**Project**

**Media Rental System**

Before attempting this project, be sure you have completed all the reading assignments, non-graded exercises, examples, discussions, and assignments to date.

**Design and implement Java program as follows:**

1) **Media hierarchy**:

Create *Media, EBook, MovieDVD, and MusicCD* classes from Week 3 -> Practice Exercise - Inheritance solution.

Add an attribute to *Media* class to store indication when media object is rented versus available. Add code to constructor.

Add any additional constructors and methods needed to support the below functionality.

2) Design and implement **Manager** class which (**Hint**: check out Week 8 Reading and Writing files example):

stores a list of Media objects

has functionality to load Media objects from files creates/updates Media files

has functionality to add new Media object to its Media list

has functionality to find all media objects for a specific title and returns that list

has functionality to rent Media based on id (updates rental status on media, updates file, returns rental fee)

3) Design and implement **MediaRentalSystem** which has the following functionality:

user interface which is either menu driven through console commands or GUI buttons or menus. Look at the bottom of this project file for sample look and feel. (Hint: for command-driven menu check out Week 2: Practice Exercise - EncapsulationPlus and for GUI check out Week 8: Files in GUI example)

selection to load Media files from a given directory (user supplies directory)

selection to find a media object for a specific title value (user supplies title and should display to user the media information once it finds it - should find all media with that title)

selection to rent a media object based on its id value (user supplies id and should display rental fee value to the user)

selection to exit program

4) Program should throw and catch Java built-in and user-defined exceptions as appropriate 5) Your classes must be coded with correct encapsulation: private/protected attributes

and value validation

6) There should be appropriate polymorphism: overloading, overriding methods, and dynamic binding

7) Program should take advantage of the inheritance properties as appropriate

**Style and Documentation:**

Make sure your Java program is using the recommended style such as:

Javadoc comment up front with your name as author, date, and brief purpose of the program

Comments for variables and blocks of code to describe major functionality

Meaningful variable names and prompts

Class names are written in upper CamelCase Constants are written in All Capitals

Use proper spacing and empty lines to make code more user-friendly, readable

**Capture execution:**

You should capture and label screen captures associated with compiling your code and running a passing and failing test cases for each functionality.

[Review the various screen shots after the Rubrics.]

**Assignment Deliverables:**

Deliverables include a single Java program (.java) and a single Word (or PDF) document. No zipped files are acceptable. The Java and Word/PDF files should be named appropriately for the assignment (as indicated in the Submission Requirements document posted in Week 1): CMIS242FP[name].java, CMIS242FP[name].docx, or CMIS242FP[name].pdf.

Though you might construct the assignment with multiple classes, the final submission should be a single Java program. The process of combining multiple classes into a single Java program is found in Week 1 Content under the Practice Exercise thread and titled Practice Exercise as ONE Java program. Submit the combined Java program as the attachment for Assignment 4.

The word (or PDF) document should include a Test Plan with multiple test cases. Each test case should be referenced to a screen capture showing the successful compiling and running of each of the test case. Each screen capture should be properly labeled clearly indicated what the screen capture represents.

Submit your files to Assignment 4 submission area no later than the due date listed in your online classroom.

**Grading Rubric:**

The following grading rubric will be used to determine your grade: Note: Elements Included in the Grading of the Program document contains additional grading elements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | **Level 3**  20 points | **Level 2**  15 points | **Level 1**  5 points | **Criterion Score** |
| Criterion 1 Add | Correct or almost correct code to meet required functionality | Mistakes in implementation | Missing or significantly incorrect implementation | / 20 |
| Criterion 2 Remove | Correct or almost correct code to meet required functionality | Mistakes in implementation | Missing or significantly incorrect implementation | / 20 |
| Criterion 3 Find | Correct or almost correct code to meet required functionality | Mistakes in implementation | Missing or significantly incorrect implementation | / 20 |
| Criterion 4 Display | Correct or almost correct code to meet required functionality | Mistakes in implementation | Missing or significantly incorrect implementation | / 20 |
| Criterion 5 Menu, program documentation and style, screen captures | Correct or almost correct menu, program comments, identifiers, and screen captures. | Mistakes or incomplete menu, documentation and/or style, and screen captures | Missing or significantly incorrect menu, documentation and/or style, or screen captures | / 20 |
| Total |  |  |  | / 100 |

**Sample User interface - command driven:**

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 1

Enter path (directory) where to load from: blah

File cannot be opened: Could not load, no such directory

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 1

Enter path (directory) where to load from: C:/tmp-umuc

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 2

Enter the title: blah

There is no media with this title: blah

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 2

Enter the title: Forever Young

EBook [ id=123, title=Forever Young, year=2018, chapters=20 available=true] MovieDVD [ id=126, title=Forever Young, year=2020, size=140.0MB available=false]

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 3

Enter the id: 123

Media was successfully rented. Rental fee = $2.00

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 2

Enter the title: Forever Young

EBook [ id=123, title=Forever Young, year=2018, chapters=20 available=false] MovieDVD [ id=126, title=Forever Young, year=2020, size=140.0MB available=false]

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 3

Enter the id: 999

The media object id=999 is not found

Welcome to Media Rental System 1: Load Media objects...

2: Find Media object... 3: Rent Media object... 9: Quit

Enter your selection : 9

Thank you for using the program. Goodbye!

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application, Word

Description automatically generatedGraphical user interface, text, application

Description automatically generated**Sample User interface - GUI:**

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, text, application, Word

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedAbove closed the window by pressing the X, results in below:

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated