilaspekte. Sie müssen hier mindestens einen Wert von 0.5 erreichen, damit die Werte/Faktoren aus den anderen Kategorien verrechnet werden. Erreichen Sie den Mindestwert im Gesamteindruck nicht, wird Ihre Prüfungsleistung mit 0 Punkten bewertet.

Bewertung

Punkte = 100 × GES × (LOC + PRD + VAL + DOK + IND) / 5

(Anmerkung: Diese Prüfungsleistung ist auf 100 Punkte normiert, sie wird aber in der Gesamtleistung für PiS hälftig mit dem Klausurergebnis verrechnet. Wenn Sie hier mehr als 100 Punkte erreichen, werden Ihnen max. 110 Punkte gutgeschrieben in Verrechnung mit der Klausurleistung.)

Es gelten die Regeln des kaufmännischen Rundens. Übrigens entscheiden wir uns bei der Bewertung immer für einen in der jeweiligen Tabelle angegebenen und dokumentierten Wert. Es ist z.B. nicht so, dass wir beliebige "Zwischenwerte" nehmen.

Anmerkungen zur Bewertung

Wir haben die Bewertungskriterien mit Bedacht ausgewählt. Das Bewertungsschema soll Ihnen in angemessener Weise transparent und klar machen, wie Sie an Ihre Wunschnote kommen können und was Sie dafür einlösen müssen. Es gibt durch die Bewertung des Gesamteindrucks einen Korridor der Unsicherheit, aber auch das können Sie durch die anderen Faktoren durchaus eingrenzen.

Sprechen Indizien für einen Täuschungsversuch, so haben Sie nicht bestanden, der Fall wird dokumentiert und dem Prüfungsausschuss angezeigt.

Eigenständigkeitserklärung

So sieht die Erklärung aus, die zusammen mit Ihrer Abgabe zu leisten ist:

Hiermit bestätige ich, dass ich die vorgelegte Projektarbeit eigenständig und ohne fremde Hilfe erstellt habe. Ich erkläre explizit, dass ich kein Plagiat begangen habe, d.h. dass ich keinen Code/Text aus nicht dokumentierten Quellen verwendet habe und dass nicht umfängliche Teile des Codes (>= 20%) eine "Kopierleistung" sind. Quellen, die ich benutzt habe, sind vollständig in der Dokumentation angegeben. Ich bestätige außerdem, dass die Projektarbeit nicht vor dem SoSe 2022 entstanden ist und noch nirgends als Prüfungs- oder Zulassungsleistung (wie z.B. zur Klausurzulassung) vorgelegt worden ist. Mir ist klar, dass die abgegebene Prüfungsleistung für PiS annulliert wird, sollte ich keine Zulassung für PiS besitzen.

Zeithorizont

Damit Sie einen groben Überblick über den zeitlichen Verlauf haben, hier eine Übersicht. Beachten Sie die Termine für Abgaben in Moodle.

VW8	Feststellung der Zulassung zur Prüfungsleistung Start der Projektarbeit (3.6.2022) mit Bekanntgabe der Bewertungskriterien
VW9	Abgabe einer Projektskizze in Moodle, Begutachtung Ihrer Skizze
VW10	Rückmeldungen zur Projektskizze, Ende Frist zur verbindlichen Prüfungsanmeldung
VW11	Vorstellung des Projektstands im Praktikum, Anwesenheitspflicht
VW12	Vorstellung des Projektstands im Praktikum, Anwesenheitspflicht

VW13 Vorstellung des Projektstands im Praktikum, Anwesenheitspflicht

VW14 Abgabe Ihres Projekts in Moodle, Demonstration und Abnahme am 15.7.2022