Übung W4: Queries & Criteria API

Ausgangslage & Ziele
In dieser Übung schliessen wir die Persistenz-Schicht des Movie-Rental Projektes ab. Ziel ist, dass Sie nach der Bearbeitung dieser Übung alle eine lauffähige Version haben in welcher die Daten mit Hilfe von JPA persistiert werden. Sie werden in darauffolgenden Übungen auf dieser Version weiterarbeiten.

Aufgabe 1) Named Queries

Die bis jetzt definierten Queries können durch *named queries* ersetzt werden. Solche Queries werden auf den Entitäten definiert. Ein *named query* für die Abfrage aller Preiskategorien würde wie folgt aussehen:

```
@NamedQuery(name="PriceCategory.all", query="from PriceCategory")
```

Nach diesen Änderungen sollten weiterhin alle Tests ohne Probleme durchlaufen, und auch die `MovieRentalClient`-Applikation sollte einwandfrei funktionieren.

Aufgabe 2) Criteria API [optional]

Sie können anstelle der Definition von (named) Queries auch das Criteria-API verwenden um ihre Queries zu formulieren. Am besten definieren Sie dazu eine abstrakte Repository-Klasse, der Sie den Typ der Entitäten als Parameter vom Typ `Class<T>` übergeben:

```
public abstract class AbstractJpaRepository<T> implements Repository<T, Long> {

    @PersistenceContext
    protected EntityManager em;

    final private Class<T> type;

    protected AbstractJpaRepository(Class<T> type) { this.type = type; }

    …
```

In dieser Klasse können dann die Methoden

- `findById`
- `findAll`
- `save`
- `delete`
- `deleteById`
- `existsById`
- `count`

implementiert werden, wobei Sie das Criteria-API nur für die Implementierung der beiden Methoden `count` und `findAll` benötigen.

In den konkreten Unterklassen für `Movie` und `User` können Sie dann die speziellen Finder-Methoden ebenfalls mit Hilfe des Criteria-APIs implementieren.

Ausgabe: 11.10.2021
Abgabe: 18.10.2021