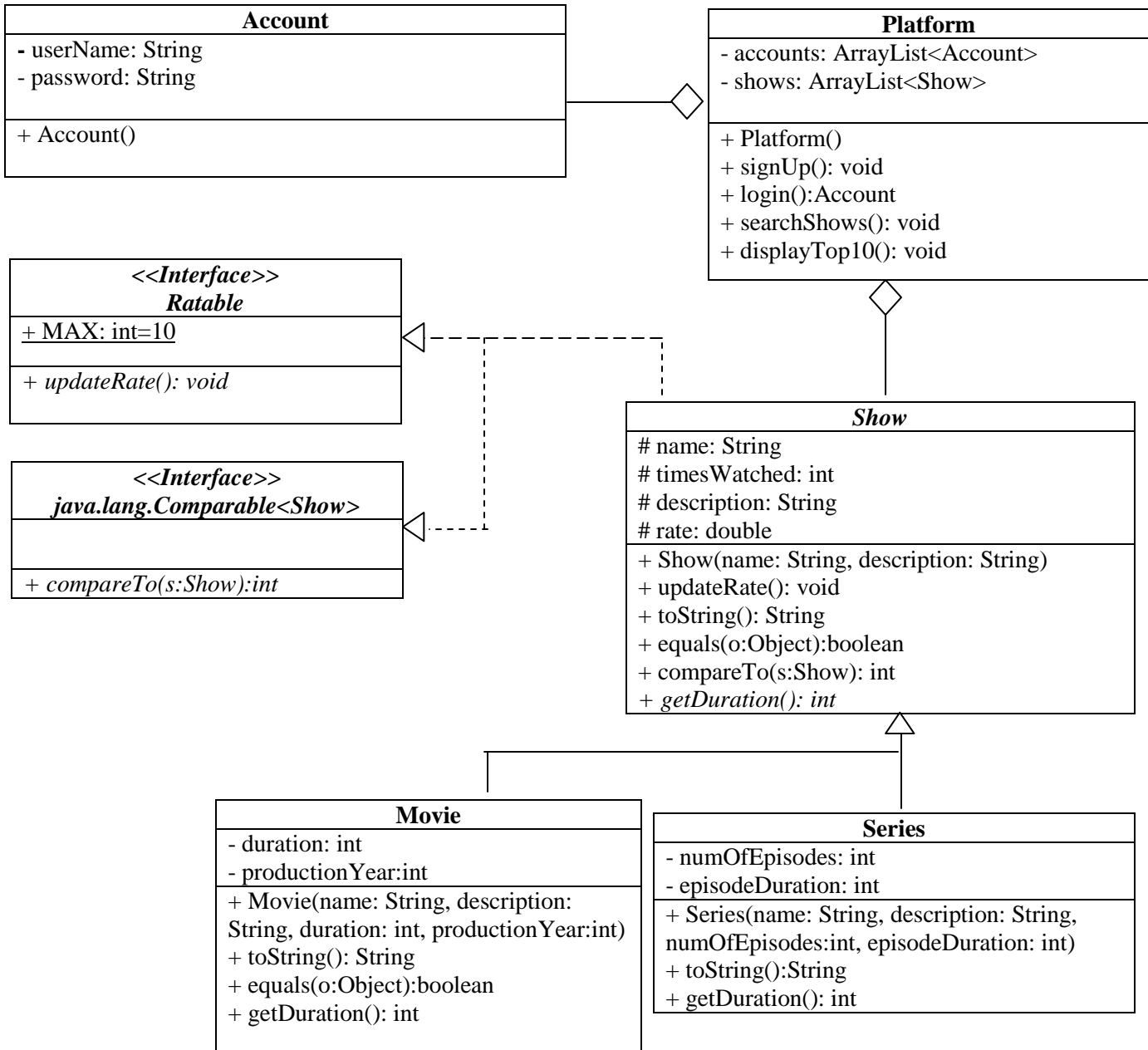




University Of Jordan
Computer Engineering Department
Object-Oriented Problem Solving Lab
Final Exam – Fall2020
Eng. Asma Abdel Karim

❖ Write a JAVA program to build an appropriate implementation for the UML diagram below, which models a platform for watching different types of shows.



