

Datenbanksysteme 2, 8. Übung

O/R-Mapping mit Frameworks

Generelles zu dieser Übung

Diese Übung ist für 2 Wochen gedacht und wird in den bereits von den Übungen 4 und 5 bekannten Gruppen bearbeitet.

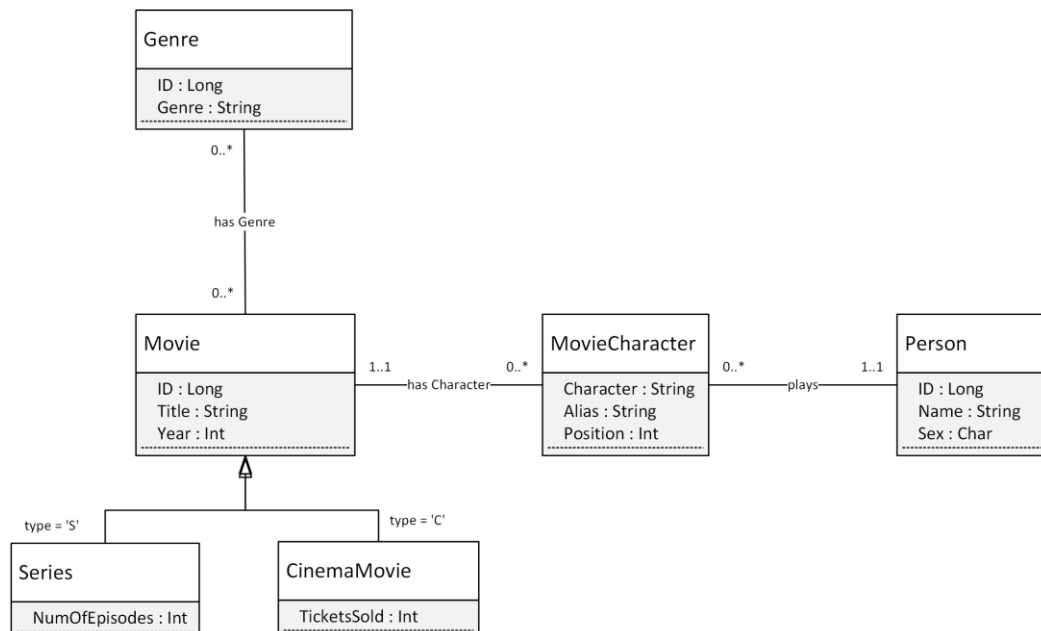
Sie sollten sich auch in dieser Übung die Arbeit in der Gruppe aufteilen!

Das Ergebnis dieser Übung wird als zweite Bonusaufgabe abgegeben. Bitte geben Sie Ihr Gruppenergebnis als ZIP-Datei (alle relevanten Java-Quellcode-Dateien und Dokumente für die analytischen Aufgaben) per Moodle ab. Eine entsprechende Abgabemöglichkeit ist eingerichtet. Die Abgabe ist bis zum 11.12. um 18.00 Uhr möglich.

Um die Bonuspunkte zu erhalten, müssen Sie Ihre Lösung zusätzlich in den Übungen am 12.12. vorstellen. Dazu ist im Moodle-Kurs eine Terminwahl eingerichtet, die Sie bitte wieder erst **NACH ABGABE** der Lösung nutzen. Zur Vorstellung müssen alle Gruppenmitglieder anwesend sein und zu allen Teilen des Programmes auskunftsfähig sein. Dies bedeutet, dass Sie auch Fragen zu Programmteilen beantworten können müssen, die andere Gruppenmitglieder programmiert haben.

Aufgabe 8.1 Aufbau eines Hibernate¹-Projektes

In dieser Aufgabe sollen Sie ein neues Hibernate-Projekt erstellen. Das Klassenmodell ist eine minimale Erweiterung des bekannten Modells aus den Aufgaben 4 und 5:



¹ Wenn Sie möchten, können Sie alternativ auch die Implementierung von OpenJPA als Persistenz-Provider verwenden. Grundsätzlich müssen Sie lediglich die `persistence.xml` entsprechend anders anpassen und eine Abhängigkeit zu OpenJPA statt Hibernate anlegen.

Gehen Sie dabei in folgenden Schritten vor:

