Universität Hamburg – Fachbereich Informatik
64-146a – Programmiertechnisches Praktikum – PTP 2022
Dozent: Dipl. Inf. Andreas Heymann
Verfasser: Azevedo, João – Matr.Nr.: 6619618
Lüdemann, Jonah – Matr.Nr.: 7311749
ptp22-mi09
Sommersemester 2022

Projektdokumentation: „Mediathek“

Projektthema:

Anwendung zur Verwaltung einer Mediensammlung (Videospiele).

Kurzbeschreibung:

Die Anwendung „Mediathek“ ermöglicht Nutzenden, einen Überblick über ihre Videospielsammlung zu behalten. Mit der Mediathek können Nutzende über die grafische Oberfläche Einträge erstellen, löschen, bearbeiten, suchen und sortieren. Außerdem bietet die Oberfläche eine Übersicht über die gespeicherten Einträge in tabellarischer Form und die Möglichkeit, eine Liste mit den gespeicherten Einträgen in Textformat zu exportieren.
Ein Eintrag besteht aus den Attributen Titel, Veröffentlichungsjahr, Plattform und eine Anmerkung darüber, ob dieser, also ein Videospiel, durchgespielt wurde.
Die Mediathek speichert ihre Einträge lokal in einer relationalen Datenbank, hierfür wird die freie SQL-Datenbank Hyper Structured Query Language Database, kurz HSQLDB, verwendet. Für die Benutzeroberfläche wurde ausschließlich das Toolkit Java-Swing verwendet.
Mögliche Erweiterungen sind die Speicherung von Bilddateien mit Covern bzw. Boxarts zu den Einträgen, die Erstellung von Statistiken und die Möglichkeit des Exports und Imports der Datenbank.

Anwendungsszenario:

Zweck der Mediathek ist, die von der bzw. dem Nutzenden eingepflegten Einträgen zu speichern und diese auf der grafischen Oberfläche abzubilden. Nach Start der Anwendung sieht die bzw. der Nutzende die in der Datenbank bereits enthaltenen Einträge. Über das

Menü „Datei“ unter dem Punkt „Neuer Eintrag“ ist es möglich, der Datenbank einen neuen Eintrag hinzuzufügen. Die Informationen zum Eintrag (Titel, Plattform, Veröffentlichungsjahr und der Wert „durchgespielt“) werden in einem separaten Fenster über Tastatureingabe in Textfeldern und Selektion in Drop-Down-Menüs hinzugefügt. Vorhandene Einträge können bearbeitet werden, indem die bzw. der Nutzende diese auswählt und in der grafischen Oberfläche auf den Knopf „Bearb.“ klickt. Mit dem Knopf „Löschen“ können einzelne Einträge von der Datenbank entfernt werden. Im Menü „Datei“ gibt es außerdem die Funktion, die Mediathek zurückzusetzen, was alle in der Datenbank gespeicherten Einträge löscht.
Für die Suche bietet die Mediathek zwei Möglichkeiten an: Bei Eingabe in das Textfeld werden automatisch Treffer angezeigt, die den gesuchten Ausdruck enthalten. Bei Auswahl über das Drop-Down-Menü „Plattform“ werden alle Einträge angezeigt, die die gesuchte Plattform enthalten.

User Stories

Medien-Klasse

Jeder Eintrag ist eine Instanz der Klasse „Medium“ und enthält die Attribute Titel, Plattform, Veröffentlichungsdatum (Jahr) und „Durchgespielt“ (true oder false).

Einrichtung der Datenbank

Die Datenbank, in der die Medieneinträge gespeichert werden. Für die Einrichtung muss eine Verbindung hergestellt und eine Tabelle erstellt werden, in der die Einträge gespeichert werden.

Hauptfunktionalität der Datenbank

Die Methoden, die die SQL-Ausdrücke an der Datenbank ausführen. Tabelle erstellen, Eintrag hinzufügen und löschen.

GUI – Hauptfenster

Grafische Oberfläche für die Darstellung und Steuerung der Datenbank. Dazu gehören Bedien- bzw. Eingabeelemente wie Menüs mit Unterpunkten, Knöpfe, Drop-Down-Menüs und ein Eingabefeld.

