Lab04: Inheritance and Polymorphism

Section 1: Import the existing project into the workspace of Eclipse

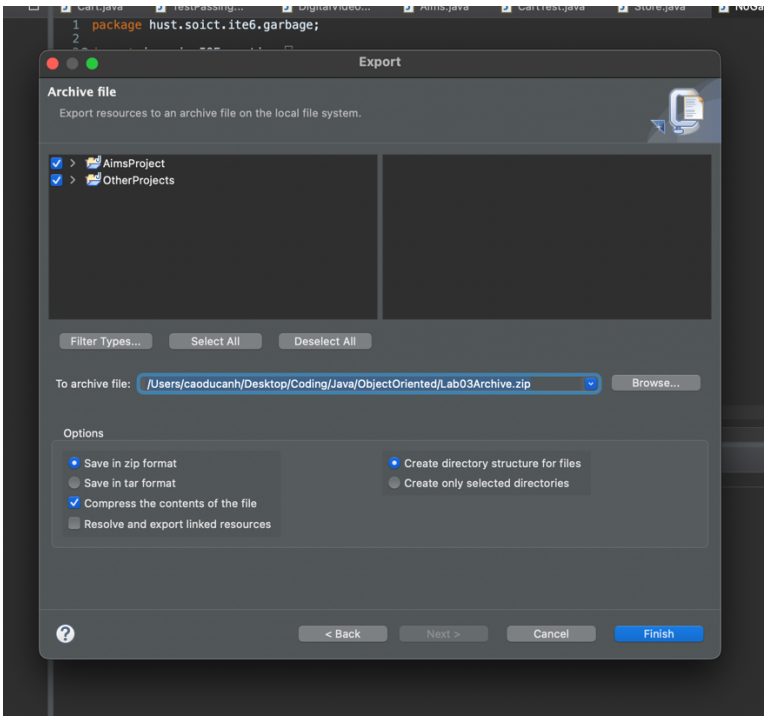


Figure 0-1 Archive the previous lab's projects to a compressed file

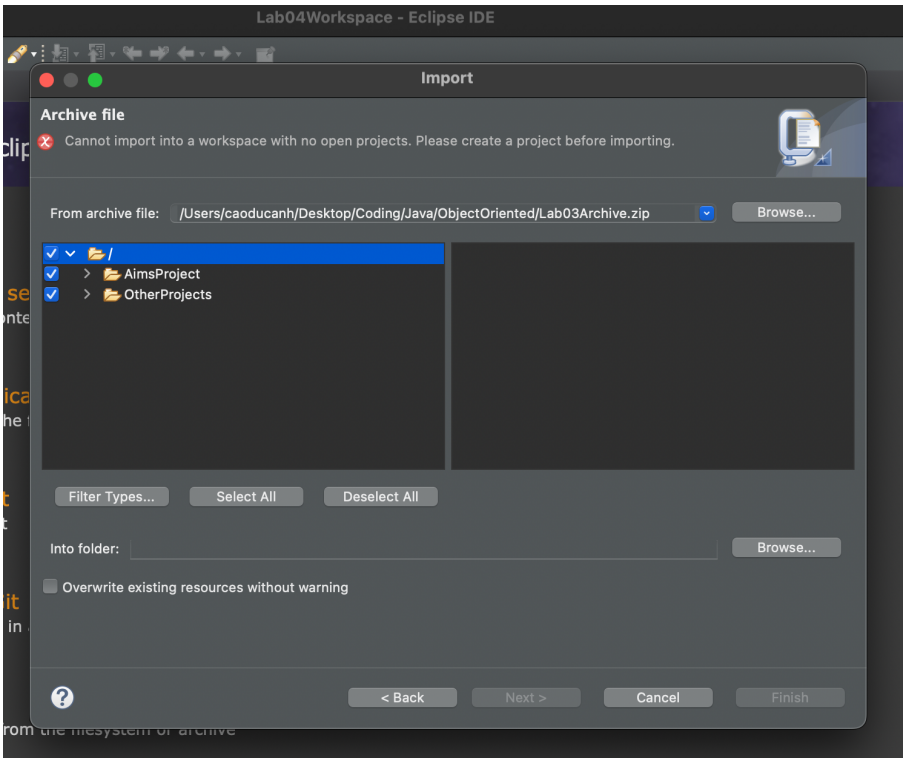


Figure 0-2 Import the archive to the new workspace

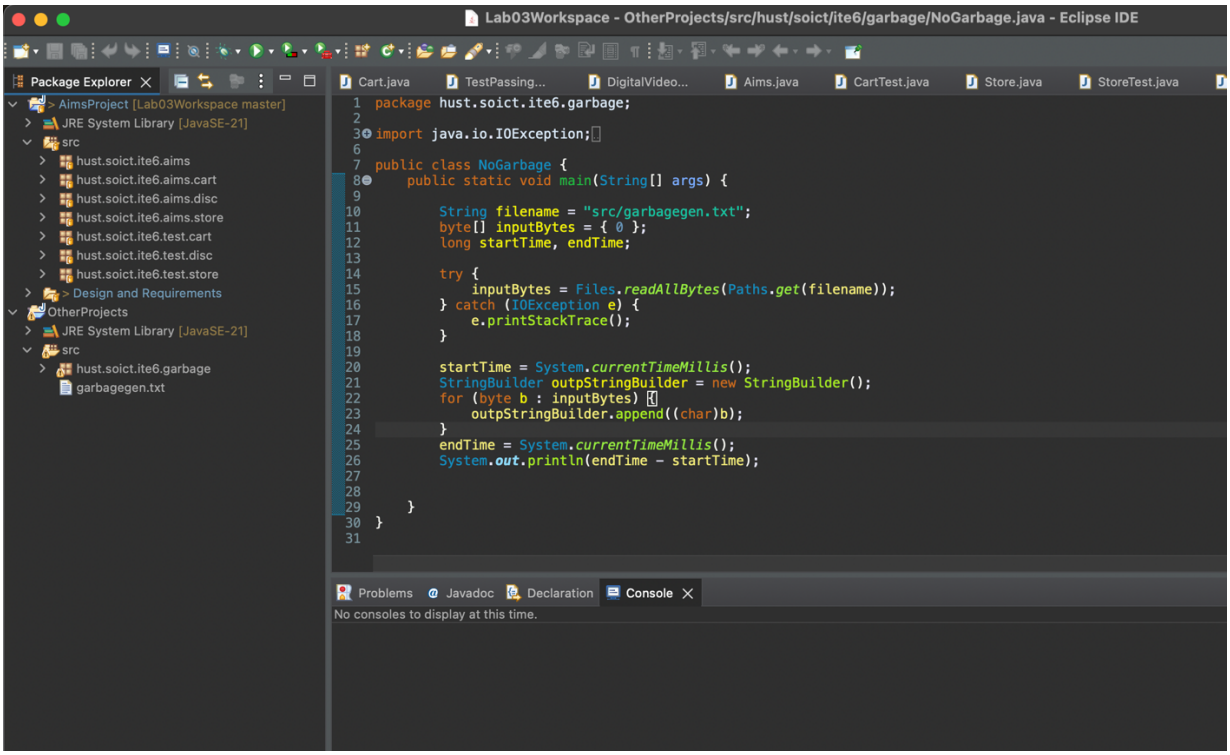


Figure 0-3 Imported project

Section 2: Additional requirements of AIMS

No tasks were given in this section

Section 3: Creating the Book class

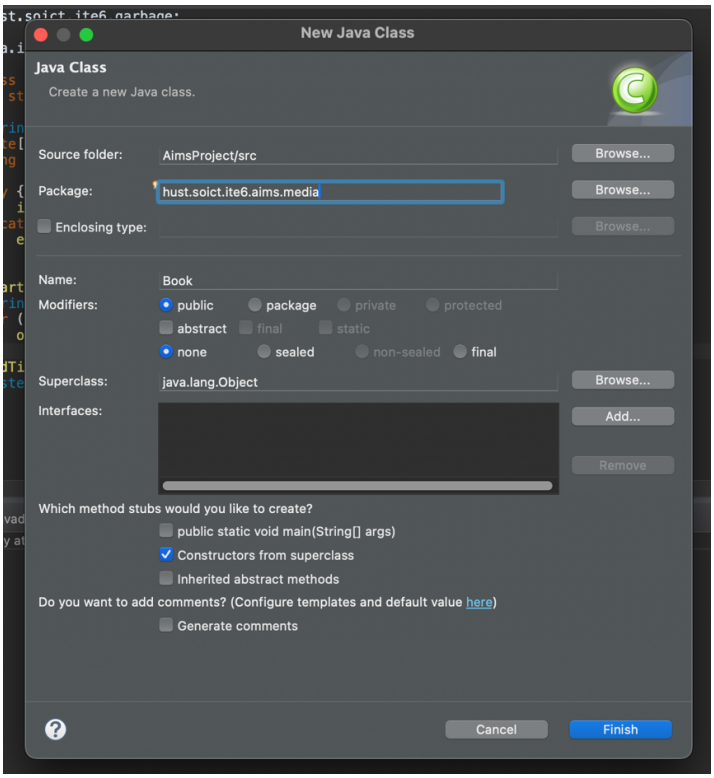


Figure 0-4 Creating the new 'Book' class

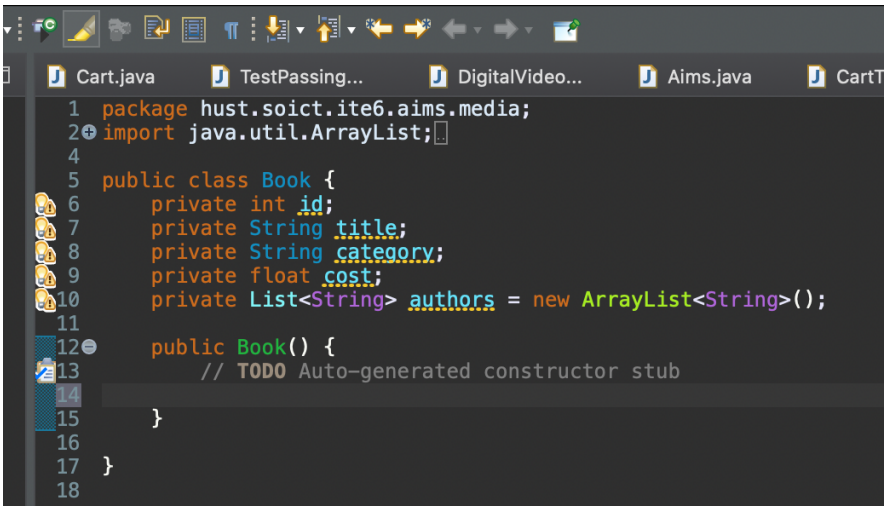


Figure 0-5 Creating private attributes for 'Book' class

Figure 0-6 Generate public accessors for private fields

```java
package hust.soict.ite6.aims.media;
import java.util.ArrayList;

public class Book {
    private int id;
    private String title;
    private String category;
    private float cost;
    private List<String> authors = new ArrayList<String>();

    public void addAuthor(String authorName) {
        if (!authors.contains(authorName)) {
            authors.add(authorName);
        } else {
            System.out.println("This author has already been added to the list!");
        }
    }

    public void removeAuthor(String authorName) {
        if (authors.contains(authorName)) {
            authors.remove(authorName);
        } else {
            System.out.println("Given author does not exist!!");
        }
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getCategory() {
        return category;
    }
}
```

