



Übungsblatt 2

Programmierung und Softwareentwicklung (WS 17/18)

Abgabe: Freitag, 17.11.2017, 23:55 Uhr — Besprechung: ab Montag, 20.11.2017

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 5 Aufgaben mit einer Gesamtzahl von 40 Punkten.

Aufgabe 1 Konzepte der objektorientierten Programmierung [Punkte: 8]

- (a) (4 Punkte) Erklären Sie, was man in der objektorientierten Programmierung unter einer *Klasse*, einem *Attribut*, einer *Methode* und einem *Objekt* versteht.
- (b) (1 Punkt) In Aufgabenblatt 0 haben Sie lokale Variablen kennengelernt. Beschreiben Sie, worin der Unterschied zwischen einer Variablen und einem Attribut besteht.
- (c) (3 Punkte) Was versteht man in der Programmierung unter einem Paket? Wie sollte man den Namen eines Pakets konstruieren? Wozu braucht man Pakete?

Aufgabe 2 Einführung in die objektorientierte Programmierung I: Klassen [Punkte: 14]

Einer Prüfung sind immer ein Kurs, eine Raumnummer, ein Prüfungsdatum sowie ein Prüfer, der die Prüfung konzipiert hat, zugeordnet. Der Prüfer muss Leiter des Lehrstuhls sein, der den Kurs anbietet. Der Kurs wird über eine Kursnummer, einen Kursnamen, die Anzahl der Semesterwochenstunden (SWS) und ein Start- sowie Enddatum identifiziert.

Als Raum kommen für eine Prüfung nur Räume in Frage, die über mindestens 50 Plätze verfügen und explizit für Prüfungen zugelassen sind.

Studenten können sich für eine Prüfung anmelden, diese ablegen oder davon zurücktreten.

Während der Prüfung erfasst ein Mitarbeiter die Namen und Matrikelnummern aller Studenten. Der Mitarbeiter muss ein Angestellter des verantwortlichen Prüfers sein.

Mitarbeiter korrigieren die Prüfung und geben den Studenten eine Note.

- (a) (7 Punkte) Identifizieren Sie in dem obigen Abschnitt alle Klassen sowie deren Attribute und Methoden. Verwenden Sie folgende Notation:

$\text{Klasse}_1 : \text{Attribut}_a, \text{Attribut}_b, \dots : \text{Methode}_\alpha, \text{Methode}_\beta, \dots;$

Hinweis: Für Klassenkandidaten, die keine Attribute und keine Methoden besitzen und die sich mit einem einzigen primitiven Datentyp oder einem String beschreiben lassen, müssen sie keine Klasse angeben. Ein Beispiel hierfür wäre das Prüfungsdatum.

- (b) (4 Punkte) Beschreiben Sie, wie der Quelltext einer Java-Klasse aufgebaut ist. Gehen Sie im Speziellen darauf ein, wie der Klassenname festgelegt wird, das Paket angegeben werden kann, andere Klassen eingebunden, Attribute deklariert und Methoden implementiert werden können. Beschreiben Sie ebenfalls, welchen Einfluss der Klassen- und der Paketname auf die Dateistruktur Ihres Java-Projekts hat.

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

- (c) (3 Punkte) Geben Sie für die Klasse *Student* gemäß Ihres Modells aus Teilaufgabe a) eine Java-Implementierung an. Die Klasse soll sich in dem Paket *de.unistuttgart.iaas.pse.klausur* befinden. Wählen Sie für die Attribute sinnvolle Datentypen aus; Sie dürfen natürlich auch andere in Teilaufgabe a) modellierte Klassen als Datentyp für die Attribute Ihrer Klasse wählen, ohne dass Sie diese implementieren müssen. Implementieren Sie die Methoden, indem Sie auf der Konsole ausgeben lassen, was die Methode machen soll, z. B.

```
void essen() {
    System.out.println("Ich esse.");
}
```

Aufgabe 3 Einführung in die objektorientierte Programmierung II: Objekte [Punkte: 5]