Einbindung der Datenbank in die GUI

Die gespeicherten Einträge bzw. die Funktionalität sind in der GUI so eingebunden, dass diese für die bzw. den Nutzenden über die grafische Oberfläche sicht- und steuerbar gemacht werden.

Dialogfenster: Eintrag hinzufügen

Beim Auswählen der Aktion „Neuer Eintrag“ wird ein Fenster geöffnet, über das die Eingabe von Informationen für einen neuen Eintrag möglich sind.

Datenbank: Weitere Funktionen

Einträge können zusätzlich bearbeitet und/oder gelöscht werden. Mehrere bzw. alle Einträge können gleichzeitig gelöscht werden (Zurücksetzen).

Dialogfenster: Eintrag bearbeiten

Beim Auswählen der Aktion „Bearb.“ öffnet sich ein Fenster, das die aktuellen Informationen zum ausgewählten Eintrag enthält, damit die Attribute bearbeitet werden können.

GUI: Bestätigungsdialoge

Bei den unterschiedlichen Aktionen auf der Benutzeroberfläche wie Fenster schließen, Eintrag löschen oder bearbeiten öffnet die Anwendung entsprechende Dialoge, die Informationen enthalten oder nach einer Bestätigung der bzw. des Nutzenden fragt.

Zeitplan und Arbeitseinteilung

Die Zuständigkeiten im Projekt wurden zu Beginn der Bearbeitungszeit festgelegt, dabei sind 2 grobe Verantwortungsbereiche entstanden: Die GUI, die Hauptsächlich von João Azevedo geschrieben wurde; und die Datenbank im Verantwortungsbereich von Jonah Lüdemann. Um die Anforderungen des Model-View-Controller-Ansatzes (MVC) zu erfüllen, wurde das Projekt nachträglich um die Klasse MediathekController erweitert, was ein größeres Refactoring voraussetzte. Dieses wurde von João Azevedo durchgeführt.

Die gemeinsame Arbeit bzw. die Besprechungen erfolgten hauptsächlich mittwochs während der Veranstaltung vor Ort, die meiste Arbeit wurde jedoch zu Hause asynchron erledigt.

Prio.	User Story	Aufwand (SOLL)	Aufwand (IST)
1	Medien-Klasse	1 Std.	1 Std.
2	Erste Tests	2 Std.	2 Std.
3	Hauptfunktionalität der Datenbank	4 Std.	3,5 Std.
4	GUI: Hauptfenster	4 Std.	2 Std.
5	Einbindung der Datenbank in die GUI	6 Std.	8 Std.
6	Dialogfenster: Eintrag hinzufügen	4 Std.	8 Std.
7	Datenbank: Weitere Funktionen	4 Std.	5 Std.
8	Dialogfenster: Eintrag bearbeiten	4 Std.	4 Std.
9	Weitere Tests	4 Std.	6 Std.
10	GUI: Bestätigungsdialoge	6 Std.	6 Std.
11	Refactoring	8 Std.	12 Std.

Übergreifende Projektbeschreibung

Ziel des Projekts war die Entwicklung einer Anwendung zur Verwaltung von Datensätzen, die sowohl eine Datenbank wie auch eine Benutzeroberfläche beinhaltet. Das Programm sollte uns Einblicke in die Herausforderungen und Möglichkeiten der Entwicklung dieser Anwendungsart geben, die unseres Erachtens relevante Fähigkeiten sind. Dabei war uns wichtig, einen nicht zu spezifischen Anwendungsfall zu wählen, sodass unsere Anwendung ebenfalls für andere zu verwaltende Dinge umgeschrieben bzw. erweitert werden könnte.

Die Aufteilung der Aufgaben war von vornherein klar und wurde meistens eingehalten. Beim Austausch in den Präsenzterminen lag der Schwerpunkt auf das Zusammenbringen der zwei Hauptteile, also Datenbank und Benutzeroberfläche, sodass hier durch gegenseitige Unterstützung ebenso übergreifend gearbeitet wurde.

