

für den Kurs

## Objektorientierte Softwareentwicklung in Java WS 17/18

---

### ANBIETER & BETREUER

Prof. Dr. Heimo H. Adelsberger  
[heimo.adelsberger@icb.uni-due.de](mailto:heimo.adelsberger@icb.uni-due.de)

Dr. Patrick Veith, Dipl.-Wirt.-Inf.  
[patrick.veith@icb.uni-due.de](mailto:patrick.veith@icb.uni-due.de)

Stefan Friese, Dipl.-Wirt.-Inf.  
[stefan.friese@icb.uni-due.de](mailto:stefan.friese@icb.uni-due.de)

Universität Duisburg-Essen  
Campus Essen  
Fakultät für Wirtschaftswissenschaften  
Informationssysteme technologiegestützter Lernprozesse  
<https://www.itl.wiwi.uni-due.de/>

Abgabe der Studienleistungen an:  
[java.pruefungsaufgabe@vawi.de](mailto:java.pruefungsaufgabe@vawi.de)

### Kurze Skizze des Kursablaufs:

Im Rahmen des Kurses erlernen Sie die Grundlagen der Objektorientierung und deren Umsetzung in der Programmiersprache Java. Das Erlernen der Inhalte wird durch drei Studienleistungen unterstützt. In der ersten Aufgabe werden Sie in einer schriftlichen Einzelarbeit eine Modellierung eines Problems als Klassendiagramm vornehmen. Die zweite Aufgabe ist eine Gruppenarbeit, die ein gemeinsames Klassendiagramm hervorbringt. Als dritte Aufgabe implementieren Sie in der Gruppe auf Grundlage Ihrer Modellierung ein Programm, das die Anforderungen erfüllt. Danach findet eine Abschlussprüfung (abhängig von der Teilnehmerzahl des Kurses entweder mündlich oder schriftlich) statt, in der Sie ihr Verständnis der Inhalte des Kurses unter Beweis stellen und Ihre Abgaben der drei Studienleistungen erklären müssen.

Sollten Sie sich entscheiden, nicht an den Studienleistungen teilzunehmen, müssen Sie in der Prüfung ein vorgegebenes Fremdprojekt erklären, das ihnen gegen Ende des Kurses zur Verfügung gestellt wird. Da praktisches Programmieren für das Erlernen der Kursinhalte unabdingbar ist, raten wir jedoch dringend von dieser Variante ab.

Wir wünschen Ihnen viel Spaß beim Lernen und viel Erfolg bei den Prüfungen!

### Überblick

In den drei Studienleistungen werden Sie ein Programmierprojekt von der Designphase bis zur programmiertechnischen Umsetzung durchführen. Als Beispiel werden wir gemeinsam ein Planungstool erstellen, das im Folgenden beschrieben wird. Beachten Sie, dass Sie für die erste Studienleistung noch nicht alle Informationen benötigen werden.

### Szenario

Die VAWi-Filmpalast GmbH ist ein mittelständischer Kinobetreiber aus Essen. Das Unternehmen verfügt über ein traditionsreiches Kino mit mehreren, verschieden großen Sälen. In den letzten Jahren ist das Unternehmen jedoch unter Druck geraten. Die Konkurrenz durch größere Kinoketten nimmt zu, und Investitionen in 3D-Technologie müssen refinanziert werden. Hinzu kommt ein zunehmend unübersichtlicher Filmmarkt. Von großen Blockbustern bis zu kleinen Independent-Filmen steht den Kinos eine vielfältige Auswahl gegenüber. Den Mitarbeitern des VAWi-

Filmpalastes fällt es zunehmend schwer, hier die richtige Auswahl und damit den Geschmack des Publikums zu treffen.

Der Programmleiter, Herr James Hollywood, erklärt das Problem wie folgt: „Wir planen immer für drei Wochen im Voraus unser Programm. Dafür gehe ich die Angebote an Filmen durch und stelle die Filmauswahl zusammen. Das ist gar nicht so einfach wie es klingt. Unser Publikum erwartet eine gewisse Auswahl, ich darf also nicht immer die gleichen Filme zeigen und muss die verschiedenen Genres abdecken. Dann gibt es natürlich unterschiedlich attraktive Filme - die sich aber auch in den Verleihpreisen unterscheiden. Wenn ich nur die teuersten Streifen einplane, steigt mir der Geschäftsführer aufs Dach. Ach ja, und ich muss natürlich auch entscheiden, welcher Film in welchem Saal läuft und wann. Es gibt immer vier Termine für Vorführungen pro Tag, 15 Uhr, 17:30 Uhr, 20 Uhr und 23 Uhr; um 1:30 Uhr schließen wir das Haus. Bei den Terminen berücksichtigen wir die FSK16-Filme erst ab 20 und FSK18-Filme erst um 23 Uhr. 3D-Filme können wir nur in Sälen mit 3D-Projektoren zeigen. Wenn ich für das alles eine Software von Ihnen bekommen könnte, das wäre schon eine tolle Sache!“

