ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προγραμματισμός Υπολογιστών με Java
Εαρινό Εξάμηνο 2024-2025
2ο Μέρος Εργασίας

🎬 Σενάριο / Κίνητρο

Η πλατφόρμα, έχοντας πλέον βασική υποδομή για την αναπαράσταση ταινιών, χρηστών και κριτικών, επιθυμεί να αξιοποιήσει πραγματικά δεδομένα με στόχο την εξαγωγή στατιστικών, την κατανόηση των προτιμήσεων των χρηστών και τη βελτίωση των μηχανισμών σύστασης. Οι χρήστες συνεχίζουν να αξιολογούν ταινίες με αριθμητική βαθμολογία και σύντομα σχόλια, και η εταιρεία θέλει να οργανώσει και να αναλύσει αυτές τις πληροφορίες σε μεγαλύτερη κλίμακα.

Ερωτήματα που την απασχολούν είναι:

- Ποιες είναι οι δημοφιλέστερες ταινίες ανά είδος με βάση τις αξιολογήσεις;
- Πώς μπορούν να εντοπιστούν οι ταινίες με τη μεγαλύτερη αποδοχή από τους χρήστες;
- Ποιες είναι οι προτιμήσεις κάθε χρήστη;
- Ποιες ταινίες θα άξιζε να προταθούν σε έναν χρήστη με βάση τις προηγούμενες αξιολογήσεις του ή τις αξιολογήσεις άλλων με παρόμοιο προφίλ;

Για την υποστήριξη αυτών των στόχων, ζητείται να επεκταθεί το αντικειμενοστραφές σύστημα ώστε:

- να διαβάζει τα δεδομένα από αρχεία,
- να οργανώνει και να ομαδοποιεί ταινίες και χρήστες βάσει βαθμολογιών,
- να εξάγει χρήσιμα στατιστικά,
- και να υποστηρίζει μηχανισμούς σύστασης με βάση την ομοιότητα χρηστών και το περιεχόμενο των ταινιών.

Σκοπός

Στόχος του δεύτερου μέρους είναι η επεξεργασία δεδομένων από αρχείο και η ανάπτυξη μηχανισμών στατιστικής ανάλυσης και σύστασης ταινιών. Καλείστε να επεκτείνετε το αντικειμενοστραφές μοντέλο σας, να αξιοποιήσετε συλλογές και τεχνικές ομαδοποίησης, και να υλοποιήσετε βασικές μορφές **collaborative filtering** (αν δύο χρήστες έχουν παρόμοιο γούστο, τότε οι ταινίες που άρεσαν στον έναν μπορεί να αρέσουν και στον άλλον) και **content-based filtering** (αν σε έναν χρήστη αρέσουν ταινίες με συγκεκριμένα χαρακτηριστικά, τότε προτείνουμε άλλες ταινίες με παρόμοια χαρακτηριστικά) για την υποστήριξη λειτουργιών πρότασης. Το πρόγραμμα θα πρέπει να διαχειρίζεται **εξαιρέσεις**.

Διαθέσιμο αρχείο: `reviews.csv`

Περιέχει γραμμές με πληροφορία για:

- **Ταινία**: τίτλος, έτος κυκλοφορίας, είδος (genre)
- **Χρήστη**: μοναδικό αναγνωριστικό ή όνομα
- **Κριτική**: αριθμητική βαθμολογία (π.χ. 1–10) και σχόλιο

Απαιτούμενα Βήματα

1. Ανάγνωση και Δημιουργία Αντικειμένων

- Να διαβάσετε το αρχείο `reviews.csv`
- Να δημιουργηθούν κατάλληλα αντικείμενα:
  - `Movie` (τίτλος, έτος, είδος, λίστα κριτικών)
  - `User` (όνομα/ID, λίστα κριτικών)
  - `Review` (ταινία, χρήστης, βαθμολογία, σχόλιο)

