**BÁO CÁO THỰC HÀNH LAB 4   
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG - Object oriented programming**

**Họ và tên: Trần Đại Hiệp**

**MSSV: 20226081**

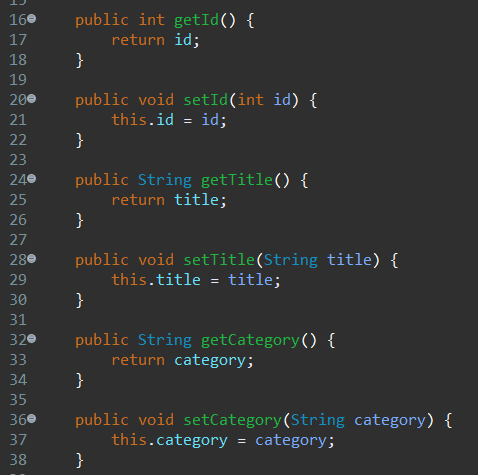
*3. Creating the Book class*

*3.1 Creating the Book class*

A screen shot of a computer program

Description automatically generated

*3.2 Getters and Setters*



A computer code with colorful text

Description automatically generated

*3.3 Next, create addAuthor(String authorName) and removeAuthor(String authorName) for the Book class*

A screen shot of a computer program

Description automatically generated

*4. Create abstract class named “Media”*

*4.1 Add fields to the Media class*

A screen shot of a computer program

Description automatically generated

*4.2 Public accessor method*

A screen shot of a computer program

Description automatically generated

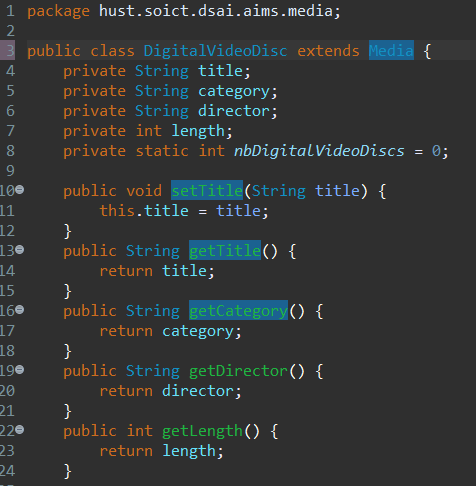
A computer screen shot of a program code

Description automatically generated

*4.3 Remove fields and methods from Book and DigitalVideoDisc classes. Extend the Media class for both Book and DigitalVideoDisc.*

A screen shot of a computer program

Description automatically generated



A screen shot of a computer program

Description automatically generated

*5. Create the CompactDisc class*

*5.1. Create the Disc class extending the Media class*

A screen shot of a computer program

Description automatically generated

*5.1.1. Create the CompactDisc extending the Disc class. Save your changes.*

A screen shot of a computer program

Description automatically generated

*5.2. Create the Track class which models a track on a compact disc and will store information including the title and length of the track*

A screen shot of a computer program

Description automatically generated

*5.3. Modify the CompactDisc class*



A screen shot of a computer code

Description automatically generated

*6. Create the Playable Interface*

*6.1 Create the Playable Interface*

A screen shot of a computer code

Description automatically generated

*6.2 Implement the Playable with CompactDisc, DigitalVideoDisc and Track*

A screen shot of a computer

Description automatically generated

A computer screen shot of text

Description automatically generated

A screen shot of a computer code

Description automatically generated

*6.3 Implement play() for DigitalVideoDisc*

A screen shot of a computer

Description automatically generated

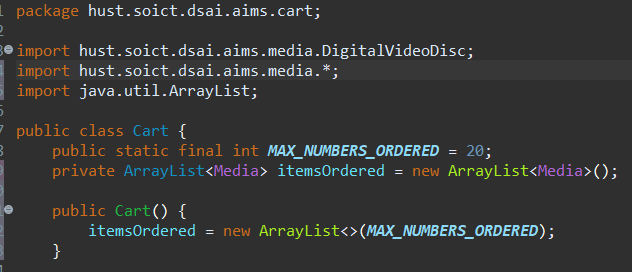
*6.4 Implement play() for Track*

A screen shot of a computer

Description automatically generated

*7. Update the Cart class to work with Media*

*7.1 Recreate the itemsOrdered field, this time as an object ArrayList instead of an array.*