Die Geschäftsführerin, Frau Karla Knausrig, ergänzt: „Eine vernünftige Planung ist aus wirtschaftlichen Gesichtspunkten entscheidend für uns. Dabei geht es um mehr als nur die Entscheidung welcher Film wann laufen soll. Auch die Finanzen müssen geplant werden: Wir müssen Eintrittspreise festlegen, die möglichst hohe Einnahmen bedeuten, und vor den Filmen müssen Werbespots gezeigt werden. Auch bei diesen müssen wir entscheiden, welche wir zeigen wollen, so dass wir viel Geld einnehmen, aber die Zeit zum Abspielen der Werbung genügt und wir unsere Zuschauer nicht mit zu langer Werbung in den Wahnsinn treiben. Die Aufgabe Ihrer Software ist es, einen Plan zu machen, der wirtschaftlich attraktiv ist. Es sollte also auch ausgegeben werden, wie wir bei einem berechneten Plan finanziell da stehen.“

Nach diesen ersten Eindrücken halten Sie noch einmal Rücksprache, und erfahren, was die Software genau leisten soll:

Als Eingabedateien stehen zur Verfügung:

- Eine Datei mit den Sälen des Kinos, die etwa deren Sitzanzahlen (Parkett/Loge) und die 3D-Fähigkeit angibt.
- Eine Datei mit den aktuell verfügbaren Filmen. Diese enthält den Titel, Angaben zu Genres und Regisseuren, die FSK-Klassifizierung, das Erscheinungsjahr, die Spieldauer, die Sprache, und Angaben zur Beliebtheit der Filme.
- Eine Liste mit Werbespots, die gezeigt werden können, mit deren Laufzeit und Angaben zur Vergütung.

Beispieldateien werden Ihnen im Kursverlauf zur Verfügung gestellt (zu Studienleitung 2, jedoch noch nicht für SL1, da Sie diese hier noch nicht benötigen und es nur zu Verwirrung kommen würde).

Ihre Anwendung soll eine Planung für die folgenden drei Wochen erstellen (jeweils als Textdatei):

- Ein Kinoprogramm (eine Datei, gegliedert nach Film, dort pro Tag um welche Uhrzeit in welchem Saal zu welchem Preis der Film gezeigt wird)
- Ein Raumplan (eine Datei, gegliedert nach Sälen, die angibt an welchem Tag um welche Uhrzeit dort welcher Film läuft)
- Einen Finanzplan (eine Datei, die gegliedert ist wie das Kinoprogramm, aber zusätzlich die erwarteten Einnahmen der jeweiligen Vorführung aus Kartenverkäufen, die vor der jeweiligen Vorführung zu zeigenden Werbespots mit ihren Einnahmen sowie die Ausgaben für Filmmieten darstellt)

Nachdem Sie diese Informationen zusammengetragen haben, fällt Ihnen auf, dass für eine Lösung des Problems noch weitere Informationen fehlen. Zum Glück stehen Ihnen die Ansprechpartner des VAWi Filmpalastes jedoch für Rückfragen zur Verfügung.

## Hinweis

*Dieses Szenario ist bewusst knapp formuliert und soll Sie anregen, bei Unklarheiten im Forum nachzufragen. Zudem erhalten Sie so die Möglichkeit Ihre eigenen Vorstellungen mit in die Umsetzung einzubringen.*

## Anforderungen an Ihre Lösung

Die Mindestanforderung an die Software, die Sie im Rahmen dieses Kurses entwickeln, ist die Abdeckung der **Grundfunktionen, also das Generieren eines gültigen Plans und Erzeugen der entsprechenden Dateien**. Zur Erreichung der maximalen Punktzahl in der *dritten* Studienleistung erwarten wir aber etwas mehr, wobei Sie selbständig entscheiden können, in welchen Aspekten Sie mit Ihrer Software glänzen wollen. Dies soll Ihnen die Möglichkeit geben, sich in den Bereichen besonders zu engagieren, die Sie interessant finden. Dies können etwa Dinge sein wie:

- ein besonders leistungsfähiger Optimierungsalgorithmus,
- eine Menüsteuerung,
- Konfigurationsdateien,
- eine grafische Oberfläche,
- eine zusätzliche HTML-Ausgabe,
- besonders interessante Komfort-Features,
- sonstige Zusatzfunktionalitäten.

