

Movie Rating System

Objective:

Create a system to rate movies, built using JavaScript ES6 classes, encapsulating related data and behavior, demonstrating inheritance, and showcasing polymorphism.

Exercise 1: The Media superclass

1.1 Start by defining a `Media` class. The constructor should accept two parameters, `title` and `duration`.

1.2 Inside the constructor, initialize instance variables for `title`, `duration`, and `ratings`. Set `ratings` as an empty array as we'll be collecting multiple ratings for a media object.

1.3 Add a method named `addRating` which accepts a parameter `rating`. This method should push the received `rating` into the `ratings` array.

1.4 Implement a `calculateAverageRating` method that calculates the average of the `ratings` array. If there are no ratings, it should return 'No ratings yet'. Use the `reduce` method to calculate the sum of the ratings, and then divide by the length of the `ratings` array to find the average. Return the average rating, rounded to two decimal places.

1.5 Lastly, create a `displayDetails` method that returns a string containing the media's `title`, `duration`, and average rating.

Exercise 2: The Movie and Series subclasses

2.1 Create a `Movie` subclass that extends `Media`. The constructor should accept additional parameters `director` and `genre`, and pass `title` and `duration` to the `super` constructor.

2.2 Override the `displayDetails` method to include `director` and `genre` in the string it returns. Call `super.displayDetails()` to include the details provided by the superclass.

2.3 Similarly, create a `Series` subclass of `Media` with an additional `seasons` parameter in its constructor.

2.4 Override the `displayDetails` method in `Series` to include the number of `seasons`. Again, use `super.displayDetails()` to leverage the superclass's implementation.

Exercise 3: The User class

3.1 Define a `User` class with a constructor that accepts a single parameter, `username`.

3.2 In the constructor, initialize instance variables for `username` and `watchedMedia`, with `watchedMedia` set as an empty array.

3.3 Implement an `addMedia` method that accepts a `media` object and adds it to the `watchedMedia` array.

3.4 Add a `rateMedia` method that takes a `media` object and a `rating`. The method should check if the `media` object is in the `watchedMedia` array. If so, it should call the `addRating` method of the `media` object with the given `rating`. If not, it should return a message indicating that the user can only rate media they've watched.

3.5 Lastly, create a `displayWatchedMedia` method that returns a string containing all the media the user has watched, including their details. Loop through the `watchedMedia` array and for each media item, call its `displayDetails` method. Join the resulting array into a single string, separated by line breaks.