



Übungsblatt 6

Programmierung und Softwareentwicklung (WS 17/18)

Abgabe: Freitag, 15.12.2017, 23:55 Uhr — Besprechung: ab Montag, 18.12.2017

Bitte lösen Sie die Übungsaufgaben in **Gruppen von 3 Studenten** und wählen EINEN Studenten aus, welcher die Lösung in ILIAS als **Gruppenabgabe** (unter Angabe aller Gruppenmitglieder) einstellt. Bitte erstellen Sie dazu einen **Header**, welcher die Namen der Studenten, die Matrikelnummern und die E-Mail-Adressen enthält.

Die Aufgaben, bei denen Quellcode abzugeben ist, sind mit Impl gekennzeichnet. Bitte beachten Sie die Hinweise zu den Implementierungsaufgaben, die in ILIAS verfügbar sind.¹ Achten Sie besonders darauf, dass Sie zu jeder Klasse und Methode JavaDoc-Kommentare erstellen.

Dieses Übungsblatt beinhaltet 3 Aufgaben mit einer Gesamtzahl von 40 Punkten.

Aufgabe 1 Modellierung mit der UML [Punkte: 20]

- (a) (5 Punkte) Was ist UML? Erläutern Sie die Elemente eines UML Klassendiagramms und die Elemente eines UML Sequenzdiagramms.
- (b) (5 Punkte) Installieren Sie sich ein UML-Tool, wie zum Beispiel Papyrus für Eclipse, Visio, Draw.io oder das UMLet-Plugin für Eclipse. Modellieren Sie damit ein UML-Klassendiagramm für ein Projekt aus einer Firma mit folgenden Klassen: Firma, Projekt, Projektmanager, Teilprojekt und Mitarbeiter. Berücksichtigen Sie dabei zunächst nur die Klassen und ihre Beziehungen zueinander, d.h. noch keine Attribute oder Methoden. Die Abgabe muss entweder im PDF- oder PNG-Format erfolgen.
- (c) (5 Punkte) Entwerfen Sie ein UML-Klassendiagramm mit Attributen, Methoden und Assoziationen inkl. Multiplizitäten für das nachfolgend beschriebene System:
Die Universitätsbibliothek will eine neue Bibliothekssoftware erstellen. Die Bibliothek hat mehrere Standorte, z.B. einen in der Stadtmitte und einen in Vaihingen. Die Bibliothekssoftware soll Bücher, Zeitschriften und Videos verwalten. Zu einem Buch kann es mehrere Buchexemplare geben. Der Benutzer kann Ausleihgegenstände (Bücher, Zeitschriften, Videos) aus der Bibliothek entleihen, jedoch höchstens drei Gegenstände gleichzeitig. Zur Identifikation eines Benutzers dient eine Personenkenzahl (PK). Diese wird bei Ausleihe und Rückgabe erfragt. Der Benutzer hat auch ein Benutzerkonto mit Loginnamen und Passwort, über das Ausleihgegenstände bestellt und reserviert werden können. Die Abgabe muss entweder im PDF- oder PNG-Format erfolgen.
- (d) (5 Punkte) Erstellen Sie ein UML-Sequenzdiagramm, welches den Anmeldevorgang für die Online-Universitätsbibliothek (siehe Teilaufgabe c) modelliert. Ein Benutzer hat hierbei drei Versuche, sein Passwort korrekt anzugeben, bevor der Anmeldevorgang automatisch abgebrochen wird. Nutzen Sie im Sequenzdiagramm nur Klassen, Operationen, etc. die auch in Ihrem Klassendiagramm aus Teilaufgabe c) enthalten sind.

Aufgabe 2 Impl IceTeaMediaPlayer: Enum, States & Exceptions [Punkte: 10]

Ziel dieser Aufgabe ist, den Zustand des Mediaplayers zu speichern. Hierfür soll ein **Enum** verwendet werden. Zudem soll der Zustand des Players in der Klasse **Player** kontrolliert werden, um Fehler zu vermeiden. Hinweise: Die Dokumentation der BasicPlayer-Library ist über das ILIAS verfügbar.

- (a) (2 Punkte) Erstellen Sie eine **Enum** und nennen Sie diese **PlayerState**. Die **Enum** sollte die Zustände **NO_FILE**, **PLAYING**, **PAUSED** und **STOPPED** beinhalten.

¹https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_fold_1318001.html

- (b) (8 Punkte) Fügen Sie die Zustandskontrolle in der Klasse **Player** hinzu. Nutzen Sie hierfür ein privates Objekt von **PlayerState** um den Zustand des Players zu speichern. Der Anfangsstatus sollte hierbei **NO_FILE** sein. Implementieren Sie ebenfalls die Methode **getState**, welche den aktuellen Status des Players zurückgibt und erweitern Sie die Methoden **play** (mit allen überladenen Methoden), **skip**, **pause**, **resume**, **stop** und **stateUpdated**, sodass diese ebenfalls den Status des Players kontrollieren. So sollte beispielsweise eine **IllegalStateException** geworfen werden, wenn der Player sich nicht im Status **PLAYING** befindet, aber dennoch die Methode **pause** aufgerufen wird oder nach erfolgreichem Stoppen des Players sollte dieser in den Zustand **STOPPED** übergehen.

Aufgabe 3 Impl IceTeaMediaPlayer: Konsoleninterface [Punkte: 10]

Laden Sie das aktuelle **IceTeaMediaPlayer** Projekt aus dem ILIAS herunter. In diesem Projekt befindet sich die leere Klasse **CInterface**. Implementieren Sie in dieser Klasse ein Text-basiertes Menü für die Benutzung des Media Players, ein sogenanntes Command-Line Interface (CLI). Der Benutzer soll den gesamten Player über die Konsole (z.B durch das Eingeben von Nummern) kontrollieren können, d.h. es soll möglich sein, alle bereitgestellten Funktionen des Player zu bedienen. Das Hauptmenü soll folgendermaßen aufgebaut sein:

Hauptmenü

1. Lied hinzufügen
2. Playlist hinzufügen
-
3. Lied abspielen
4. Playlist abspielen
-
5. Play
6. Skip
7. Pause
8. Stop
-
0. Programm beenden