BIRZEIT UNIVERSITY

Engineering and Information Technology Faculty

Computer Science Department

Comp231-Advanced Programming

Final Project- Phase I

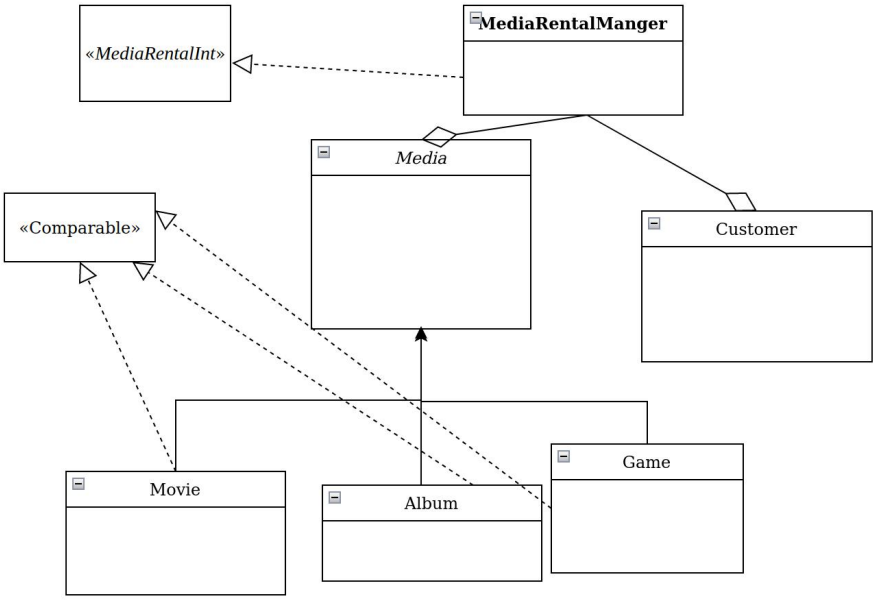
(Be Ready for the second phase- It depends on the first phase, so you have to submit the first on time)

Deadline:Monday 20/12/2021 before midnight.

Objectives:

(Chapter 1 - Chapter 13)

This project goal is to develop a rental media system This underneath UML diagram may help you to understand the system requirements



Specifications

What You Must Implement

You must define a class named **MediaRental** that implements the *MediaRentalInt interface* functionality (index A). You must define classes that support the functionality specified by the interface. The following specifications are associated with the project:

1. Define a class named **MediaRental**. Feel free to add any instance variables you understand are needed or any private methods. Do not add any public methods (beyond the ones specified in the *MediaRentalInt* interface).
2. The media rental system keeps track of customers and media (movies ,music albums and games). A customer has a name, address as string , a plan and two lists. One list represent the media the customer is interested in receiving and the second one represents the media already received (rented) by the customer. There are two plans a customer can have: UNLIMITED and LIMITED. UNLIMITED allows a customer to receive as many media as they want; LIMITED restricts the media to a default value of 2 (this value can be change via a media rental class method).
   A movie has a title, a number of copies available and a rating (e.g., "HR"). An album has a title, number of copies available, an artist and the songs that are part of the album. A game has title,number of copies,and the weight (in grams) is also stored.
3. You must define and use at least four classes (not including MediaRental) as part of your design. At least two of those classes must be in a superclass/subclass relationship (Inheritance Requirement). The other two can be defined as you wish. Feel free to define as many classes as you want. **These classes must support the functionality of the system otherwise you will not receive any credit.**
4. One of your classes must define an equals method that has as parameter an Object parameter.
5. The database for your system needs to be represented using two ArrayList objects. One ArrayList will represent the customers present in the database; the second will represent the media (movies, albums , and games).
6. Regarding the *searchMedia* method: the songs parameter represents a substring (fragment) or the full list of songs associated with the album. If the full list is provided you can assume commas will be part of the string. Hint: you may want to consider using the indexOf method of the String class.
7. Your program should store the results in a file between executions of the program, so that when the program is run again it will start up with the same inventory contents as when it last terminated.
8. Feel free to use Collections.sort to sort your data.
9. Write test driver.
10. Handle expectations where it's needed.
11. Not all the details associated with the project can be fully specified in this description. The sooner you start working on the project the sooner you will be able to address any doubts you may have.