Um Missverständnissen vorzubeugen: Hiermit ist nicht gemeint, dass Sie *alle* diese Dinge umsetzen sollen oder müssen. Sie sollen sich gemäß ihrer Interessen gezielt vertiefen.

## Hinweis

*Bei Unklarheiten zu den fachlichen Anforderungen sind Sie eingeladen, im Forum nachzufragen. Dazu dient in der Lernumgebung das Szenario-Forum. In diesem Forum werden die Kursbetreuer in die Rolle der Verantwortlichen Mitarbeiter treten und entsprechende Rückfragen beantworten.*

## Erste Studienleistung (4 Punkte)

---

Modellieren Sie das Planungsprogramm in UML und erläutern Sie Ihr Modell.

Beschreiben Sie, in Einzelarbeit, mittels **UML-Klassendiagrammen** ein Klassensystem, das eine mögliche Struktur des Szenarios wiedergibt. Es ist in dieser Studienleistung noch **keine Darstellung der Attribute und Methoden erforderlich**, jedoch sollte jede Klasse kurz erläutert und das Gesamtsystem beschrieben werden (**Was sind die Aufgaben der Klassen, wo ist der Einstiegspunkt in das Klassensystem etc.**).

Sie haben die Möglichkeit die inhaltlichen Aspekte der Aufgabe im Forum zu diskutieren. Betrachten Sie Beiträge der Tutoren zu Diskussionen im Szenario-Forum als Gespräch mit den Fachleuten aus der Reederei. Natürlich steht es Ihnen offen, Fragen zur Modellierung im Allgemeinen zu stellen, nur tatsächliche Modellierungsvorschläge dieses Szenarios sollen *nicht* ins Forum gestellt werden.

Falls sie unsicher sind, wie sie vorgehen sollen, empfehlen wir Ihnen das **KFZ-Beispiel im Skript (Kap. 5) anzuschauen**.

## Abgabe der Lösung

Ihre Lösung senden Sie uns an [java.pruefungsaufgabe@vawi.de](mailto:java.pruefungsaufgabe@vawi.de) bis spätestens **Montag, 13. November 2017 um 23.59 Uhr** per E-Mail. Verwenden Sie als **Betreff** „OSJAVA Lösung SL1 <Ihr Name>“. Ihre Lösung besteht aus:

1. Entweder einem Dokument im PDF-Format mit ihren Erläuterungen oder aber einem ZIP-gepackten BlueJ-Projektverzeichnis. Im Falle von BlueJ befinden sich Ihre Erläuterungen in der Readme-Datei des Projekts.
2. Zusätzlich einer Grafikdatei mit ihrem Modell im PNG-, JPG- oder GIF-Format (Diese Datei ist unabhängig von Punkt 1 immer erforderlich!)

**Bitte beschränken Sie sich bei Ihren Erläuterungen auf maximal 4 Seiten** (bei wesentlicher Überschreitung ziehen wir Punkte ab). Sie benötigen kein Deckblatt oder Inhaltsverzeichnis, da dies für uns nur zu mehr Blättern führt, aber bei solch kurzen Dokumenten keinen wesentlichen inhaltlichen Mehrwert darstellt.

Eine Auswahl Ihrer Ausarbeitungen wird anschließend von uns in Moodle gestellt. Auf Basis Ihrer Ausarbeitungen wird dann - in der 2. Studienleistung - je ein Klassensystem von jeder Gruppe erarbeitet.

### **Hinweise:**

*Gilt für alle drei Studienleistungen: Bitte benutzen Sie weder in Klassennamen noch in sonstigen Dateinamen Umlaute (ä,ö,ü) oder Unterstriche („\_“)!*

Die zum Einlesen der Eingabedaten aus einer Datei notwendigen Klassen werden Ihnen gestellt (siehe Moodle). **Im Prinzip muss ihr Klassensystem nur die Klassen der Geschäftslogik- und der Datenschicht enthalten**. Sie benötigen keine grafische Oberfläche, allerdings muss ihr Programm in SL3 eigenständig, d. h. ohne die Benutzung von BlueJ, ablaufen können.