Figure 0-7 Methods to add and remove authors from list

Section 4: Creating the abstract 'Media' class

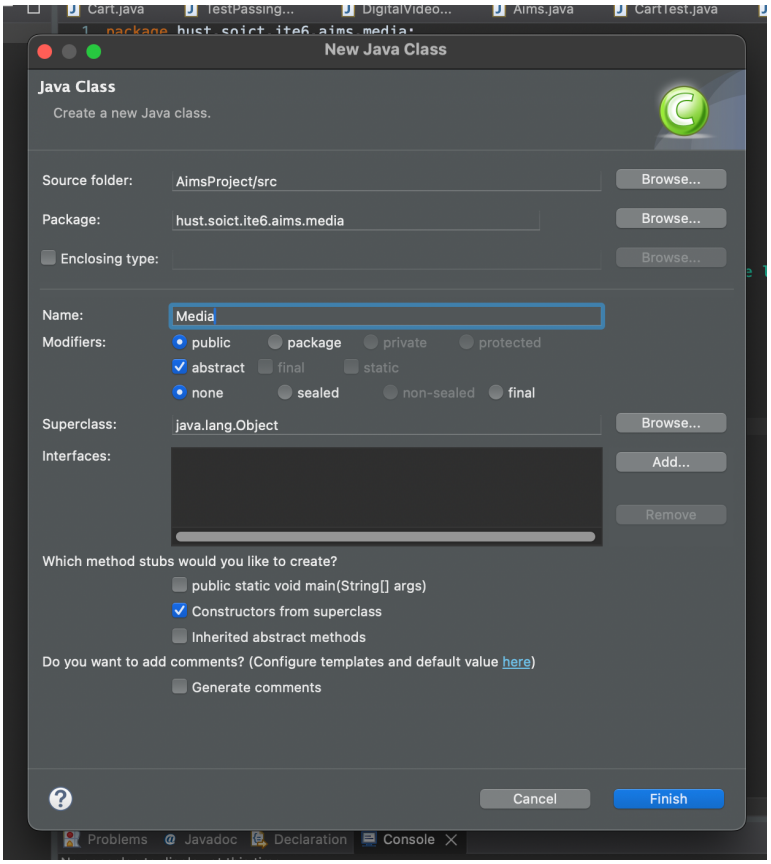


Figure 0-8 Creating new abstract class 'Media'

Cart.java DigitalVideo... Book.java Media.java

```java
package hust.soict.ite6.aims.media;

public abstract class Media {

    private int id;
    private String title;
    private String category;
    private float cost;

    public Media() {
        // TODO Auto-generated constructor stub
    }

    public int getId() {
        return id;
    }
    public String getTitle() {
        return title;
    }
    public String getCategory() {
        return category;
    }
    public float getCost() {
        return cost;
    }

}
```

Figure 0-9 Creating private attributes and generating public accessors

Cart.java DigitalVideo... Book.java Media.java

```java
package hust.soict.ite6.aims.media;
import java.util.ArrayList;

public class Book {

    private List<String> authors = new ArrayList<String>();

    public void addAuthor(String authorName) {
        if (!authors.contains(authorName)) {
            authors.add(authorName);
        } else {
            System.out.println("This author has already been added
        }
    }

    public void removeAuthor(String authorName) {
        if (authors.contains(authorName)) {
            authors.remove(authorName);
        } else {
            System.out.println("Given author does not exist!!");
        }
    }
```

Figure 0-10 Deleting mutual attributes and their accessors in the 'Book' class

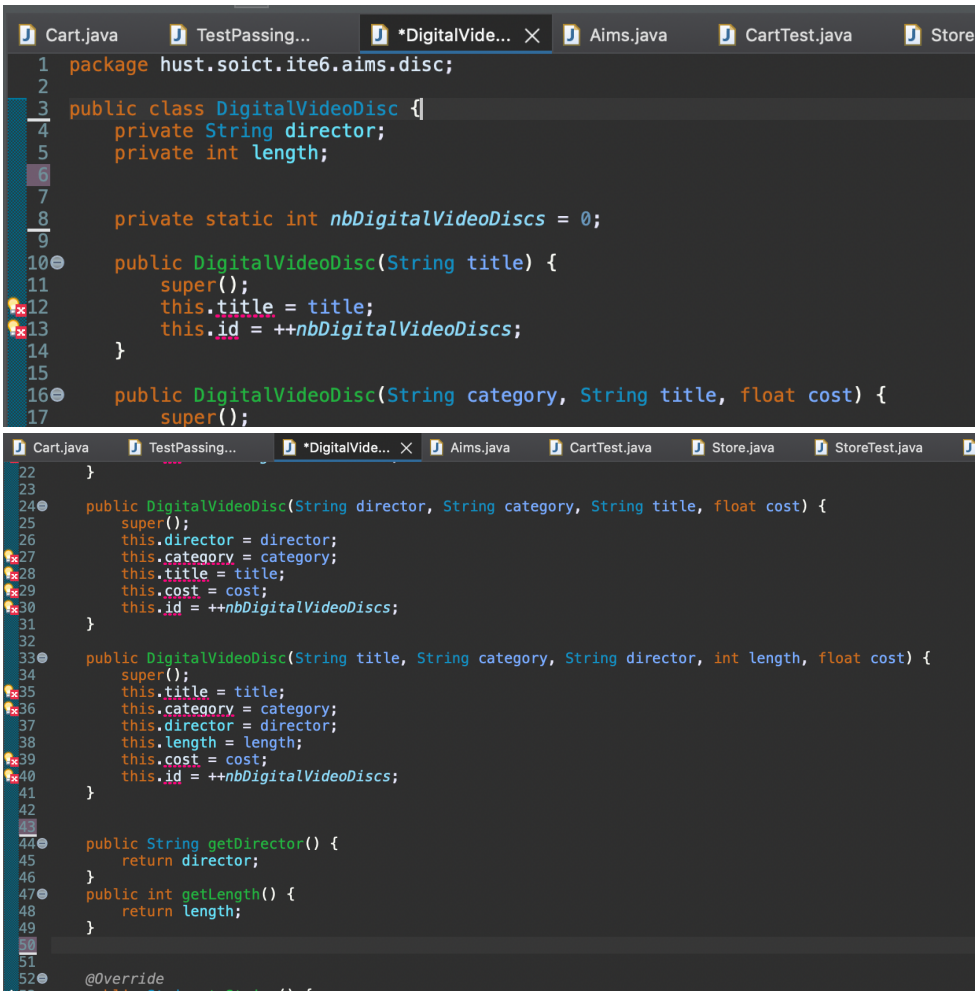


Figure 0-11 Similarly, delete mutual attributes and their public accessors in 'DigitalVideoDisc' class

