

# Movie Recommender

## Ce este aplicația **Movie Recommender**?

**Movie Recommender** este o aplicație creată în Java, cu ajutorul căreia utilizatorul primește recomandări de filme pe baza unor întrebări și răspunsuri. Atât întrebările, răspunsurile, cât și recomandările sunt salvate în fișiere XML sub formă de reguli. Opțional, utilizatorul poate folosi un fișier XML pentru prescurtările întrebărilor și răspunsurilor. De exemplu: **Ce tip de filme îți plac? → q1.**

Întrebări noi sau răspunsuri noi, cât și prescurtări noi, pot fi adăugate direct în fișierele XML, respectând structura dată, iar aplicația va ști să pună întrebări noi, cu răspunsuri noi, nefiind nevoie de schimbare de cod. De aici rezultă caracterul dinamic al programului.

## Cum poți folosi aplicația?

Având în vedere că proiectul este făcut în Eclipse, iar utilizatorul interacționează cu aplicația prin consolă, este nevoie de **JDK**, **Eclipse**, **Maven** și aplicația, care poate fi găsită pe: <https://github.com/andreimihut95/Movie-Recommender/>. Se pornește Eclipse, se deschide clasa **MainRules** din pachetul **com.furiapolitehnicii.main**. Rulăm programul, utilizatorul poate interacționa cu aplicația prin consolă, poate să selecteze o configurație default, prin apăsarea tastei Y sau o configurație dată de el prin apăsarea tastei N. Prin configurație se înțelege specificarea celor 3 fișiere: fișierul de reguli cu întrebări și răspunsuri, fișierul de reguli cu recomandări și opțional fișierul de mapări pentru prescurtări. După selectarea configurației, urmează o serie de întrebări, unde userul trebuie să răspundă. Utilizatorul poate să răspundă la toate întrebările sau doar la o parte din ele, recomandările primindu-le în oricare din cele două cazuri.

## Cum funcționează aplicația?

Există o clasă **MovieRecommenderRules**, care primește ca și parametrii 3 fișiere(unul fiind opțional, existând doi constructori, unul cu doi parametri si celălalt cu trei parametri), primul

fișier fiind fișierul XML cu întrebări și răspunsuri sub formă de reguli, iar al treilea fișier fiind un fișier de mapări pentru prescurtări. Pe lângă cei trei membri ai clasei, mai există încă doi membri, unul reprezintă o listă de filme(recomandări), pe care clasa o construiește la execuția metodei **loadRecommendations(List<String> states)** unde **List<String> states**, iar celălalt membru o mapă(HashMap), pe care clasa o construiește la execuția metodei **loadMapFile()**. Fișierul XML pentru mapări trebuie să respecte următoare structură:

- numele elementul root al fișierului nu are relevanță(în exemple a fost pus **maps**).
- copiii elementului root trebuie sa fie elemente cu numele de **map**, ce au un atribut cu numele de **id**, având valoarea unei întrebări sau răspuns din celălalt fișier XML, iar ca și conținut text, prescurtarea dorită de creatorul fișierului.
- elementul **map** nu poate sa aibă copii.

Dacă aplicația o sa aibă ca parametru și un fișier de mapări, atunci fișierul XML pentru întrebări și răspunsuri, trebuie să conțină prescurtarea pentru fiecare întrebare și pentru fiecare răspuns, din fișierul de mapări.

Metoda **run()** din cadrul clasei este cea care se ocupă de parsarea fișierului XML pentru întrebări și răspunsuri, și de interacțiunea cu utilizatorul. Fișierul XML, trebuie să respecte următoarea structură:

- numele elementului root nu are relevanță(în exemple a fost pus **movies**)
- copiii elementului root trebuie sa fie elemente cu numele: **rootQuestion** și **rules**.
- **rootQuestion** nu are copii. Are un atribut **question** prin care programul caută prima întrebare și afișează răspunsurile.
- Copiii elementului root trebuie sa fie elemente cu numele: **rule**.
- **rule** nu are copii. Are trei attribute: **if**, **then** și **state** (opțional). În **if** se salvează ori întrebarea, ori răspunsul, ca apoi pe **then** să fie răspunsul pentru întrebarea pusă, sau întrebarea care urmează de la răspunsul de pe **if**, iar in **state** se salvează starea curentă pentru recomandări.