*7.2 Create addMedia() and removeMedia() to replace addDigitalVideoDisc() and removeDigitalVideoDisc()*

A screen shot of a computer program

Description automatically generated

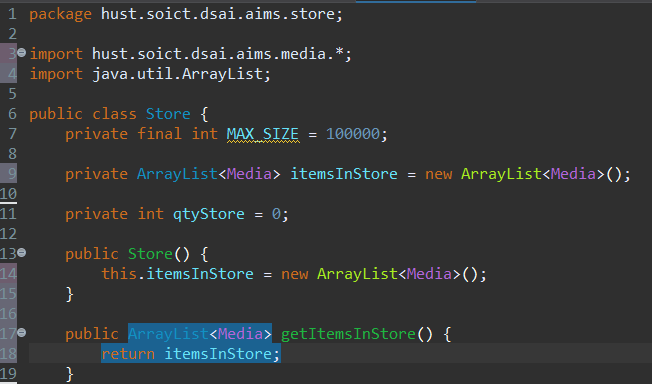
*7.3 Update the totalCost() method*

A screen shot of a computer code

Description automatically generated

*8. Update the Store class to work with Media*

*8.1 Similar to the Cart class, change the itemsInStore[] attribute of the Store class to ArrayList<Media> type.*



*8.2 Replace the addDigitalVideoDisc() and removeDigitalVideoDisc() methods with addMedia() and removeMedia()*

A screen shot of a computer code

Description automatically generated

*9. Constructors of whole classes and parent classes*

*9.1 Update the UML class diagram for the AimsProject.*

A yellow computer screen with text

Description automatically generated with medium confidence

*9.2 Which classes are aggregates of other classes?*

* Answer: Both Cart and Store are the aggregate of Media, while CompactDisc is the aggregate of Track.

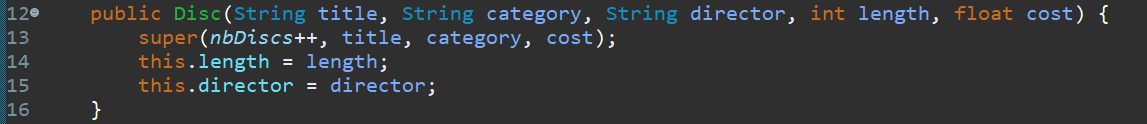
*9.3 Write constructors for parent and child classes. Remove redundant setter methods if any*

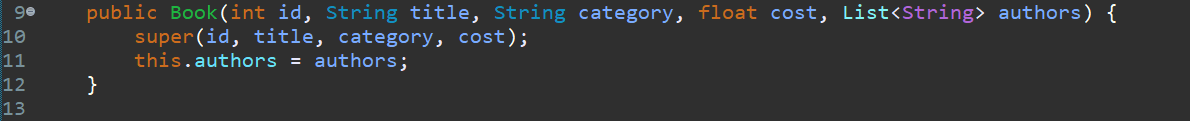
*9.3.1 Media constructor*

A screenshot of a computer program

Description automatically generated

*9.3.2 Disc constructor (Child class of Media)*

*9.3.3 Book constructor (Child class of Media)*



*9.3.4 CompactDisc constructor (Child class of Disc)*

A screen shot of a computer program

Description automatically generated

*9.3.5 DigitalVideoDisc constructor (Child class of Media)*

A screen shot of a computer program

Description automatically generated

*10. Unique item in a list*

*10.1 Override equals() method of class Media*

A screen shot of a computer code

Description automatically generated

*10.2 Override equals() method of class Track*

A screen shot of a computer code

Description automatically generated

*10.3 If the passing object is not an instance of Media, what happens?*

* Answer: If the object passed in the method is not an instance of Media, the equals() method returns false.