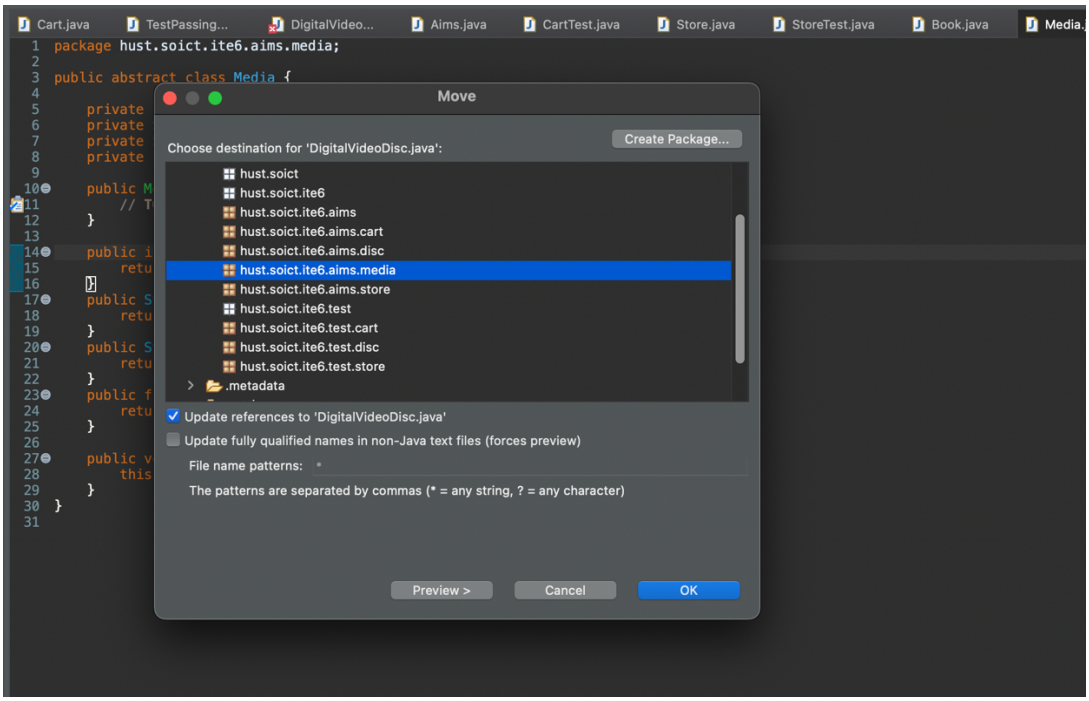


Figure 0-12 Move 'DigitalVideoDisc' to 'hust.soict.ite6.aims.media'

Cart.java TestPassing... DigitalVideo... Aims.java CartTest.java Store.ja

```java
package hust.soict.ite6.aims.media;
import java.util.ArrayList;

public class Book extends Media {

    private List<String> authors = new ArrayList<String>();

    public void addAuthor(String authorName) {
        if (!authors.contains(authorName)) {
            authors.add(authorName);
        } else {
            System.out.println("This author has already been added to the list!");
        }
    }
```

Cart.java TestPassing... DigitalVideo... Aims.java

```java
package hust.soict.ite6.aims.media;

public class DigitalVideoDisc extends Media {
    private String director;
    private int length;


    private static int nbDigitalVideoDiscs = 0;

    public DigitalVideoDisc(String title) {
        super();
        this.title = title;
        this.id = ++nbDigitalVideoDiscs;
```

Figure 0-13 Extends 'Book' and 'DigitalVideoDisc'

Section 5: Creating the ‘CompactDisc’ class

Cart.java TestPassing... DigitalVideo... CompactDisc.... Aims.java

```java
package hust.soict.ite6.aims.media;

public class CompactDisc extends Media {

    public CompactDisc() {
        // TODO Auto-generated constructor stub
    }

}
```

Figure 0-14 Creating the 'CompactDisc' class which extends 'Media'

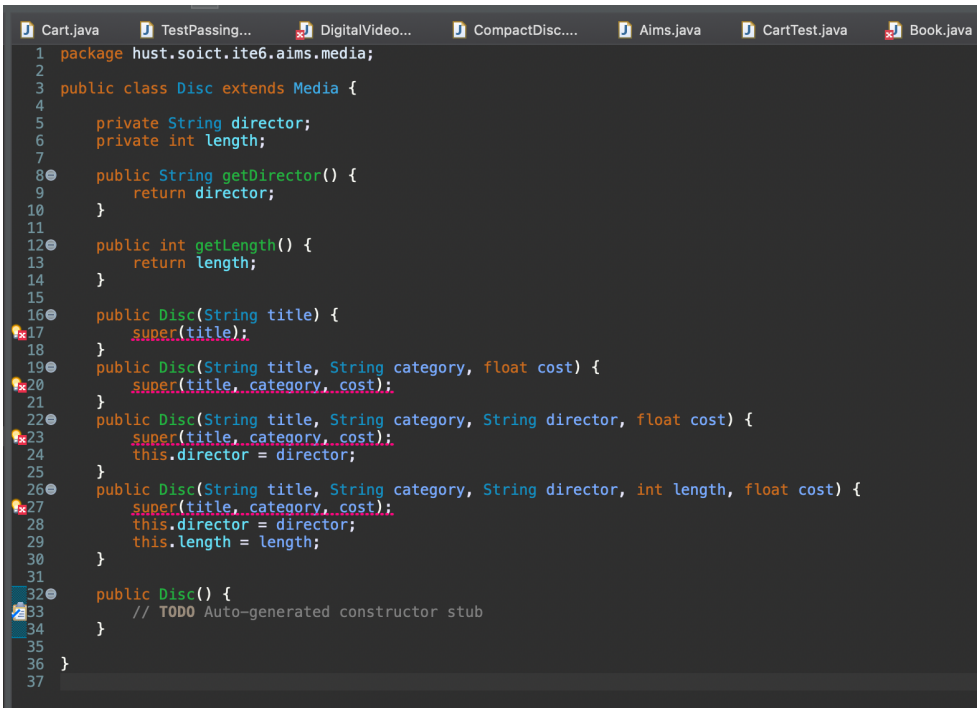


Figure 0-15 Creating 'Disc' class and declare attributes and generate getters and constructors

