02.12.2024

Design a Feature Film Management System that manages feature films, actors, production companies, viewer reviews, and movie recommendations. The system should also simulate revenue calculations and dynamically update popularity rankings. The project must incorporate at least six data structures: Linked Lists, Queues, Stacks, Hash Tables, Heaps, and Binary Search Trees (BSTs).

Requirements:

1. Film and Actor Management:

- Create a Film class with attributes: Film name, Unique Film ID, Actor list (Linked List), Genre, Release year, Total
revenue.
- Methods: Add, delete, and update films.
- Create an Actor class with attributes: Actor name, Unique Actor ID, Films participated in (Linked List).
- Methods: Add and update actor profiles.

2. Viewer Feedback System:

- Allow users to rate and review films.
- Ratings influence the film's popularity rankings.

3. Revenue Simulation:

- Simulate revenue based on viewer count and ticket price.
- Use Queue: Schedule and manage film screenings.
- Use Stack: Store historical revenue data for auditing and trends.

4. Popularity Rankings:

- Dynamically rank films based on revenue and viewer feedback.
- Resolve ties using secondary factors such as release year. - Use Heap: Efficiently manage and update rankings.

5. Search Functionality:

- Allow users to search for films or actors by name.
- Use Binary Search Tree (BST): Efficiently perform searches and sort results.

6. Additional Features:

- Revenue history retrieval: Retrieve the last n revenue calculations.
- Recommendations: Recommend films based on genre or viewer ratings.

<u>Features:</u>

Core Features:

- Film and Actor Management: Add new films and actors. Update actor filmography and film cast details.

- Revenue Simulation: Calculate revenue based on user-defined parameters. Store revenue data for each film.

- Popularity Rankings: Display films ranked by revenue and other factors. Resolve ties using release year.

- Search Functionality: Quickly find films or actors.

- Revenue History: Retrieve the last n revenue calculations.

Advanced Features:

- Viewer Feedback System: Allow viewers to rate and review films. Integrate ratings into popularity rankings.

<u>Data Structures and Their Uses:</u>

1. Linked List: Store actor details for each film and film details for each actor. Grow or shrink dynamically.
2. Queue: Manage film screening schedules. Films are processed in the order they are queued.
3. Stack: Store revenue calculations chronologically. Provide quick access to the latest revenue data.
4. Hash Table: Efficiently access film and actor data using unique IDs. Store key-value pairs for fast lookups.
5. Heap: Efficiently manage film popularity rankings. The most popular film is always at the root.
6. Binary Search Tree (BST): Efficiently search for films or actors by name. Store data in sorted order.

Deliverables:

1. Code Files: All Java classes (.java files) implementing the required functionalities.
2. Documentation: Explain the purpose of each class and method. Provide a user guide to run the program.
3. Report: A detailed report discussing:

- The data structures used and the justification for their selection.
- Challenges faced during the implementation.
- Examples and screenshots demonstrating the functionality.

Sample Workflow:

1. Film and Actor Addition: Add films and their respective actors. Store details in a hash table for quick access.
2. Revenue Simulation: Enqueue films for screening. Simulate revenue generation and push revenue data onto the stack.
3. Popularity Rankings Update: Use a heap to dynamically update and display rankings.
4. Search for Films or Actors: Use the BST to efficiently search names.
5. Revenue History Retrieval: Retrieve revenue calculations from the stack.

Notes:

- The project can be done in groups of maximum 3 people. In this case, it is sufficient for one student from the group to upload the assignment.
- Only assignments sent via Teams will be evaluated. Do not send assignments via e-mail or message.
- In case of cheating, assignments that comply with the detection will not be evaluated.