Exemple: `<rule if="Alege durata filmului" then="Peste 2:30" state="FilmLung" />`

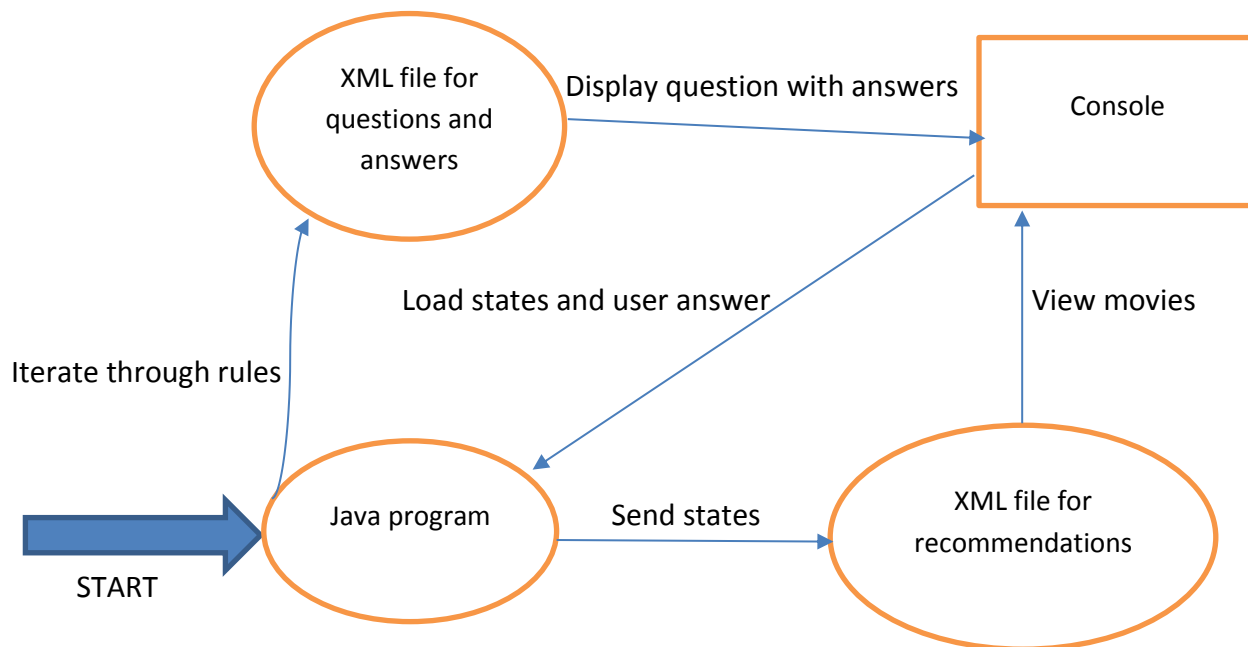
`<rule if="Peste 2:30" then="Castigator la oscar?" />`

In funcție de **if** si **then** aplicația știe sa parcurgă regulile, și sa formeze un arbore, astfel încât întrebările cu răspunsurile corespunzătoare sunt afișate în consolă. De câte ori utilizatorul dă un răspuns, de atâtea ori se salvează într-o listă starea curentă. Lista cu stările salvate vor fi pasate la o metodă **loadRecommendations(List<String> states)**, care va încărca în membrul clasei(lista de recomandări) toate recomandările care respectă stările date ca parametru. Încărcarea

recomandărilor în lista se face printr-o parsare în fișierul XML pentru recomandări. Fișierul XML pentru recomandări trebuie să aibă următoarea structură:

- numele elementului root nu are relevanță(în exemple a fost pus **recommendations**)
- copiii elementului root trebuie să fie elemente cu numele: **if**.
- **if** are un atribut cu numele **state**, unde se salvează, concatenat, toate stările unei recomandări. Exemplu: un film ce are durată mare, de acțiune, câștigător la oscar ar fi **state="Actiune-DaOscar-FilmLung"**.
- **if** are ca și copii elemente cu numele **recommendation**(doar un fiu) .
- **recommendation** nu are copii, are doar atribute cu numele de: **title**, **imdbRating**, **imdbURL**, **imdbPhoto**, iar ca și conținut text va avea o descriere.

Tehnica utilizată pentru a parcurge arborele este "forward chaining". Aceasta constă în determinarea concluziilor (frunzele arborelui) pornind de la premise (răspunsurile date de utilizator). Pentru afișarea recomandărilor, clasa dispune de o metodă numită **viewRecommendations()**, care afișează recomandările salvate.



## Tehnologii, API-uri, Framework-uri, IDE-uri folosite

Următoarele au fost folosite pentru crearea aplicației: **Eclipse**, **Maven**, **Java**, **Git**, **XML**, **xPath**, **DOM4j** și **Jaxen**.