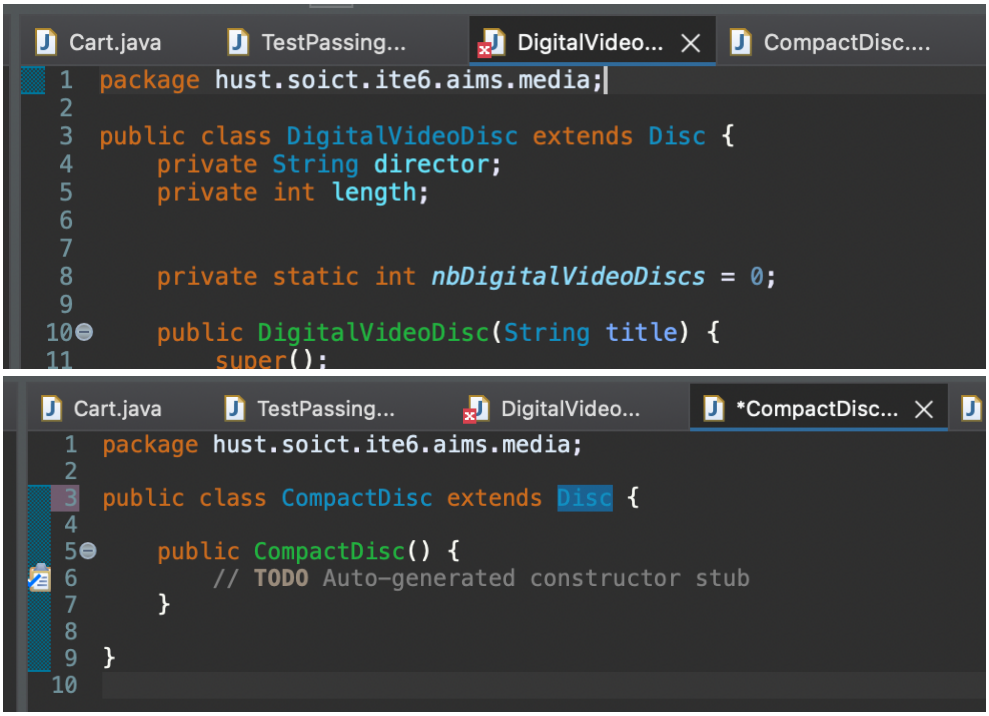


Figure 0-16 Modify 'DigitalVideoDisc' and 'CompactDisc' to extend 'Disc'

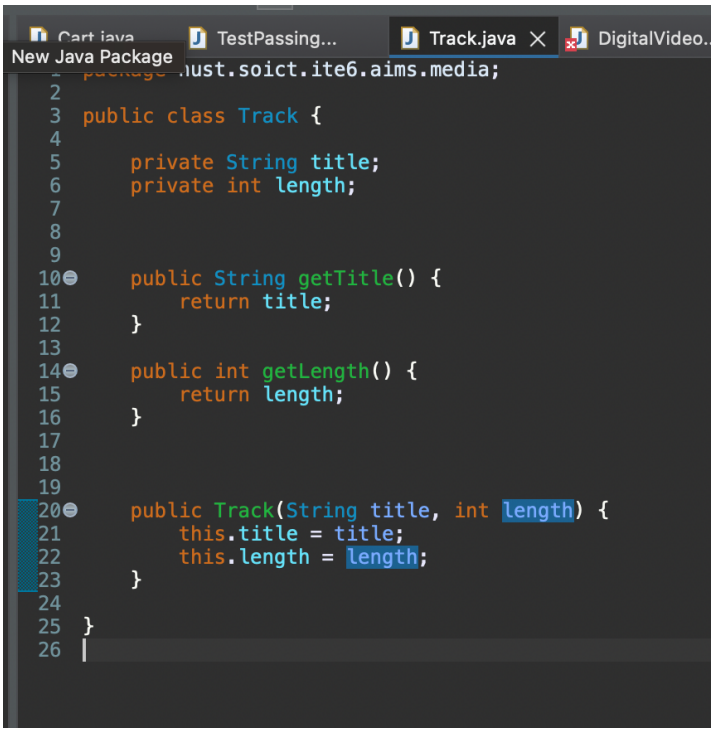


Figure 0-17 Create new 'Track' class, declare it's attributes and generate getters, constructors