Zu Beginn des Projektes lag die Herausforderung darin, die Vorstellung des fertigen Produktes im Einklang mit dem umsetzbaren Ergebnis zusammenzubringen. Das lag zum einen an den Möglichkeiten der Technologie (wie z. B. Java-Swing), zum anderen auch an unseren Fertigkeiten und Erfahrung, den Aufwand und die Machbarkeit der einzelnen Schritte einzuschätzen. Aufgrund dessen wurde manchmal erst nach geleisteten Arbeitsstunden klar, dass gewisse Teile der Anwendung nicht zum erwünschten Ergebnis führen bzw. diese sich nicht mit anderen Teilen verbinden ließen. So entstand die Situation, dass zuerst gedachte komplizierte Schritte weniger Zeit in Anspruch nahmen, während scheinbar leichte Sachen mehr Zeit kosteten als geplant.

Der Test-First-Ansatz klappte anfangs besser, später im Projekt lag der Fokus jedoch darauf, Funktionalitäten zu implementieren und Fehler zu beseitigen. Aus Zeitgründen wurde auf JUnit-Tests für die Swing-Komponente verzichtet, vielmehr wurde die Benutzeroberfläche interaktiv getestet. Zusätzlich sind Fehlermeldungen an die Nutzenden eingebaut, um unzulässige Eingaben zu verhindern oder diese zu korrigieren. Hier wurde versucht, für jedes Eingabefeld eine passende Exception mit Fehlermeldung an die Nutzenden einzubauen, dies erwies sich jedoch als kompliziert. Aus dem Grund wurde z. B. auf die Eingabe der Plattform durch Selektion im Drop-Down-Menü gesetzt, um der bzw. dem Nutzenden wenig Spielraum für fehlerhafte Eingaben zu geben, was die Überprüfung der Eingaben erleichterte.

In der vorliegenden Anwendungsversion sind die Interaktionen auf der Benutzeroberfläche hauptsächlich mit der Maus für Aktionen und der Tastatur für Eingaben umgesetzt. Eine Steuerung ausschließlich mit der Tastatur wurde aus Zeitgründen nicht implementiert. Ebenfalls gewisse Swing-Elemente könnten anders programmiert werden: Die Auswahl der Attribute „Durchgespielt“ könnte als JRadioButton anstelle eines JComboBox, die Eingabe des Veröffentlichungsjahrs über JSpinner implementiert werden. Die Darstellung des Wertes „Durchgespielt“ könnten zudem in einer nutzendenfreundlicheren Art mit Ja und Nein statt true und false dargestellt werden.

Trotz der Herausforderungen in der Kommunikation in der Gruppe und des Lernprozesses bei der praktischen Umsetzung liegt eine Anwendung vor, die ihre wesentlichen Anforderungen erfüllt.

Systemtechnische Beschreibung

Die Anwendung ist nach dem Model-View-Controller-Konzept (MVC) aufgebaut, wie in Hubert Partls Java-Einführung beschrieben. Sie enthält eine Applikation (Main), das Datenmodell (MediathekData), die Ansicht (MediathekGUI), die Steuerung (MediathekController) und nicht zuletzt die Klasse Medien bzw. Testklassen.

Ein Medieneintrag wird in der Klasse Medien definiert, sie enthält folgende Attribute: Name vom Typ String, der für den Mediennamen steht; Plattform ebenfalls vom Typ String steht für die Spielkonsole, für die das Spiel veröffentlicht wurde; Der Integer Veröffentlichung steht für das Veröffentlichungsjahr des Spiels; der Boolean-Wert Durchgespielt enthält die Information, ob das Spiel bereits durchgespielt wurde.