Note: To understand the requirement, you might lock at the indices (A,B, and C)

Good Luck

Index A:

MediaRentalInt interface Includes:

Method Detail

addCustomer

void **addCustomer**(String name,String address,String plan)

Adds the specified customer to the database. The address is a physical address (not e-mail). The plan options available are: **LIMITED** and **UNLIMITED**. LIMITED defines a default maximum of two media that can be rented.

Parameters: name - ,address -,plan -

---

addMovie

void **addMovie**(String title, int copiesAvailable,String rating)

Adds the specified movie to the database. The possible values for rating are "DR", "HR", "AC".

Parameters:title - ,copiesAvailable -,rating -

---

addGame

void **addGame**(String title, int copiesAvailable,double weight)

Adds the specified game to the database.

Parameters:title - ,copiesAvailable -,weight -

---

addAlbum

void **addAlbum**(String title,int copiesAvailable,String artist,String songs)

Adds the specified album to the database. The songs String includes a list of the title of songs in the album (song titles are separated by commas).

Parameters: title - ,copiesAvailable -,artist -,songs -

---

setLimitedPlanLimit

void **setLimitedPlanLimit**(int value)

This set the number of media associated with the LIMITED plan.

Parameters: value -

---

getAllCustomersInfo

String **getAllCustomersInfo**()

Returns information about the customers in the database. The information is presented sorted by customer name.

Returns:

---

getAllMediaInfo

String **getAllMediaInfo**()

Returns information about all the media (movies, albums, and games) that are part of the database. The information is presented sorted by media title.

Returns:

---

addToCart

boolean **addToCart**(String customerName,String mediaTitle)

Adds the specified media title to the cart associated with a customer.

Parameters:customerName - ,mediaTitle -

Returns: false if the mediaTitle is already part of the cart (it will not be added)

---

removeFromCart

boolean **removeFromCart**(String customerName, String mediaTitle)

Removes the specified media title from the customer's cart.

Parameters:customerName - ,mediaTitle -

Returns: false if removal failed for any reason (e.g., customerName not found)

---

processRequests

String **processRequests**()

Processes the requests cart of each customer. The customers will be processed in alphabetical order. For each customer, the requests cart will be checked and media will be added to the rented cart, if the plan associated with the customer allows it, and if there is a copy of the media available. For UNLIMITED plans the media will be added to the rented cart always, as long as there are copies associated with the media available. For LIMITED plans, the number of entries moved from the requests cart to the rented cart will depend on the number of currently rented media, and whether copies associated with the media are available.

For each media that is rented, the following message will be generated:
"Sending [mediaTitle] to [customerName]"

Returns:

---

returnMedia

boolean **returnMedia**(String customerName,String mediaTitle)

This is how a customer returns a rented media. This method will remove the item from the rented cart and adjust any other values that are necessary (e.g., copiesAvailable)

Parameters:customerName - ,mediaTitle -

Returns:

---

searchMedia

ArrayList<String> **searchMedia**(String title,String rating, String artist,String songs)

Returns a SORTED ArrayList with media titles that satisfy the provided parameter values. If null is specified for a parameter, then that parameter should be ignore in the search. Providing null for all parameters will return all media titles.

Parameters:

title - ,rating - ,artist -, songs -

Returns:

Index B:

Public Driver:

Method Detail

public void **testAddingCustomers**()
public void **testAddingMedia**()
public void **testingAddingToCart**()
public void **testingRemovingFromCart**()
public void **testProcessingRequestsOne**()
public void **testProcessingRequestsTwo**()
public void **testReturnMedia**()
public void **testSearchMedia**()

Index C:

Academic Integrity

Please make sure you read the academic integrity section of the syllabus so you understand what is permissible in our programming projects. We want to remind you that we check your project against other students' projects and any case of academic dishonesty will be referred to **fail in the course and the student is referred to a system committee.**

NOTE: THERE IS PASS OF (There is a face-to-face discussion)