

Funktionale Programmierung

Blatt 4

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik
Hochschule München

Letzte Änderung: 09.10.2018 06:54

Wir werden auf diesem Blatt weiter an der Verwaltung für eine Videothek von Blatt 3 arbeiten. Arbeiten Sie daher weiter im Projekt von Blatt 3. Implementieren Sie immer auch gleich Tests und schreiben Sie Dokumentation.

Aufgabe — Higher Order Moviestore

- a) Implementieren Sie eine Funktion

```
add :: (Name, FSK) → Moviestore → Moviestore
```

die einen Filmnamen und die dazugehörige FSK-Freigabe in den Bestand einfügt. Für die Id berechnen Sie die höchste bisher vergebene Id und addieren Sie 1. Ein neu eingefügter Film soll natürlich ausleihbar sein. Berechnen Sie die höchste bisher vergebene Id mit Hilfe von `map` und `maximum`.

- b) Implementieren sie eine Funktion

```
availableForAgeOf :: Age → Moviestore → [Movie]
```

die die Liste der Filme berechnet, die für das übergebene Alter ausleihbar sind. Verwenden Sie `filter`.

- c) Implementieren Sie eine Funktion

```
showNumberedMovieList :: [Movie] → String
```

die eine Liste von Filmen so in einen `String` umwandelt, dass vor jedem Film (Umwandlung durch `showMovie`) eine laufende Nummer steht und dahinter ein Zeilenumbruch. Hinter dem letzten Film darf kein Zeilenumbruch mehr stehen. Verwenden Sie `zipWith` und `intercalate` aus dem `Data.List`-Modul.

d) Implementieren Sie schließlich noch

```
showAvailableForAge :: Age -> Moviestore -> String
```

mit der Sie die nummerierte Liste (Teilaufgabe c) der Filme für das übergebene Alter (Teilaufgabe b) als `String` berechnen können.