Die Speicherung der Medien, die in die Mediathek eingefügt werden, findet über eine Datenbank statt. Die nur lokal verfügbare Datenbank wird mit Hilfe von HSQLDB implementiert. Wenn eine neue Tabelle erstellt wird, wird zunächst überprüft, ob bereits eine Tabelle mit dem angegebenen Namen existiert. Sollte dies nicht der Fall sein, dann wird eine neue Tabelle mit den Spalten Name, Plattform, Veröffentlichung und Durchgespielt erstellt. Name und Plattform werden als Varchar gespeichert, Veröffentlichung als Int und Durchgespielt als Boolean. Im späteren Verlauf des Projekts ist aufgefallen, dass es sinnvoll gewesen wäre, das Veröffentlichungsdatum auch als Varchar zu speichern, da es einfacher gewesen wäre direkt auf die Zahl als String zugreifen zu können, jedoch war es an dem Punkt nicht mehr die Mühe wert, es zu ändern.
Um den Inhalt der Datenbank in der GUI darzustellen, wird eine JTable genutzt. Das hierfür benötigte Tablemodel wird dafür mit den Einträgen aus der Datenbank gefüllt. Wenn nach Namen gesucht oder nach Konsolen gefiltert wird, werden die Parameter von der GUI über den Kontroller übergeben und beim Aktualisieren des Tablemodels werden nur die Einträge, die den Filtern entsprechen, übergeben.
Soll ein Eintrag in der Datenbank gelöscht oder aktualisiert werden, dann muss nur die Zeile, in der sich der Eintrag in der JTable befindet, übergeben werden, um den Eintrag in der Datenbank zu löschen oder zu aktualisieren.

Die Klasse MediathekGUI, wie der Name suggeriert, ist für den Aufbau der Benutzeroberfläche Zuständig. Sie dient dazu, nicht nur den Inhalt der Datenbank darzustellen, sondern sie fungiert als Schnittstelle zwischen der bzw. dem Nutzenden und der Datenbank. Ihr wird das DefaultTableModel und der Steuerungsklasse übergeben, um die Verbindung zum Datenmodell herzustellen. In der View werden sämtliche Fenster wie das Hauptanwendungsfenster, die Eingabe- und Bearbeitungsfenster für die Einträge sowie Bestätigungs- und Meldungsdialoge erzeugt. Um die Klasse übersichtlicher zu gestalten und eine Wiederverwendbarkeit der Fenster zu gewährleisten, werden die Anweisungen zur Erzeugung dieser Anwendungsfenster jeweils in Methoden ausgelagert. Diese Methoden nehmen für die Funktionalität nötigen Komponente entgegen, z. B. die Steuerungsklasse wird übergeben, damit hier den einzelnen Swing-Komponenten die Listener hinzugefügt werden können. Das Hauptfenster bekommt zusätzlich das DefaultTableModel als Parameter, um so die Datenbank als JTable darzustellen.

Das Bearbeitungsfenster wird aus dem Hinzufügen-Fenster erzeugt. Hier wird bspw. den Knopf „Hinzufügen“ unsichtbar gemacht und durch „Speichern“ ersetzt. Die zu bearbeitenden Informationen werden dann auf diesem Fenster dargestellt.
Die MediathekGUI verfügt außerdem über setter- und getter-Methoden, damit das Lesen und Bearbeiten von nötigen Swing-Komponenten außerhalb der Klasse möglich sind. So können die einzelnen Variablen mit der Sichtbarkeit private deklariert werden, was die Sicherheit der Anwendung erhöht.

In der Steuerungsklasse MediathekController, wie das MVC-Konzept besagt, werden die Listener-Interfaces implementiert: In der Mediathek die ActionListener und KeyListener. Die unterschiedlichen Aktionen, die auf der View durch die bzw. den Nutzenden ausgelöst werden, werden über Befehle an die Methode befehlAusfuehren() übergeben. Diese Methode ruft dann die entsprechenden Methoden auf.
Zusätzlich zu den Listenern, hat MediathekController die Aufgabe, Aktionen und Eingaben auf der View an das Datenmodell weiterzugeben. Bei jeder Aktion auf der View wird also in dieser Klasse die entsprechende Methode aus dem Datenmodell aufgerufen. Auch hier werden Eingaben auf mögliche Fehler geprüft und ggf. den entsprechenden Meldungsdialog aus der View aufgerufen.

Bekannte Bugs

Folgende Bugs sind bekannt:
Bei der Verwendung eines Apostrophs im Titel erfolgt das Hinzufügen eines Eintrags nicht korrekt;
Es ist möglich, den Inhalt der JTable in der View alphabetisch zu sortieren. Bei solch einer Sortierung wirf die Anwendung aber einen Fehler, wenn die bzw. der Nutzende versucht, andere Aktionen an einem Eintrag wie das Bearbeiten durchzuführen (Funktion gestrichen).