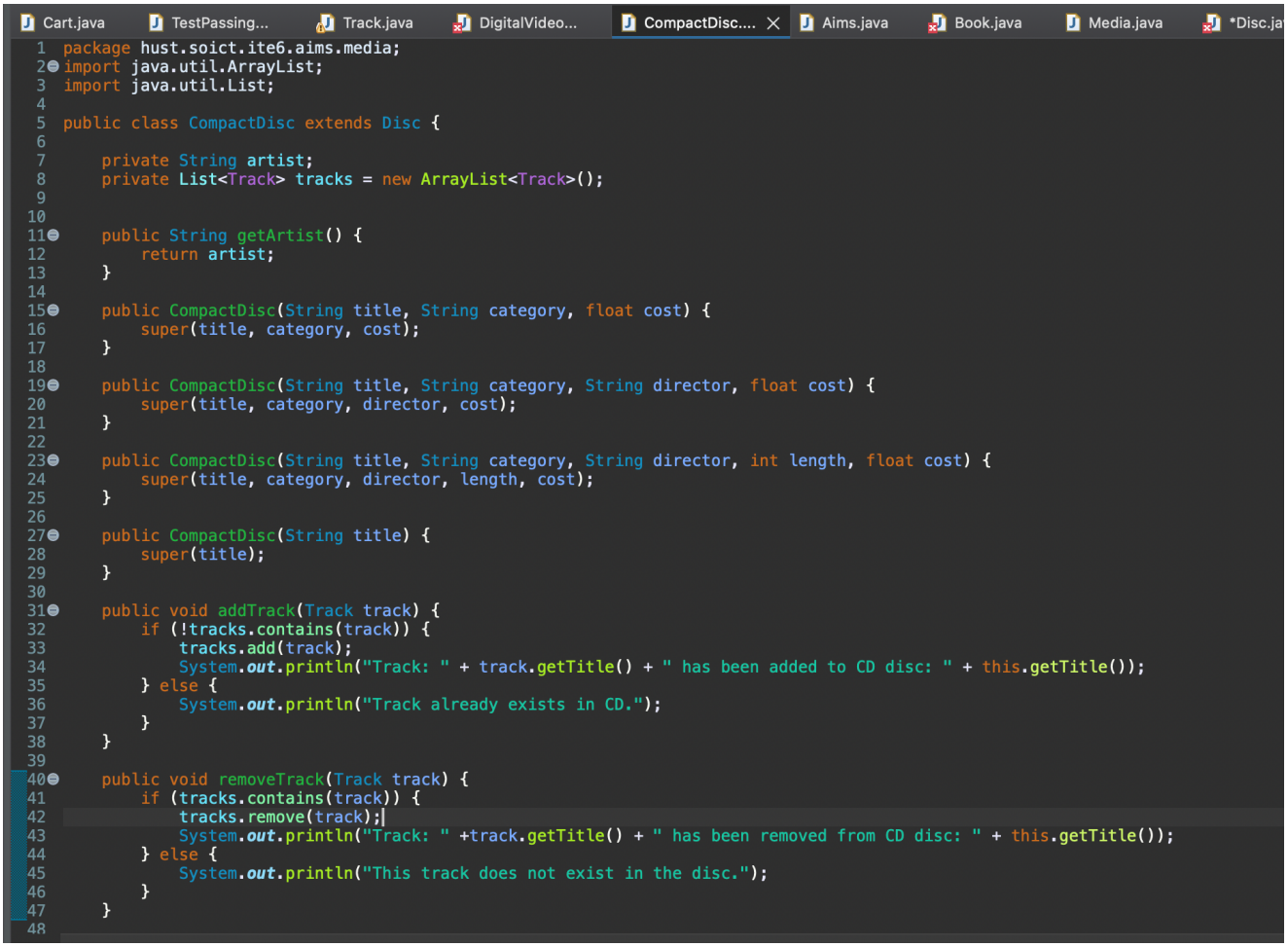


Figure 0-18 Adding attributes, making public getters, constructors and creating 'addTrack', 'removeTrack' methods for 'CompactDisc'

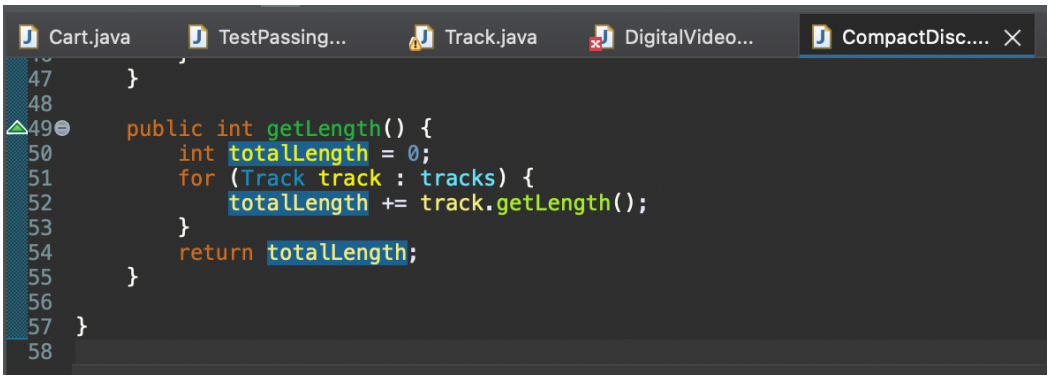


Figure 0-19 Write 'getLength' method for 'CompactDisc'

Section 6: Create the 'Playable' interface

Figure 0-20 Create 'Playable' interface and add method prototype

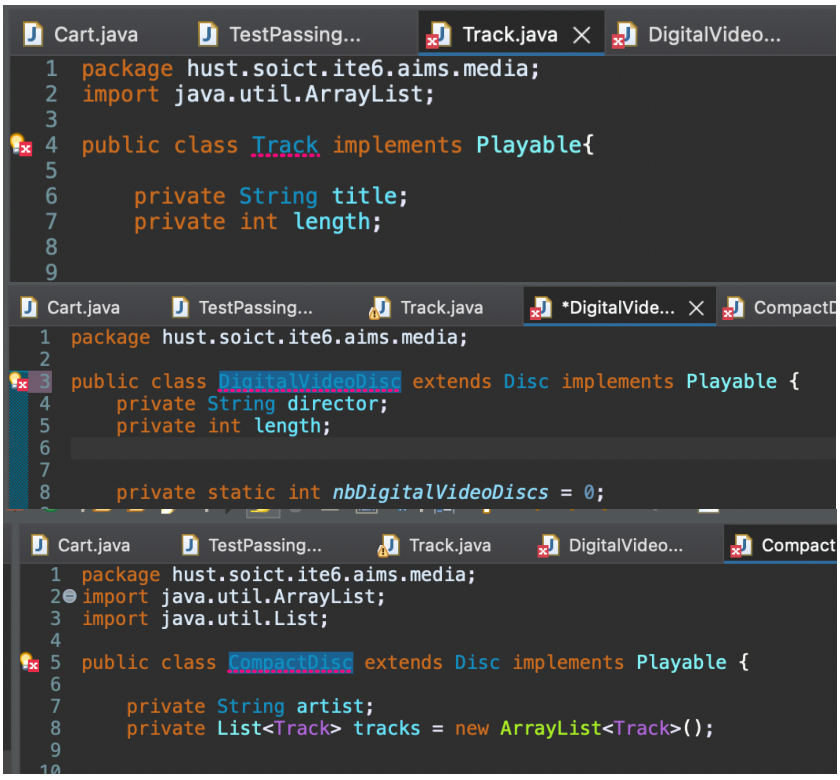


Figure 0-21 Implements 'Playable' with 'CompactDisc', 'DigitalVideoDisc' and 'Track'

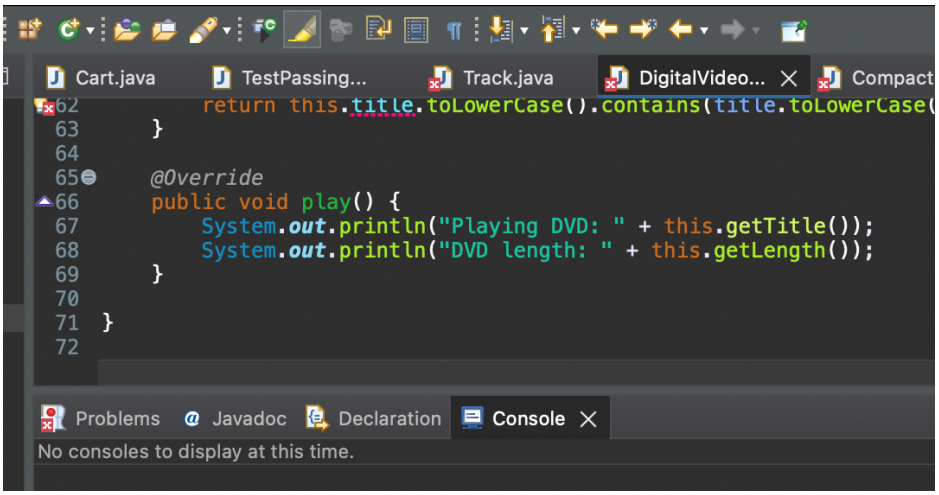


Figure 0-22 Add new method 'play()' to 'DigitalVideoDisc' and 'Track'

