SEP/MP 2022

– Aufgabenblatt 3 –

spätester Abgabetermin: 22.06.2022 12:00Uhr

Ziel des Blattes

Auf dem letzten Aufgabenblatt wurde die Entwurfsphase abgeschlossen. Auf diesem Blatt soll nun mit der Umsetzung begonnen werden.

Aufgaben

1. Schreiben Sie das Grundgerüst Ihrer Implementierung basierend auf Ihrem Entwurfsmodell. Hierbei sollen die Interfaces/Klassen jeweils mit ihren Methodensignaturen und die Beziehungen der Klassen umgesetzt werden. Die Methoden für das eigentliche Spiel sollen hier noch nicht implementiert werden. Jedoch sollten Basisfunktionalitäten wie Anmelden/Abmelden, Spielraum erstellen/beitreten usw. schon vorhanden sein.

2. Schreiben Sie für jede Methode Java-Doc Kommentare, die darlegen, was diese Methode tun soll, welche Parameter sie erwartet, was ihre Rückgabewerte sind und welche Exceptions geworfen werden können. Sollte die Methode Anforderungen an die Parameter stellen (z.B. darf nicht "null" sein), so sind diese auch entsprechend zu dokumentieren.

3. Schreiben sie für jede öffentliche Methode Unit-Tests. Hierbei sind nicht nur die "normalen" Anwendungsfälle, sondern auch entsprechende Fehlerfälle zu testen.

4. Bereiten sie sich auf eine Zwischenabnahme vor. Hierbei sollen Sie ein Demovideo vorbereiten, aus dem ersichtlich ist, wie das bereits implementierte Systemteile funktionierten. **Bei dieser Präsentation ist eine einfache GUI mit funktionierendem Chat und den Basisfunktionalitäten aus Aufgabe 1 von Nöten.** Außerdem sollen Sie demonstrieren können wie die JavaDocs kompilieren und was sie beinhaltet als auch die Durchführung der Systemtests mit Coverage.

Hinweise

1. Für reine GUI-Klassen sind keine Unit-Tests notwendig.

2. Die meisten Ihrer Testfälle sollten zu diesem Zeitpunkt scheitern. Dies ist durchaus gewollt und demonstriert, dass der Testfall scheitern kann und nicht auch von einer falschen Implementierung erfüllt wird.

Checkliste zur Vermeidung typischer Fehler beim Dokumentieren

- ☐ Dokumentation ist wirklich vorhanden und nicht nur das Grundgerüst dafür. Und zwar für alle Methoden, außer Getters/Setters.
- ☐ Automatisch generierte Dokumentation wurde nach den letzten Änderungen aktualisiert.
- ☐ Die komplette Dokumentation ist sinnvoll.

Checkliste zur Vermeidung typischer Fehler bei den Unit-Tests

- ☐ Unit-Tests sind keine Integrationstests. (Alle Units werden von anderen isoliert getestet[1].)
- ☐ *BeforeClass* wird richtig verwendet.
- ☐ Die Funktionen der Units sind ausreichend durch Tests abgedeckt. (Es wurden genug Tests geschrieben)[2].
- ☐ Zunächst werden die Tests geschrieben und erst danach die Implementierung. Das ist Pflicht und muss durch die Commits in Ihrer Repo belegbar sein.

Kontakt

OLAT	https://olat.vcrp.de/url/RepositoryEntry/3654517002
Leitung	apl. Prof. Dr. Achim Ebert ebert@cs.uni-kl.de
Organisation	Dr. Taimur Khan tkhan@cs.uni-kl.de
Hiwis	Andreas Tomasini
	Hasan Albakkour
	Marvin Häuser
	Mail an alle sep-support@cs.uni-kl.de

[1] Das wird normalerweise mithilfe von Stubs/Mocks erledigt
[2] Zur Erinnerung: Das zukünftige Spiel soll zu ca. 70% durch Unit-Test abgedeckt sein