Tests

Die JUnit-Tests für die Anwendung konnten aus Zeitgründen nicht vervollständigt werden. JUnit-Tests liegen im Paket mediathek.tests für die Klassen Medien und MediathekData vor. Die Tests für die Datenbank sind nur auf die Grundfunktionen Einfügen, Löschen und Aktualisieren beschränkt.
Weitere Testansätze, die noch implementiert werden könnten, sind beispielsweise das Verhalten der Datenbank bei Eingaben, die die maximale Eingabelänge überschreiten. Auch

könnte die Ausgabe der vorhandenen Plattformen oder die Suche in der Datenbank nach Namen und/oder Plattformen auf Richtigkeit überprüft werden.
Die Benutzeroberfläche wurde hauptsächlich interaktiv getestet, sie wirft bei einer falschen Jahreseingabe im Feld Veröffentlichung eine Exception als Dialogfenster an die bzw. den Nutzenden. Auch beim Versuch, einen Eintrag zu bearbeiten, ohne dass der gewünschte Eintrag ausgewählt wurde, wird eine Fehlermeldung ausgegeben.

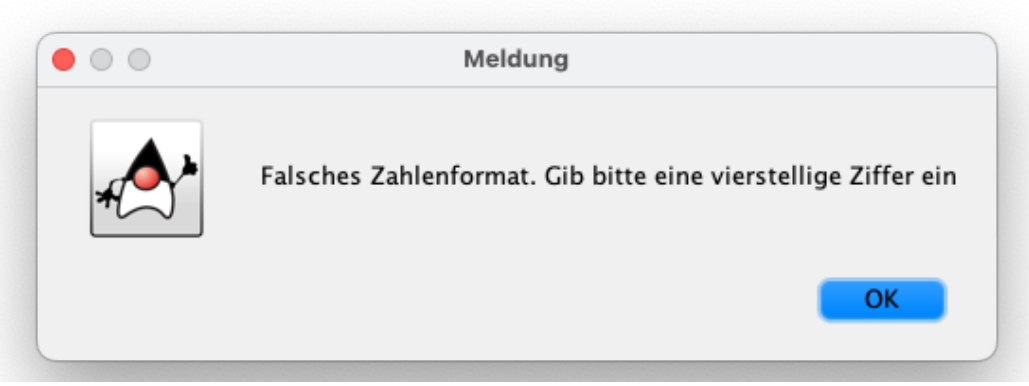

Abb. 1: Fehlermeldung beim Versuch, eine falsche Eingabe zu tätigen.

Wie bereits erwähnt wurden Fehlermeldungen wie in Abbildung 1 nicht für jedes Feld geschrieben. Um dies zu umgehen, wird zum Beispiel den Wert „Neuer Titel“ gesetzt, falls bei der Eingabe das Textfeld Titel leer gelassen wird. Das Attribut Veröffentlichung bekommt den Wert 1970, falls die eingegebene Zahl kleiner als 1970 oder größer als 2022 beträgt. Das Attribut Plattform bleibt leer, falls bei der Erstellung eines Eintrags keinen Wert ausgewählt wird. Bei einer nicht Auswahl erhält der Eintrag den Wert „nein“ für das Attribut Durchgespielt. Hier kann die bzw. der Nutzende einen Eintrag nachträglich bearbeiten.

Anleitung

Die Anwendung ist in der Programmiersprache Java geschrieben und läuft daher auf Windows, Linux oder macOS. Voraussetzung dafür ist die Installation des letzten Java Runtime Environments (JRE), das unter https://www.java.com für die jeweilige Plattform heruntergeladen werden kann.

Die jar-Datei ist im Projektordner unter Anwendung gespeichert. Mit einem Doppelklick auf dem Icon Mediathek.jar öffnet sich das Hauptfenster mit den in der Datenbank bereits gespeicherten Einträgen. Beim ersten Start ist die Mediathek leer und zeigt keine Einträge.