- (a) (1 Punkt) Geben Sie für die Klasse *Student* gemäß Ihres Modells aus Aufgabe 2 a) eine konkrete Instanz an. Weisen Sie den Attributen realistische Werte zu. Verwenden Sie folgende Notation: Instanzname : Klassenname; Attribut_a = Attributwert_a, Attribut_b = Attributwert_b, ...;
- (b) (2 Punkte) Warum werden bei der Beschreibung von Objekten zwar Attribute, aber keine Methoden angegeben?
- (c) (2 Punkte) Ergänzen Sie das folgende Code-Fragment so, dass der Variable *s* eine Instanz der Klasse *Student* aus Aufgabe 2 c) zugewiesen wird. Weisen Sie dieser Instanz eine beliebige Matrikelnummer zu und führen Sie eine in *Student* definierte Methode aus.

```
Student s = ...
```

Beschreiben Sie Ihr Vorgehen.

Aufgabe 4 IceTea-MediaPlayer: Einbindung von Bibliotheken [Punkte: 8]

Laden Sie sich das Archiv *IceTeaMediaPlayer* herunter. In diesem befinden sich mehrere *.jar* Dateien des *BasicPlayers* sowie die Klasse *PlayerCore*. Erstellen Sie ein Projekt mit dem Namen *IceTeaMediaPlayer*. Binden Sie die *JARs* in das Projekt korrekt ein und dokumentieren Sie jeden Schritt mit Hilfe eines Screenshots. Geben Sie die Screenshots und eine Beschreibung, wie Sie vorgegangen sind, ab. Es ist nur nötig diese Dokumentation für eine *JAR* zu führen. Binden Sie dennoch alle *JARs* ein, da Sie diese in Aufgabe 5 benötigen. Nach dem Import der Bibliotheken ist es Ihnen möglich, die Funktionen des *BasicPlayer* in Ihrem Projekt zu nutzen.

Hinweis: Fügen Sie die Screenshots in die abzugebene Beschreibung ein. Geben Sie nur eine einzige PDF ab.

Aufgabe 5 Impl IceTea-MediaPlayer: Verwendung von externen Paketen [Punkte: 5]

Verwenden Sie Ihr *IceTeaMediaPlayer*-Projekt aus Aufgabe 4 und kopieren Sie die Klasse *PlayerCore* aus dem *IceTeaMediaPlayer*-Archiv in das Paket *de.unistuttgart.iaas.icetea*. Vervollständigen Sie die Klasse *PlayerCore*. Diese Klasse soll entwickelt werden, um die Funktionen des *BasicPlayers* zu steuern. Implementieren Sie dazu die folgenden Methoden und erstellen Sie eine *main*-Methode zum Testen dieser Methoden.

Hinweis: Geben Sie lediglich die von Ihnen implementierte Klasse *PlayerCore* ab.

- (a) (1 Punkt) Die Methode *open* soll die als URL angegebene Datei mit dem *BasicPlayer* öffnen. Führen Sie zu diesem Zweck auf dem Attribut *player* die *open*-Methode des *BasicPlayers* aus.
- (b) (1 Punkt) Die Methode *play* soll die Wiedergabe eines Lieds starten. Führen Sie zu diesem Zweck auf dem Attribut *player* die *play*-Methode des *BasicPlayers* aus.
- (c) (1 Punkt) Die Methode *pause* soll die Wiedergabe pausieren. Führen Sie zu diesem Zweck auf dem Attribut *player* die *pause*-Methode des *BasicPlayers* aus.
- (d) (1 Punkt) Die Methode *resume* soll die pausierte Wiedergabe fortsetzen. Führen Sie zu diesem Zweck auf dem Attribut *player* die *resume*-Methode des *BasicPlayers* aus.
- (e) (1 Punkt) Die Methode *stop* soll die Wiedergabe stoppen. Führen Sie zu diesem Zweck auf dem Attribut *player* die *stop*-Methode des *BasicPlayers* aus.