*11. Polymorphism with toString() method*

*11.1 Override toString() in Media class*

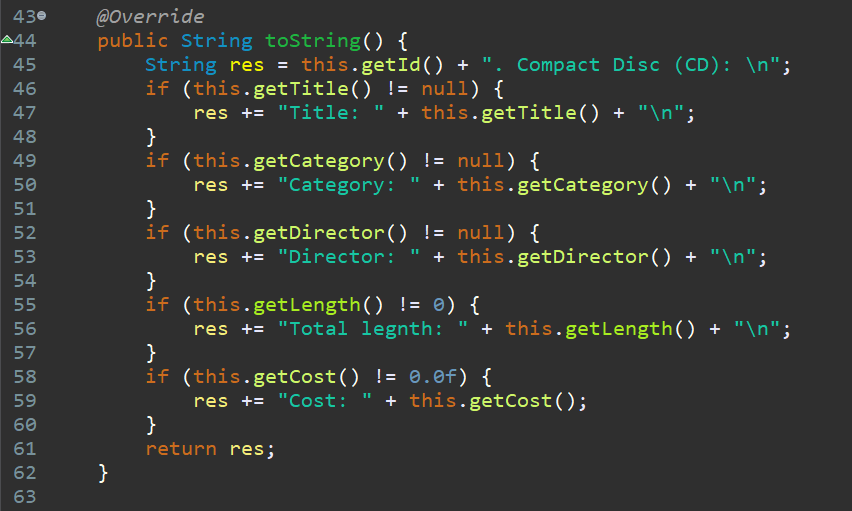
A screen shot of a computer code

Description automatically generated

*11.2 Override toString() in Book class*



*11.3 Override toString() in CompactDisc class*



*11.4 Override toString() in DigitalVideoDisc class*



*11.5 Iterate through the list and print out the information of the media by using toString() method. Observe what happens and explain in detail.*

* Answer: The polymorphism sample code program will run normally, each object (CD, DVD, BOOK) will be printed by its toString() method in their separate objects. The reason is that because Book, CompactDisc and DigitalVideoDisc inherit Media, so objects book, cd, dvd are also instances of Media.

*12. Sort media in the cart*

*12.1 Create class MediaComparatorByTitleCost*

A computer screen shot of a program code

Description automatically generated

*12.2 Create class MediaComparatorByCostTitle*



*12.3 What class should implement the Comparable interface?*

* Answer: The Media Class should implement the Comparable interface.

*12.4 In those classes, how should you implement the compareTo()method to reflect the ordering that we want?*

* Answer: The compareTo() method needs to return a negative integer, zero, or a positive integer if the current object is respectively "smaller", equal to, or "greater" than the specified object.

*12.5 Can we have two ordering rules of the item (by title then cost and by cost then title) if we use this Comparable interface approach?*

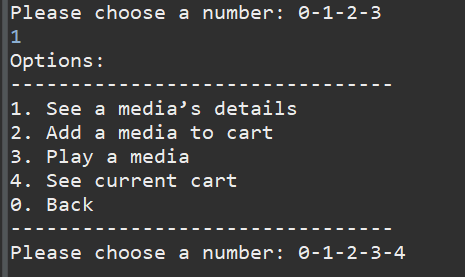
* Answer: No since Comparable interface is used for internal comparison and can be implemented only once by class Media.

*12.6 Suppose the DVDs has a different ordering rule from the other media types, that is by title, then decreasing length, then cost. How would you modify your code to allow this?*

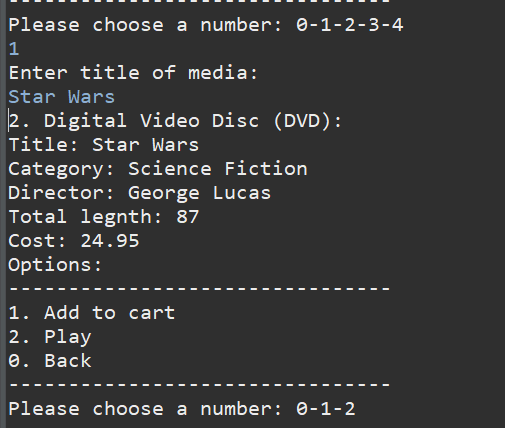
* Answer: I would modify my code to perform that required task by overriding compareTo() method in DigitalVideoDisc class.