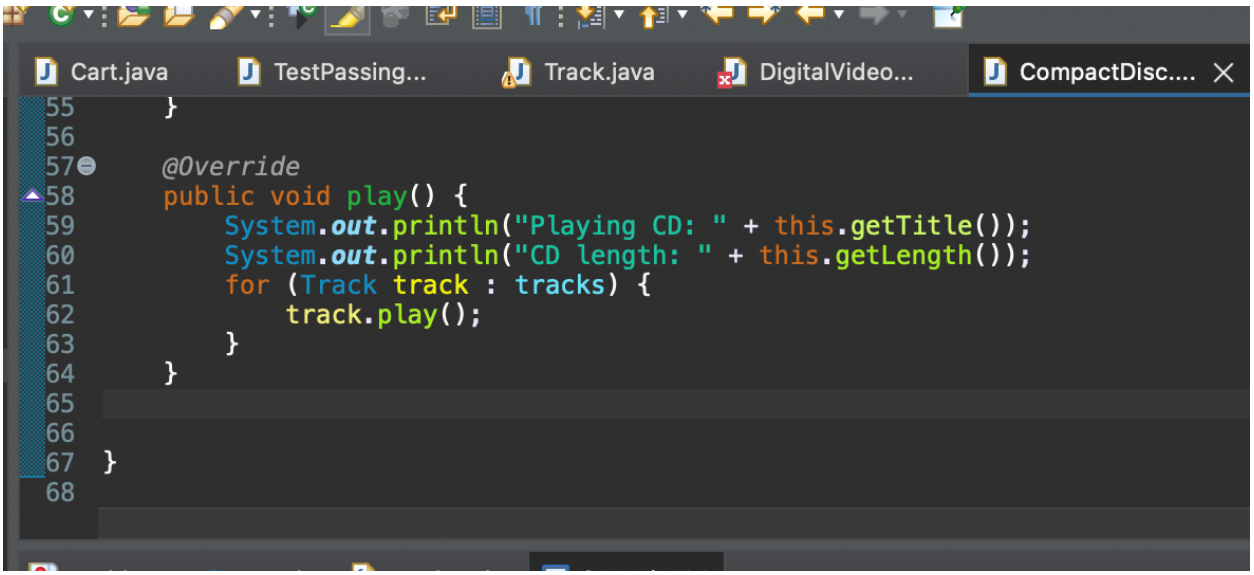


Figure 0-23 'play()' method for 'CompactDisc' which loops through every tracks of a disc

Section 7: Update the 'Cart' class to work with 'Media'

*Cart.java | TestPassing... | Track.java | DigitalVideo... | CompactDisc.... | Book.java | Media.java

```java
package hust.soict.ite6.aims.cart;
import java.util.ArrayList;
import hust.soict.ite6.aims.media.DigitalVideoDisc;
import hust.soict.ite6.aims.media.Media;

public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private ArrayList<Media> itemsOrdered = new ArrayList<Media>();

    public int qtyOrdered = 0;

    public void addMedia(Media media) {
        if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
            System.out.println("The cart is full!");
        } else if (itemsOrdered.contains(media)) {
            System.out.println(media.getTitle() + " was already added to cart!");
        } else {
            itemsOrdered.add(media);
            System.out.println(media.getTitle() + " has been added!");
            qtyOrdered +=1;
        }
    }
    public void removeMedia(Media media) {
        if (itemsOrdered.size() == 0) {
            System.out.println("Nothing to remove!");
        } else {
            if (itemsOrdered.remove(media)) {
                System.out.println(media.getTitle() + " has been removed from the cart.");
                qtyOrdered -=1;
            } else {
                System.out.println("Media not found in cart!");
            }
        }
    }
```

Figure 0-24 Removing the old 'addDigitalVideoDisc()', 'removeDigitalVideoDisc()' methods and the 'itemsOrdered' array to work with the new 'Media' class

Cart.java | TestPassing... | Track.java | DigitalVideo... | CompactDisc.... | Book.java | Media.java

```java
    }

    public float totalCost() {
        float totalCost = 0;
        for (Media media : itemsOrdered) {
            totalCost += media.getCost();
        }
        return totalCost;
    }

    public void searchByID(int id) {
        boolean found = false;
        for (Media media : itemsOrdered) {
            if (media.getId() == id) {
                System.out.println("Found " + media);
                found = true;
            }
        }
        if (!found) {
            System.out.println("Sorry, no media were found that match the ID provided!");
        }
    }

    public void searchByTitle(String keyword) {
        boolean matchFound = false;
        for (Media media : itemsOrdered) {
            if (media.isMatch(keyword)) {
                System.out.println("Found " + media);
                matchFound = true;
            }
        }
        if (!matchFound) {
            System.out.println("Sorry, no media were found with \"" + keyword + "\" in the title!");
        }
    }

    public void printCart() {
        System.out.println("**********************CART**********************");
        System.out.println("Ordered Items:");
        for (Media media : itemsOrdered) {
            System.out.println(media);
        }
        System.out.println("Total cost: " + totalCost());
        System.out.println("************************************************");
    }

}
```

Figure 0-25 Update other methods to work with the new 'itemsOrdered' object

Section 8: Update the ‘Store’ class to work with ‘Media’

Cart.java | Track.java | DigitalVideo... | CompactDisc.... | Store.java | Book.java | Media.java

```java
package hust.soict.ite6.aims.store;
import java.util.ArrayList;
import java.util.List;
import hust.soict.ite6.aims.media.Media;


public class Store {
    private List<Media> itemsInStore = new ArrayList<Media>();

    public void addMedia(Media media) {
        if (itemsInStore.contains(media)) {
            System.out.println("The media " + media.getTitle() + " is already in the store!");
        } else {
            itemsInStore.add(media);
            System.out.println("The media " + media.getTitle() + " has been added to the store.");
        }
    }
    public void removeMedia(Media media) {
        if (itemsInStore.remove(media)) {
            System.out.println("The media " + media.getTitle() + " has been removed from the store.");
        } else {
            System.out.println("The media " + media.getTitle() + " is not in the store!");
        }
    }

    public void print() {
        if (itemsInStore.size() == 0) {
            System.out.println("The store is empty");
        } else {
            System.out.println("**********STORE INVENTORY**********");
            for (Media media : itemsInStore) {
                System.out.println(media);
            }
            System.out.println("***********************************");
        }
    }
}

