

## Erläuterungen zu Aufgabe 8.4

Nach Aufgabenstellung haben wir nur jenen Teil der Analyseklassendiagramme aus 8.3 modelliert, der die Funktionen `MarkSeriesLoanable` und `UnmarkSeriesLoanable` umsetzt. Da `Series` `Seasons` enthalten und diese ihrerseits `Episodes`, haben wir entsprechende Attribute mitgeführt und den allgemeinen Namen `MarkItemLoanable` und `UnmarkItemLoanable` verwendet. (Die Objektklasse `Movie` und sich darauf beziehende Funktionen und Attribute haben wir der Übersichtlichkeit halber weggelassen.)

Bei den Objekten `Series`, `Season` und `Episode` aus dem `movies.ecore`-Model handelt es sich um Interfaces. Diesen Objekten gemeinsam ist das auffällige Fehlen einer Methode `setLoanable()`, die wir in unserem Entwurfsklassendiagramm ergänzt haben. (Bemerkung: `isLoaned()`, `setLoaned()` und `isLoanable()` sind jeweils vorhanden, es fehlte einzig `setLoanable()`.)

Nach erschöpfender Analyse des Modells der UI-Klassen ergab sich eine Gemeinsamkeit: View- oder Dialogklassen besitzen jeweils einen Handler mit einer typischen Funktion. Einen solchen haben wir modelliert, genau so wie die der Aufgabenstellung entsprechende Dialog- bzw. View-Klasse `MarkItemLoanableView`. Für die Ausstattung derselben haben wir uns an ähnlichen Klassen aus dem UI-Klassenzoo orientiert.

Den Übergang ins Modell haben wir entsprechend der Vorlesungsfolie 8 / F.52 modelliert, die wir intensiv studiert haben. Mit den dort vorhandenen Informationen wurden die Interface-Klassen `Project` und `MoviesPackage` ausgestattet. Schließlich färbten wir auch die Klassen nach der vorherrschenden Konvention ein.

Das Layout des vorliegenden Entwurfsklassendiagramms ist in etwa identisch mit dem Layout der Analyseklassendiagramme aus Aufgabe 8.3 damit die Analogien auch graphisch zutage treten. Da für die Klassen in allen Diagrammen auch ähnliche Namen verwendet wurden, haben wir auf eine weitere Beschreibung verzichtet.

Das Sequenzdiagramm müssen wir nachreichen, denn in wenigen Stunden beginnt die Vorlesung.