*13.Create a complete console application in the Aims class*

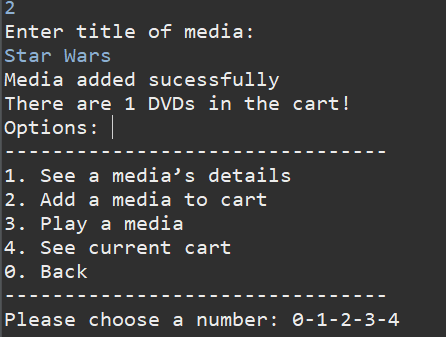
*13.1 From the main menu, if the user chooses option “View store”, the application will display all the items in the store, and a menu as following*



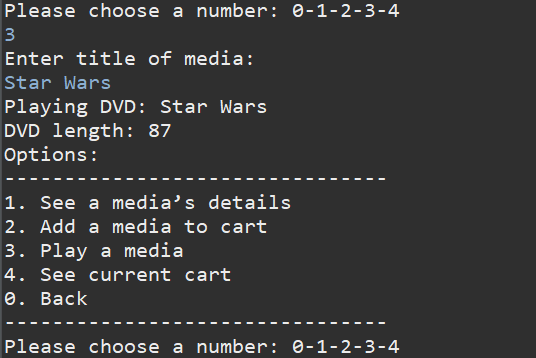
*13.1.1 The option “See a media’s details” will ask the user to enter the title of the media and display the information of that media. Please remember to check the validity of the title. Under the information display, the system also shows the following menu (note that the “Play” option is only available to CD and DVD type.*

**

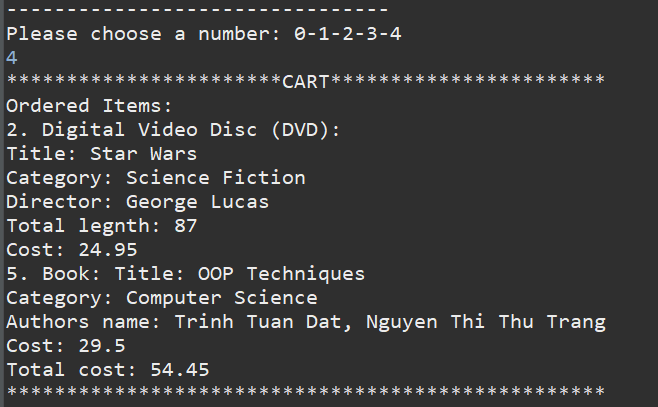
*13.1.2 The option “Add a media to cart” will ask the user to enter the title of the media that he/she sees on the screen (the list of media in store), then add the media to cart. Please remember to check the validity of the title. After adding a DVD to the cart, the system will display the number of DVDs in the current cart.*



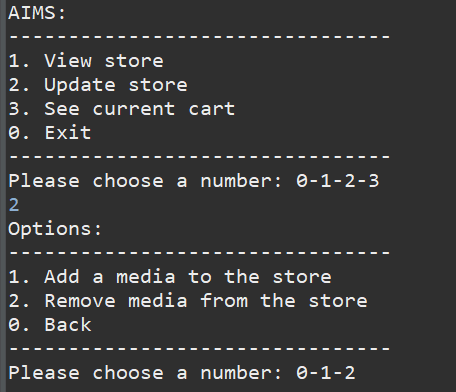
*13.1.3 The option “Play a media” will ask the same input from the user as option 2. You should again check the validity of the title.*