2. Οργάνωση και Επεξεργασία

- Να οργανωθούν οι ταινίες:
  - **Ανά είδος (genre)**
    - Χρησιμοποιήστε έναν `Map<String, List<Movie>>`, όπου το `String` είναι το είδος (π.χ. "Action") και η `List<Movie>` περιέχει τις αντίστοιχες ταινίες.
    - Αν μια ταινία ανήκει σε **πολλαπλά είδη** (π.χ. `"Action|Sci-Fi"`), μπορείτε να υποθέσετε ότι ανήκει **σε ένα μόνο είδος.** Εναλλακτικά, μπορείτε να τη **διαχωρίσετε** (π.χ., `split("|")`) και να την εισάγετε **σε κάθε είδος ξεχωριστά,** αλλά πρέπει να **μην διπλο-προσμετρώνται** σε συγκεντρωτικά στατιστικά (π.χ., πλήθος ταινιών).
  - **Κατά μέση βαθμολογία και κατά πλήθος κριτικών**
    - Μπορείτε να ταξινομήσετε κάθε λίστα (`List<Movie>`) με χρήση `Comparator<Movie>` ως προς `getAverageRating()` ή `getReviewCount()`.
    - Προαιρετικά, μπορείτε να χρησιμοποιήσετε `TreeMap` ή `PriorityQueue` αν θέλετε αυτόματη διατήρηση ταξινόμησης.

3. Στατιστικά Ερωτήματα

- Οι **Top-5 ταινίες ανά είδος** με βάση τη μέση βαθμολογία.
- **Μέση βαθμολογία ανά χρήστη**
- **Ταινίες με "υψηλή αποδοχή"**: Ταινίες που έχουν βαθμολογηθεί με >7 από τουλάχιστον το 80% των χρηστών που τις είδαν.

---

4. Υλοποίηση Προτάσεων Ταινιών (Recommender System)

Υλοποιήστε δύο απλούς μηχανισμούς σύστασης:

α. Collaborative Filtering (User-Based)

- Υπολογίστε **ομοιότητα χρηστών** (π.χ. συσχέτιση ή μέση διαφορά βαθμολογιών σε κοινές ταινίες)
- Βρείτε τους πιο όμοιους χρήστες και προτείνετε στον χρήστη ταινίες που έχουν αυτοί βαθμολογήσει ψηλά και εκείνος δεν έχει δει.

β. Content-Based Filtering

- Για έναν χρήστη, αναλύστε τα είδη και τις βαθμολογίες των ταινιών που έχει ήδη δει.
- Προτείνετε ταινίες από είδη που του αρέσουν, με βάση τον μέσο όρο των κριτικών του σε αυτά τα είδη.

Υποδείξεις

- Μπορείτε να χρησιμοποιήσετε `Map<String, List<Movie>>` για την ομαδοποίηση κατά είδος.
- Για την εύρεση των Top-5, μπορείτε να χρησιμοποιήσετε `Comparator` και `Collections.sort()`.
- Ο υπολογισμός της ομοιότητας μπορεί να γίνει με τη μέση τετραγωνική διαφορά (mean squared error). Όσο μικρότερη είναι η διαφορά αυτή, τόσο πιο παρόμοιες είναι οι αξιολογήσεις των χρηστών. Στην ειδική περίπτωση της μηδενικής διαφοράς, οι χρήστες έχουν δώσει ακριβώς τις ίδιες βαθμολογίες σε όλες τις κοινές ταινίες. Μπορείτε να θεωρήσετε παρόμοιες τις προτιμήσεις δύο χρηστών αν έχουν MSE < 4.

  Παρακάτω δίνεται ένα παράδειγμα με δύο χρήστες και τον υπολογισμό του MSE:

Ταινία	**User A**	**User B**
The Matrix	9	10
La La Land	7	6
Joker	8	7

  MSE = $[(9-10)^2 + (7-6)^2 + (8-7)^2] / 3 = (1 + 1 + 1) / 3 =$ **1.0** ➔ οι χρήστες έχουν παρόμοιες προτιμήσεις, γιατί MSE < 4.

- Μπορείτε να δημιουργήσετε μία κλάση `Recommender` με τις δύο μεθόδους πρότασης (collaborative and content-based filtering, δείτε και το προσχέδιο του πηγαίου κώδικα).

- Αν χρησιμοποιήσετε δομές όπως `HashSet<Movie>` ή `Map<Movie, …>`, πρέπει να υλοποιήσετε τις μεθόδους `equals()`και `hashCode()`στην κλάση Movie. Η σύγκριση ταινιών μπορεί να βασίζεται σε **τίτλο και έτος**, π.χ.,

```java
@Override
public boolean equals(Object obj) {
    if (this == obj) return true;
    if (!(obj instanceof Movie)) return false;
    Movie other = (Movie) obj;
    return this.title.equals(other.title) && this.year == other.year;
}

@Override
public int hashCode() {
    return Objects.hash(title, year);
}
```

Οδηγίες

Η εργασία είναι ομαδική και μπορεί να υλοποιηθεί από ομάδες δύο ή τριών φοιτητών. Παρακάτω, δίνεται ένας σκελετός πηγαίου κώδικα, στον οποίο μπορείτε να βασιστείτε, αν θέλετε. Παρακαλούμε ακολουθήστε τις παρακάτω οδηγίες:

1. Η εργασία να παραδοθεί ως Java project με:

    1.1. Τεκμηριωμένο πηγαίο κώδικα.