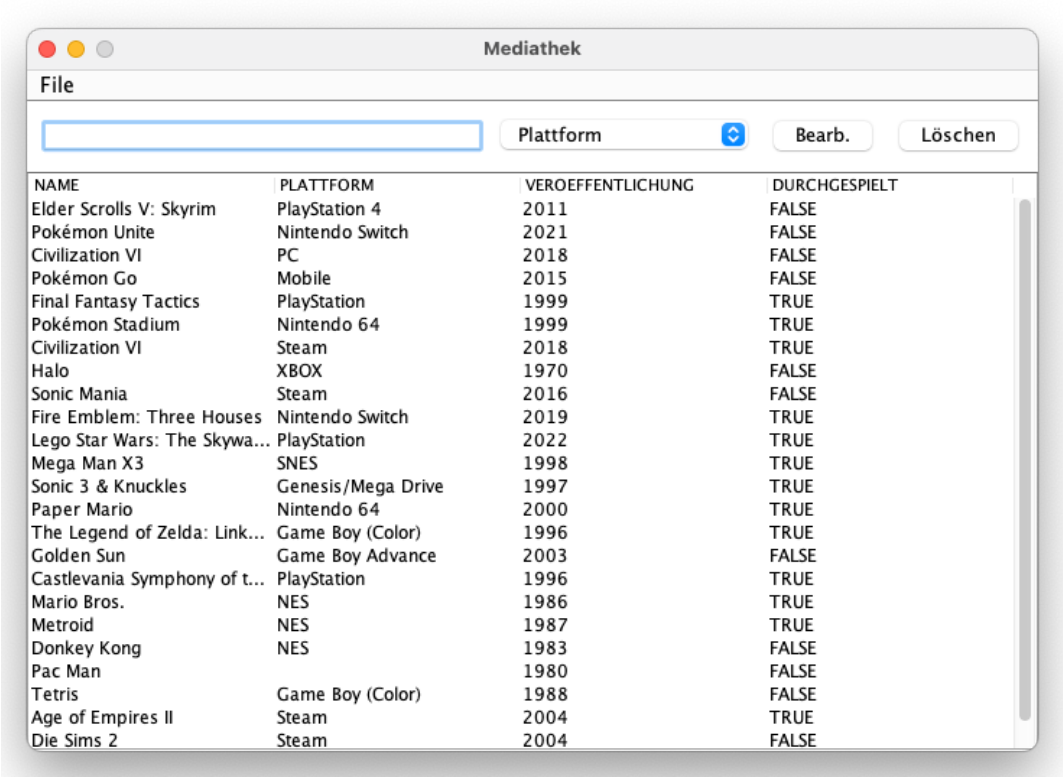

Abb. 2: Hauptfenster der Mediathek

Wichtig: Beim ersten Öffnen erzeugt die Mediathek den Datenbankordner. Solange die jar-Datei und der Datenbankordner sich im selben Pfad befinden, kann die Mediathek auf die gespeicherten Einträge zugreifen.

Eintrag hinzufügen

Unter File auf Neuer Eintrag klicken, ein Dialogfenster öffnet sich. Die Daten zum Spiel eingeben bzw. auswählen und schließlich auf Hinzufügen klicken.

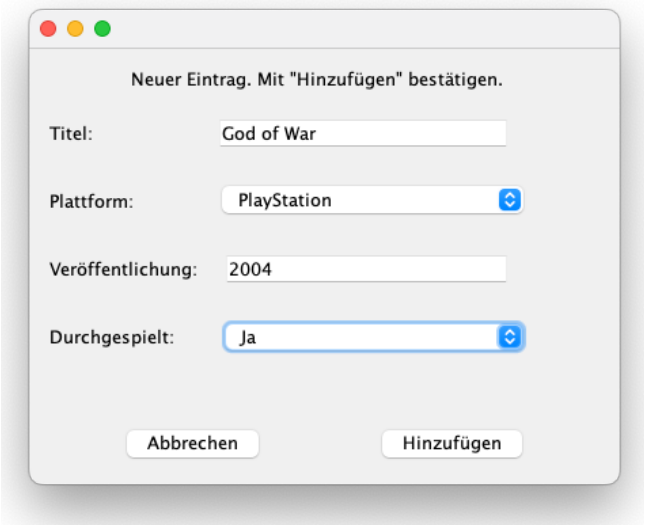

Abb. 3: Eintrag Hinzufügen, Dialogfenster

Eintrag bearbeiten

Für die Bearbeitung den Eintrag im Hauptfenster auswählen und auf Bearb. klicken. Es öffnet sich ein ähnliches Dialogfenster wie in der Abb. 3 dargestellt. Dieses Dialogfenster zeigt die aktuellen Daten zum Eintrag, die dann berichtigt werden können. Schließlich mit Speichern bestätigen.

Eintrag löschen

Um einen Eintrag zu löschen, muss dieser im Hauptfenster ausgewählt sein. Auf Löschen klicken und den Dialog bestätigen. Es können einzelne sowie mehrere Einträge zugleich gelöscht werden.

Einträge suchen

Bei der Eingabe eines Suchwortes in das Textfeld im Hauptfenster zeigt die Mediathek sofort alle Treffer. Auf dieser Weise werden nur Titel gesucht. Zusätzlich kann mittels Drop-Down-Menü die Suche nach Plattformen gefiltert werden. Im Drop-Down-Menu die Auswahl auf Plattform setzen, um das Filter auszuschalten.

Einträge als Textdatei exportieren

Unter File auf Exportieren klicken. Eine Textdatei mit allen Einträgen alphabetisch sortiert wird erzeugt und im Home-Verzeichnis gespeichert.

Mediathek zurücksetzen

Um die Mediathek zurückzusetzen, unter File den Menüpunkt Zurücksetzen wählen und den Dialog bestätigen. Alle Einträge werden aus der Datenbank endgültig gelöscht.

Nächste Schritte

Weitere Implementationsmöglichkeiten für die Mediathek sind: Speicherung von Eintragsdaten in anderen Typen wie z. B. das Veröffentlichungsjahr als String oder eine Darstellung des Werts Durchgespielt in einer nutzendenfreundlicheren Sprache; der Export der Einträge als CSV-Datei, die dann wieder importiert werden kann; eine Papierkorb-Funktion, um versehentlich gelöschte Einträge wiederherzustellen; die Möglichkeit, den Einträgen Coverbilder hinzuzufügen; umfassendere Tests; tastaturbasierte Steuerung; visuelle Gruppierung der Einträge nach Plattform auf der View (z. B. alle Einträge zu einer Plattform werden gebündelt angezeigt); Erstellung von unterschiedlichen Profilen.

Gewonnene Erkenntnisse

Kommunikation ist in einer Gruppenarbeit von hoher Bedeutung und das ist nicht selbstverständlich. Eine bessere Kommunikation und beidseitiges Engagement können einen Unterschied machen, doch dafür müssen sich alle Gruppenmitglieder einsetzen. Der Vorteil einer Gruppenarbeit liegt darin, sich mit seinen Stärken zu ergänzen bzw. bei Schwächen gegenseitig zu unterstützen. In unserem Projekt war dies leider nur bedingt möglich.

Planung ist wichtig, doch diese muss ständig hinterfragt bzw. an das Projekt angepasst werden. Neben dem Hauptziel des Projekts sollte öfter über die kleineren Ziele diskutiert werden, um diese somit an aktuellen Herausforderungen und Umstände anzupassen. Hier ist die Kommunikation ebenfalls ein Stichwort.

Aufgaben, die so aussehen, als ob man sie mal eben schnell schaffen kann, können ganz schön viel Zeit fressen.

Häufig hilft es, wenn eine andere Person mal auf den eigenen Code schaut, um schnell Fehler zu finden, die man selbst nicht entdeckt hat.

Trotz der Herausforderungen im Semester ist es uns gelungen, eine funktionierende Anwendung zu liefern und, auch wenn nicht alles optimal geklappt hat, schließen wir das Praktikum mit einer positiven Bilanz und mit dem Gefühl ab, unsere Kenntnisse in Java und ins Programmieren deutlich vertieft zu haben.