Getters and setters are omitted from the UML diagram. You may add any getters/setters you need.

● **In the *Show* class:**

- The constructor initializes the *name* and *description* data fields with the passed parameters. The *timesWatched* and *rate* must be initialized to zero.
- The *updateRate* method should read the user rate from the user in an input dialog box and re-compute the rate data field. Note that:
 - The rate is always computed as the average rates entered by the users.
 - Assume that all users who watched a show must rate it, which means that the number of times a show is watched is the same as the number of times it is rated.
 - If the user does not enter a rate between 0 and the max defined by the *Ratable* interface, your code should keep on asking the user to enter a rate in the valid range.
 - The *updateRate* method should also increment the number of times the show is watched (*timesWatched* data field).
- The *toString* method must be overridden such that it returns a *String* that contains the name and rate on one line, the description on the next line in the following format:
name – rate
description
- The *equals* method must be overridden such that two shows are equal if they have the same name. If an object of type other than show is passed, it must return false.
- The *compareTo* method must be overridden such that the rates of the two objects are compared, and returns 1 if the first object rate is larger than the second, 0 if they have equal rates, and -1 if the first object rate is less than the second.

● **In the *Movie* class:**

- The constructor must invoke the super class constructor in order to initialize inherited data fields.
- The *toString* method must be overridden such that it invokes the super class *toString* method and add to it on a new line the duration and production year.
- The *equals* method must be overridden such that two movies are equal if they have the same name (invoke super class *equals* method), and if they have the same production year.
- The *getDuration* method must be overridden to return the duration data field.

● **In the *Series* class:**

- The constructor must invoke the super class constructor in order to initialize inherited data fields.
- The *toString* method must be overridden such that it invokes the super class *toString* method and add to it on a new line the number of episodes and the episode duration.
- The *getDuration* method must be overridden to return the duration of all series episodes by multiplying the number of episodes by the episode duration.

● **In the *Account* class:**

- The *ArrayList* of shows must be initialized to be empty.
- The constructor must ask the user to enter his user name and password in input dialog boxes.

- **In the Platform class:**

- The *ArrayLists* must be both initialized to be empty.
- The *signUp* method must add a new *Account* object (created using the no-arg constructor) to the *ArrayList accounts*.
- The login method must ask the user to enter his user name and password using input dialog boxes. It should search for an account with the entered username if found, it should match the entered password with the password of the account, if the passwords are matched it should return a reference to the account. Note the login may fail in two cases:
 1. If no account is found with the entered username, in this case a message dialog box with the message "Cannot find username:"
 2. If the passwords do not match, it should display a message dialog box with the following message "Wrong password".
 3. In either case, if the login fails, the method should return null.
- The *searchShows* method must ask the user in an input dialog box if he's searching for movies or series, then it should ask the user to enter a keyword in another input dialog box. Then, based on the selected type (movie or series), it should print the names of all movies/series whose descriptions contain the entered keyword.
- The *displayTop10* method must print the *toString* method of the ten top rated shows. Note that this method must use the *sort* method of the *Arrays* class which will in turn use the *compareTo* method to sort your shows.

- **In your main class:**

Define a method named *operatePlatform* that takes an object of type platform, and:

- Continuously ask the user in an input dialog box what he wants to do on the platform, the options are:
 1. Sign up: invoke the corresponding method *signUp* on the platform.
 2. Login: invoke the corresponding method *login* on the platform. Note that this method may return a reference to an account if the login is successful and null if the login is not successful.
 - If the login is not successful the program should go back to the input dialog box of the two options: sign up or login.
 - If the login is successful, a new list of options must appear continuously in a message dialog box, that includes:
 1. Search for show: invoke the corresponding method.
 2. Display top 10: invoke the corresponding method.
 3. Logout: it should exit the currently viewed options and return to the original options of sign up or login.

Good Luck ☺