    1.2. README με περιγραφή σχεδίασης, λειτουργικότητας και οδηγιών εκτέλεσης.

    1.3. Main.java με αντιπροσωπευτικά παραδείγματα λειτουργίας (εσείς επιλέγετε το πώς θα λειτουργεί η `main`).

2. Σύνθεση ομάδας: Τα μέλη μια ομάδας μπορούν να προέρχονται είτε από το τμήμα Α-Λ, είτε από το Μ-Ω, είτε και από τα δύο τμήματα (Α-Λ και Μ-Ω). Η σύνθεση των ομάδων πρέπει να είναι η ίδια και για τις δύο ομαδικές εργασίες (1ο μέρος και 2ο μέρος).
3. Δήλωση ομάδας: Δεν χρειάζεται να γίνει εκ των προτέρων δήλωση των μελών της ομάδας. Πρέπει όμως να γράψετε ως σχόλιο, στην αρχή του αρχείου (Main) που περιέχει την κύρια μέθοδο, τα προσωπικά στοιχεία (επώνυμο, όνομα, αριθμό μητρώου και διεύθυνση ηλεκτρονικού ταχυδρομείου) ΟΛΩΝ ΤΩΝ ΜΕΛΩΝ ΤΗΣ ΟΜΑΔΑΣ. Το επώνυμο και το όνομα σας παρακαλούμε να το σημειώσετε με λατινικούς χαρακτήρες.
4. Υλοποίηση εργασίας: Κατά την υλοποίηση της εργασίας σας, παρακαλούμε να λάβετε υπόψη τα παρακάτω:

    4.1. Ονομάστε την κλάση που περιέχει τη κύρια μέθοδο της εφαρμογής σας Main και το αρχείο Main.java. Μην ξεχάσετε εδώ να σημειώσετε ως σχόλιο τα προσωπικά στοιχεία όλων των μελών της ομάδας σας.
    4.2. Μη χρησιμοποιείτε ελληνικούς χαρακτήρες ούτε ως σχόλια, ούτε ως μηνύματα προς το χρήστη.

4.3. Μην ομαδοποιήσετε τις κλάσεις σας σε πακέτα κλάσεων.
4.4. Ο κώδικας που θα υποβάλετε θα πρέπει να μεταγλωττίζεται και να εκτελείται από τη γραμμή εντολών, χωρίς τη χρήση κάποιου περιβάλλοντος (IDE).
4.5. Συμπιέστε τα αρχεία της εφαρμογής σας (java files) σε ένα αρχείο με όνομα τους αριθμούς μητρώου των μελών της ομάδας, για παράδειγμα (3230001_3230002_3230003.zip). Υποβάλετε το συμπιεσμένο αρχείο στο eclass.

5. Υποβολή εργασίας: Κάθε μέλος της ομάδας υποβάλλει ατομικά ένα αντίγραφο της εργασίας στο eclass. Δηλαδή, αν για παράδειγμα μια ομάδα αποτελείται από τρεις φοιτητές τότε και οι τρεις πρέπει να υποβάλλουν την – ίδια – εργασία στο eclass (ο κάθε ένας στο τμήμα που ανήκει Α-Λ ή Μ-Ω). Σε κάθε αντίγραφο της εργασίας, παρακαλούμε να αναφέρονται τα προσωπικά στοιχεία όλων των μελών της ομάδας.

Τέλος, υπενθυμίζεται ότι έχετε το δικαίωμα να διατηρήσετε βαθμό εργασιών που τυχόν έχετε υλοποιήσει σε προηγούμενο ακαδημαϊκό έτος. Στην περίπτωση αυτή, ζητείται να υποβάλετε ένα αρχείο κειμένου (word ή text), στο οποίο να σημειώνετε μόνο τα προσωπικά σας στοιχεία, καθώς και το ακαδημαϊκό έτος κατά το οποίο κάνατε τις ομαδικές εργασίες. Επίσης, να σημειώσετε τα παραπάνω και στο γραπτό σας, στο τελικό διαγώνισμα του Ιουνίου 2025 ή/και Σεπτεμβρίου 2025.

Προσχέδιο UML (επιτρέπεται να αλλάξει)