Sollte Ihnen die Modellierung des komplexen Szenarios Schwierigkeiten bereiten, so analysieren und strukturieren sie es mit dem im Kurs beschriebenen Object Engineering Process. Streichen Sie zuerst alle überflüssigen Angaben.

## Zweite Studienleistung (4 Punkte)

---

Nachdem Sie nun in Gruppen eingeteilt sind, entwerfen Sie ein gemeinsames Klassensystem, mit dem Sie weiterhin arbeiten möchten. In der dritten Studienleistung werden Sie dieses System als Grundlage für Ihre Programmierung verwenden.

Begründen Sie Ihre Lösung ebenso wie in Studienleistung 1 auf max. 4 Seiten (bei wesentlicher Überschreitung ziehen wir Punkte ab). Wägen Sie hierbei die unterschiedlichen Möglichkeiten gegeneinander ab *und dokumentieren Sie dies*. Das Klassensystem soll kurz und prägnant beschrieben und die Zusammenhänge der Klassen des Klassensystems dokumentiert werden (Als Pi-mal-Daumen Regel: pro Zusammenhang ein bis zwei Sätze). Sie benötigen kein Deckblatt oder Inhaltsverzeichnis, da dies für uns nur zu mehr Blättern führt, aber bei solch kurzen Dokumenten keinen wesentlichen inhaltlichen Mehrwert darstellt.

Wichtig ist, dass jede Klasse auch einer Person zugeordnet wird, die für sie „verantwortlich“ ist bzw. sie später entwickelt/programmiert. Hierfür bitte den Namen der betreffenden Person in den Quellcode einkommentieren (Tag @author). Dieses ist insbesondere für die dritte Studienleistung und die Prüfung relevant. Jedes Gruppenmitglied muss mindestens einer Klasse zugeordnet sein.

Das zu entwickelnde Java-Projekt soll folgendes beinhalten:

- alle Klassen des Systems (inklusive deren Java-Dokumentation)
- alle öffentlichen - public - Attribute (inklusive deren Java-Dokumentation)
- alle öffentlichen Methoden der Klassen (nur Java-Dokumentation und Deklaration des Rückgabewerts und der Eingabeparameter. Keine Anweisungen mit Ausnahme von einer return-Anweisung pro Methode um die Klassen übersetzbar zu machen!)

*Hinweis:* Es kann durchaus vorkommen, dass keine öffentlichen Attribute in Ihrem Projekt vorkommen. Das ist eine Designentscheidung, die Sie in der Gruppe fällen müssen, natürlich dürfen sie über die Vor- und Nachteile von „public“-Attributen im Forum (auch im allgemeinen Forum) diskutieren und sich so ein eigenes Bild der Sachlage machen.

*Gilt für alle drei Studienleistungen: Bitte benutzen Sie weder in Klassennamen noch in sonstigen Dateinamen Umlaute (ä,ö,ü) oder Unterstriche („\_“)!*

## Abgabe der Lösung

Siehe auch Studienleistung 1. Senden Sie uns die Lösung **bis spätestens Montag, 11. Dezember 2017 um 23:59** an [java.pruefungsaufgabe@vawi.de](mailto:java.pruefungsaufgabe@vawi.de)

Ihre Abgabe umfasst:

1. Das beschreibende Dokument im PDF-Format.
2. Das Java-Projekt als ZIP-gepacktes BlueJ-Projekt (bzw. falls sie kein BlueJ verwenden, als ZIP-gepackten Ordner).
3. Das Klassendiagramm zusätzlich als Grafik im PNG-, JPG- oder GIF-Format

Senden Sie uns die Lösung an [java.pruefungsaufgabe@vawi.de](mailto:java.pruefungsaufgabe@vawi.de) zu. Verwenden Sie als **Betreff** „OSJAVA Studienleistung 2 Gruppe Nr. <Gruppennummer>“.

## Dritte Studienleistung (10 Punkte)

---

Erstellen Sie in Ihrer Gruppe auf Grundlage des Klassensystems ein lauffähiges JAVA-Programm zum gegebenen Szenario.

