

1. Vraag 1 (10p)

Download de bestanden uit vraag 1

a. Hibernate (2 punten)

Maak twee tabellen aan film en regisseur.

Kolommen regisseur:

- naam

Kolommen film:

- film id (opgelet, dit is niet de id van de tabel)
- titel
- land
- datum toegevoegd
- uitgifte jaar
- omschrijving

Leg ook een relatie tussen regisseur en film, merk op dat een film meerdere regisseurs kan hebben!

b. File import (2 punten)

Importeer bij opstart van je toepassing alle films (dus enkel type movie) uit de CSV naar je database.

Zodra de import klaar is, print je in de console hoeveel films én hoeveel regisseurs je hebt toegevoegd.

Tip: datum staat in het Engels, dus gebruik Locale.ENGLISH bij formatering

c. Console (1 punt)

Maak een console toepassing met volgende opties:

1. Zoeken op jaar
2. Zoeken op titel
3. Zoeken op regisseur
4. Einde

Je console toepassing blijft werken tot je op einde drukt.

d. Zoeken op jaar (2 punten)

Vraag het jaar als input.

Als het jaar in de toekomst ligt maak je een custom exception aan en deze handel je af in je console toepassing. Zorg voor een duidelijke foutboodschap en blijf het jaar vragen tot het correct is.

Schrijf een database query om de films van een jaar te zoeken.

Toon het resultaat in dit formaat:

Titel – Datum toegevoegd (dd/mm/jjjj) – Omschrijving

Bv: Moonwalkers – 08/09/2017 - A brain-addled war vet, a failing band manager and a Stanley Kubrick impersonator help the CIA construct an epic scam by faking the 1969 moon landing.

e. Zoeken op titel (1 punt)

Vraag een deel van een titel en schrijf vervolgens een query zodat dit werkt:

Search “man” -> Results:

- Manhattan Romance
- The Most Assassinated Woman in the World
- Man Up

Toon enkel de titels van de films in je resultaat.

f. Zoeken op regisseur (2 punten)

Vraag de naam van een regisseur. Zoek exact op die naam, niet hoofdlettergevoelig, in je films. **Los dit op met een stream !**

Als er voor de gevraagde regisseur geen films gevonden worden, maak dan een custom exception aan en handel deze af in je console toepassing. Zorg dat je gebruiker een duidelijke foutboodschap krijgt.

Toon het resultaat in dit formaat:

Titel – uitgifte jaar – regisseurs (indien meerdere, gescheiden door komma)

2. Vraag 2 (5p)

Download de code van vraag 2.

De vijf opgaves staan als commentaar in de code.

Los deze vragen op met enkel de technieken gevraagd in de opgave (sortering of streams).

Gebruik je een andere techniek dan gevraagd heb je 0 op die opgave.

3. Vraag 3 (5p)

Download de code van vraag 3.

Schrijf Javadoc op elke method van de BankApp class. (2 punten)

Schrijf nadien unit testen op de **BankApp** class. (3 punten)

Enkele tips voor de oplossing van het proefexamen ...
(bij het examen zal dit er uiteraard niet bij zitten)

Hibernate:

Gebruik een @ManyToOne relatie, vergeet de @JoinTable niet.

File import:

Je kan dit op 2 manieren oplossen.

1. Bij het omzetten van String naar Film haal je de regisseursnamen eruit. Je controleert voor elke naam of deze al bestaat in de database, zoniet maak je de regisseur aan. Je voegt vervolgens het lijstje van Regisseur objecten toe aan je Film.
2. Je werkt met twee loops. In de eerste loop haal je alle unieke regisseursnamen uit de lijst en je maakt deze aan in de database. In de tweede loop heb je de volledige lijst met Regisseur objecten bij aanmaak van je Film en je zoekt in die lijst de regisseurs voor de film op.

Optie 1 is het makkelijkste te schrijven, optie 2 is het snelste in uitvoering (omdat je veel minder de database moet ondervragen).

Pas op bij uitlezen van de regisseursnamen dat er geen voorloopspaties in de naam zitten. Gebruik de .trim() methode van String om enkel de naam over te houden zonder spaties voor en achter de naam.

Output formatting

Als je de gegevens op een andere manier moet tonen dan de standaard toString() output kan je best **niét** de toString() aanpassen. Je maakt best een nieuwe methode aan op het object bv customOutput() waar je de gevraagde output samenstelt.

Zoeken op lijst in lijst via Stream (zoek op naam regisseur van film)

Je kan in je filter argument van je stream opnieuw een stream of regisseurs gebruiken met bv anyMatch . Belangrijkste is dat je filter argument true of false teruggeeft.

Sortering

Oplopend: 1 - 2 - 3 - 4

Aflopend : 4 - 3 - 2 - 1

Alfabetisch : a - b - c - d

Omgekeerd alfabetisch : d - c - b - a