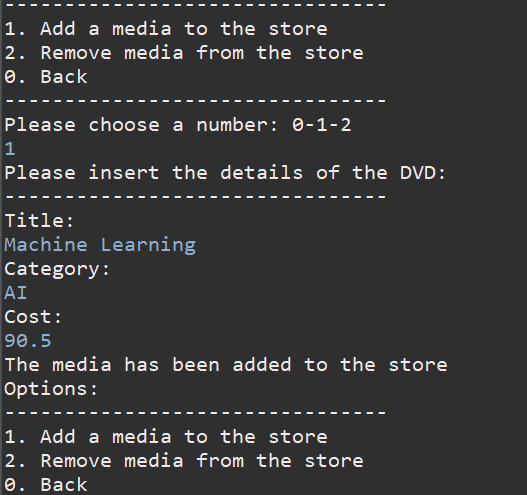
*13.1.4 The option “See current cart”.*



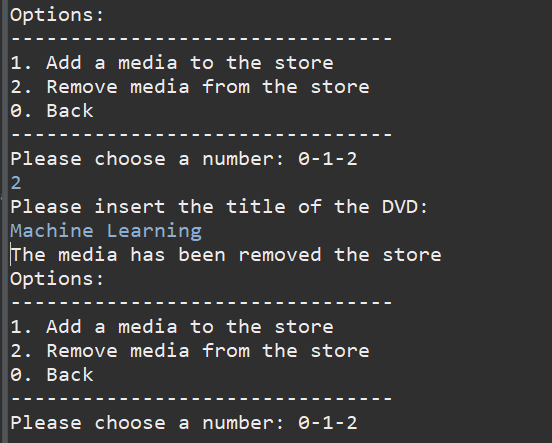
*13.2 From the main menu, if the user chooses option “Update store”, the application will allow the user to add a media to or remove a media from the store*



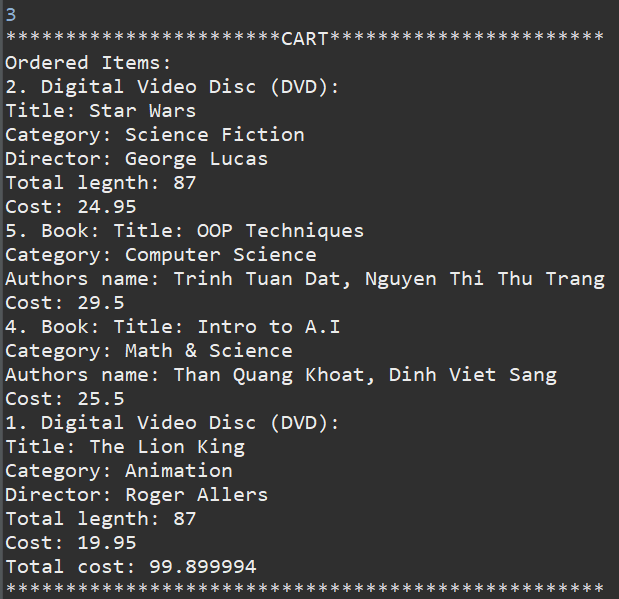
*13.2.1 Add a new media to the store*

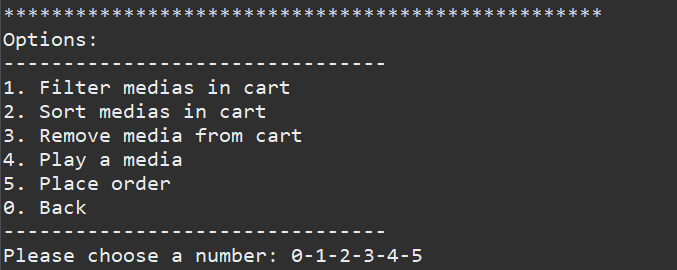


*13.2.2 Remove Media from the Store*



*13.3 From the main menu, if the user chooses option “See current cart”, the application will display the information of the current cart, and a menu as following:*





*13.3.1 The “Filter medias in cart” option should allow the user to choose one of two filtering options: by id and by title.*

A screenshot of a computer program

Description automatically generated

*13.3.1.1 Filter by id*

A screenshot of a computer program

Description automatically generated

*13.3.1.2 Filter by Title*

A screenshot of a computer program

Description automatically generated

*13.3.2 The “Sort medias in cart” option should allow the user to choose one of two sorting option: by title or by cost.*

A black screen with white text

Description automatically generated

*13.3.2.1 Sort by title*

A computer screen shot of a black screen

Description automatically generated

*13.3.2.1 Sort by cost*

A screen shot of a computer

Description automatically generated

*13.3.3 Place order*

A screenshot of a computer

Description automatically generated