- a) Legen Sie ein neues Projekt an (wenn Sie sich mit Maven auskennen, sollten Sie ein Maven-Projekt verwenden, sonst ein „normales“ Java-Projekt).
- b) Binden Sie die Hibernate-Bibliotheken und den JDBC-Treiber des gewünschten Datenbanksystems ein. Für Hibernate ist es am einfachsten, wenn Sie ein Maven-Projekt verwenden (geht auch mit IDEs wie Eclipse, IntelliJ IDEA², Netbeans) und in der `pom.xml` die in der auf Moodle bereitgestellten Musterdatei genannten Dependencies einbinden. Dort finden Sie Abhängigkeiten zum Oracle und zum PostgreSQL-DB-Treiber; Sie benötigen natürlich nur die Abhängigkeit für das gewählte DB-System.
- c) Erstellen Sie eine `persistence.xml`-Datei (Muster dafür auf Moodle verfügbar). Diese muss im Verzeichnis `META-INF` unter Ihrem Quellverzeichnis liegen und dieses ggfs. in der IDE als Ressourcen-Verzeichnis definiert werden. Passen Sie die Datei an (Datenbankverbindung, Username, Passwort).
- d) Erstellen Sie Entity-Klassen zu den Entitäten `Person`, `Movie`, `Series`, `CinemaMovie`, `MovieCharacter`, `Genre`. Annotieren Sie diese Klassen mit passenden JPA-Annotationen. Entscheiden Sie sich dabei zunächst für die aus Ihrer Sicht sinnvollste Umsetzung der Annotationen bzgl. der Vererbung (vgl. Folie 6-52/53) und der Beziehungen (vgl. Folie 6-54-59). **Vergeben Sie in den Annotationen neue Tabellennamen, so dass Sie die bisher verwendeten Tabellen nicht überschreiben.**
- e) Schreiben Sie eine Main-Klasse, die nur die `EntityManagerFactory` anlegt.
- f) Starten Sie das Programm. Beheben Sie evtl. auftretende Fehler.
- g) Vergleichen Sie das von Hibernate angelegte Datenmodell in der Datenbank mit Ihrem bisherigen Datenmodell.
- h) Erstellen Sie eine Testmethode, die Objekte erzeugt und persistiert.
- i) Verändern Sie nun die Annotationen, indem Sie jeweils einmal auch die beiden anderen Arten der Vererbung vorsehen. Zusätzlich können Sie auch die Annotationen bei den Beziehungen variieren (bi-direktional vs. uni-direktional). **Vergeben Sie auch hier jeweils neue Tabellennamen, so dass Sie die bisher verwendeten Tabellen nicht überschreiben.**
- j) Wiederholen Sie die Schritte f) und g) für jede der Alternativen und vergleichen Sie das Datenmodell jeweils auch mit dem in g) ursprünglich erzeugten Modell. Was sind jeweils Vor- und Nachteile der entsprechenden Realisierung der Vererbung in Bezug auf das Datenmodell? Welche Variante würden Sie in der Praxis für am besten geeignet halten?

Aufgabe 8.2 Integration der Entities und Entwicklung weiterer Methoden

In dieser Aufgabe sollen Sie die Vorarbeiten aus Aufgabe 8.1 mit der für Aufgabenblatt 5 bereit gestellten GUI zu einem Projekt bündeln. Die in 8.1 neu hinzugekommenen Klassen `Series` und `CinemaMovie` sowie ihre Attribute müssen Sie für diese Aufgabe nicht weiter berücksichtigen.

² Wenn Sie IntelliJ nutzen und dort ein Jakarta EE Projekt/Module anlegen, ist es wichtig, dass Sie als Typ des Projekts/Moduls „Library“ verwenden.

Gehen Sie in folgenden Schritten vor:

- a) Erstellen Sie eine Kopie ihres Projekts aus Aufgabenblatt 5 (falls Sie das Blatt 5 nicht bearbeitet haben, starten Sie stattdessen mit dem bereit gestellten Gerüst für Blatt 5. Sie müssen in diesem Fall nicht Teil d) machen und in g) die Manager-Klassen erstmals implementieren).
- b) Ergänzen Sie im Projekt die Hibernate-Bibliotheken und den JDBC-Treiber.
- c) Ergänzen Sie eine passende `persistence.xml`-Datei.
- d) Entfernen Sie Ihre `ActiveRecord`-Klassen aus dem Projekt.
- e) Integrieren Sie als Ersatz die `Entity`-Klassen aus Aufgabe 8.1.
- f) Entfernen Sie die `JDBC-Connection` und ersetzen Sie sie durch eine `EntityManagerFactory`.
- g) Implementieren Sie die Methoden in den `Manager`-Klassen neu. Passen Sie ggfs. auch die `Factory`-Klassen aus dem alten Projekt an, falls erforderlich.
- h) Testen Sie das Programm ausführlich³, bspw. durch Einfügen, Ändern und Löschen von Filmen sowie das Eintragen, Ändern und Löschen von Rollen. Wie schon für Blatt 5 ist es sinnvoll, nach dem ersten Starten des Programms und Anlegen des Datenmodells durch Hibernate, einige Genres und Personen aus der „echten“ Movie-DB zu kopieren, um Basisdaten für die folgenden Tests zu haben, da diese Entitäten nicht über die GUI angelegt werden können.

³ Nach jedem Start der GUI dauert es nach dem ersten Click im Hauptfenster recht lange, bis die Anwendung weiterläuft. Sie müssen also etwas Geduld mitbringen. Überlegen Sie, warum dies so ist! Nutzen Sie ggfs. auch die Ausgaben auf der Konsole, um zu sehen, was in dieser Zeit passiert.