Das Hauptziel ist es, ein Programm zu erstellen, welches die Grundanforderungen aus dem Szenario löst und lauffähig ist. Hierfür werden bis zu sieben Punkte vergeben. Die Dateinamen für die Eingabedateien nehmen Sie als Aufrufparameter entgegen und erzeugen entsprechende Ausgabedateien. Das Programm muss in einer JAVA-Umgebung 1.7 lauffähig sein (unabhängig vom Betriebssystem - beachten Sie dies bei Pfadangaben, die keine hartcodierten Backslashes beinhalten dürfen). Die bereitgestellten Klassen zum Einlesen der Dateien dürfen sie selbstverständlich in Ihrem Programm verwenden. Für die weiteren drei Punkte der dritten Studienleistung wird erwartet, dass sie zusätzliche Features realisieren. Beachten Sie dazu den Abschnitt „Anforderungen an Ihre Lösung“ am Beginn dieses Dokuments. Um Missverständnissen vorzubeugen: Es wird nicht erwartet, dass Sie alle genannten Zusatzfunktionalitäten implementieren - dies sind lediglich Vorschläge, aus denen Sie sich die für Sie interessanten Funktionen aussuchen können.

Schreiben Sie als Teil des Java-Projekts eine Datei README.txt, in der der Aufruf des Programms dokumentiert ist. Dokumentieren Sie im JAVA-Code mindestens alle Klassen und Methoden mit der Sichtbarkeit „public“ und „protected“. Verwenden Sie hierfür den üblichen Stil des JavaDoc. Inhaltlich müssen aus der Methoden- und Klassendokumentation der Zweck, die Aufruf- und Rückgabeparameter beschrieben sein. Fügen Sie in komplexere Methoden einzelne Kommentare in den Quellcode ein, so dass Ihre Lösung nachvollzogen werden kann. Die Dokumentation ist Teil der Bewertung!

Geben Sie in jeder Klasse an, wer diese erstellt bzw. programmiert hat (Tag @author). In der Datei README.txt geben Sie bitte an, wer welche Bereiche des Programms bearbeitet hat. Diese Zuordnung muss aus formalen Gründen vorhanden sein.

**Hinweis:** Gilt für alle drei Studienleistungen: Bitte benutzen Sie weder in Klassennamen noch in sonstigen Dateinamen Umlaute (ä,ö,ü) oder Unterstriche („\_“)!

### Abgabe:

Senden Sie uns die Lösung **bis spätestens Montag, 29. Januar 2018 um 23:59** an [java.pruefungsaufgabe@vawi.de](mailto:java.pruefungsaufgabe@vawi.de) zu. Verwenden Sie als **Betreff** „OSJAVA Studienleistung 3 Gruppe Nr. <Gruppennummer>“.

Ihre Abgabe umfasst:

1. Ein Paket (ZIP, JAR oder TAR) mit Ihrem Java-Projekt. Bitte bemühen Sie sich, hier keine überflüssigen Dateien mit zu schicken (in der Vergangenheit hatten wir oft Abgaben, in denen ganze svn-Repositories, Backup-Versionen bzw. veraltete Versionsstände, gruppeninterne Planungsdokumente etc. dabei waren die nicht zur Abgabe gehörten und sie für uns sehr unübersichtlich gemacht haben)
2. Ein Dokument mit einer Beschreibung wie das Programm zu bedienen ist und welche Funktionen es bietet (dies kann sehr kurz gehalten werden und mit der Readme-Datei identisch sein) im Format PDF. **Gehen Sie aber unbedingt explizit darauf ein, an welchen Stellen genau Ihr Programm über die Minimalanforderungen hinaus geht.** Es wäre schade, wenn ihnen Punkte entgehen, weil wir eine Besonderheit ihres Programms nicht entdecken.
3. Das Klassendiagramm zusätzlich als Grafik im PNG-, JPG- oder GIF-Format. Das Klassendiagramm kann sich natürlich vom Diagramm aus SL2 unterscheiden, wenn sich im Verlauf der Implementierung Änderungen ergeben haben. Achten Sie hierbei unbedingt auf die Druckbarkeit: Die Datei muss im Hochformat auf DIN-A5-Größe ausgedruckt lesbar sein. Teilen Sie das Diagramm notfalls auf mehrere Dateien auf. Diese Anforderung ist sehr wichtig, da sie ansonsten ggf. in der Prüfung ein nicht lesbares Diagramm vorgelegt bekommen. Dies liegt in Ihrer Verantwortung!

*Hinweis zur Abgabefrist:* Die Abgabefrist wurde absichtlich etwas vor den ersten Prüfungsblock gelegt. Wir werden Ihre Lösungen durchsehen und möchten Ihnen gerne vor dem Prüfungsblock noch Feedback geben. Dafür brauchen wir allerdings ein wenig Zeit und auch Sie sollen noch ein paar Tage haben, um sich mit unseren Kommentaren auf die Prüfung vorzubereiten.

Wir wünschen Ihnen viel Erfolg im OSJAVA-Kurs!

Die Kursbetreuer