```

Figure 0-26 Updated 'itemsInStore' object and all methods to use 'Media'

Section 9: Constructors of whole classes and parent classes

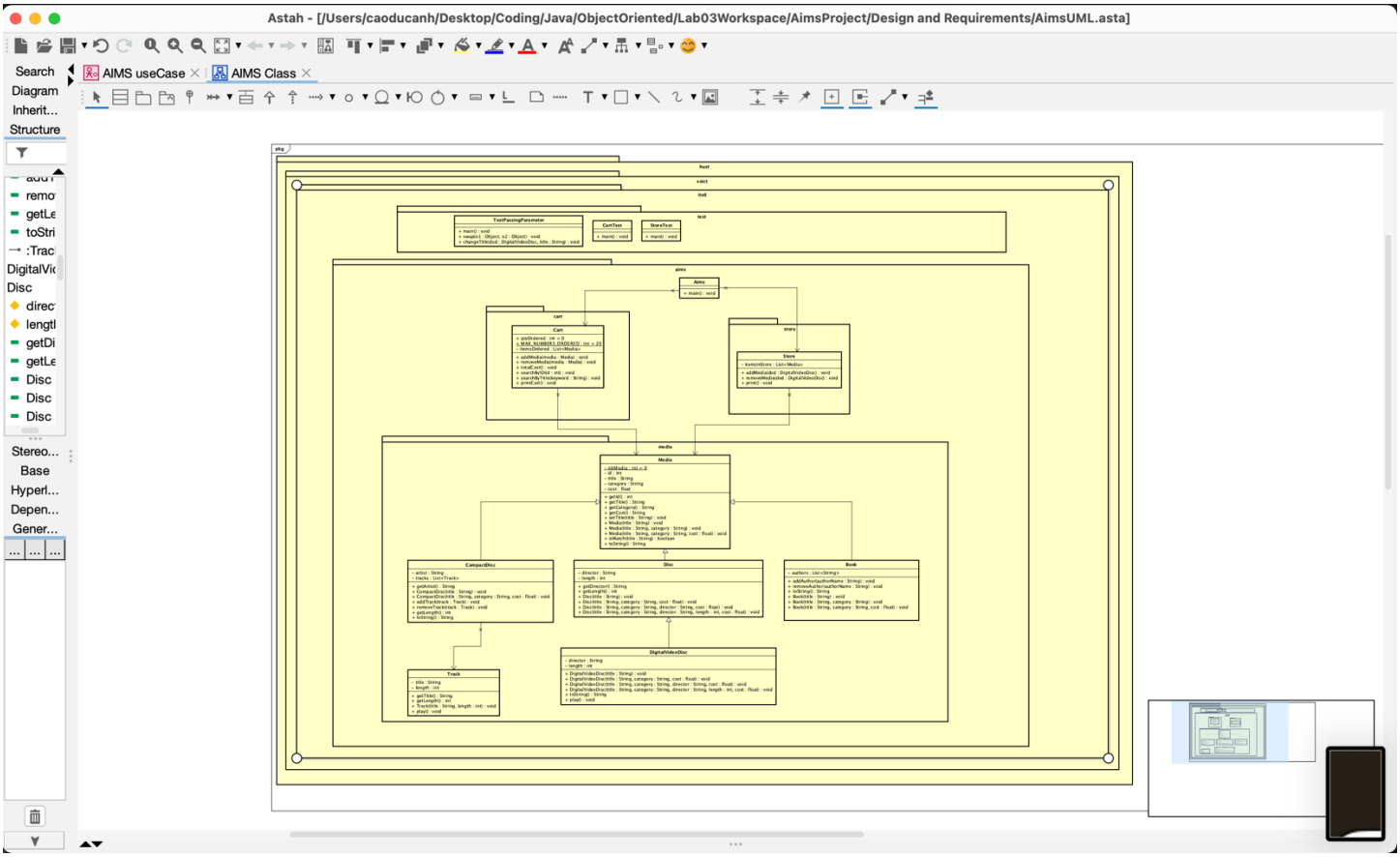


Figure 0-27 Update the project's UML class diagram

Cart.java | Track.java | DigitalVideo... | CompactDisc.... | Store.java | Book.java | Media.java | Disc.java | Playable.java

```
    private float cost;

    public int getId() {
        return id;
    }
    public String getTitle() {
        return title;
    }
    public String getCategory() {
        return category;
    }
    public float getCost() {
        return cost;
    }

    public void setTitle(String title) {
        this.title = title;
    }


    public Media(String title) {
        this.title = title;
    }
    public Media(String title, String category) {
        this.title = title;
        this.category = category;
    }
    public Media(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
    }
```

Figure 0-28 Rewrite constructors of parent class

Section 10: Unique item in a list

Cart.java | Track.java | DigitalVideo... | CompactDisc.... | Store.java | Book.java | Media.java | Disc.java

```java
    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }

    @Override
    public void play() {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (!(obj instanceof Track)) {
            return false;
        }
        return ((Track)obj).getTitle() == this.getTitle() && ((Track)obj).getLength() == this.getLength();
    }
}
```

Cart.java | Track.java | DigitalVideo... | CompactDisc.... | Store.java | Book.java | Media.java

```java
        return this.getTitle().toLowerCase().contains(title.toLowerCase());
    }

    @Override
    public String toString() {
        return "Media: " + this.getTitle() +
                " - Category: " + this.getCategory() +
                " - Cost: " + this.getCost() + "$";
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == this) {
            return true;
        }
        if (!(obj instanceof Media)) {
            return false;
        }
        return ((Media)obj).getTitle() == this.getTitle();
    }

}
```

Figure 0-29 Override 'equals()' method of 'Media' and 'Track'

Section 11: Polymorphism with toString() method

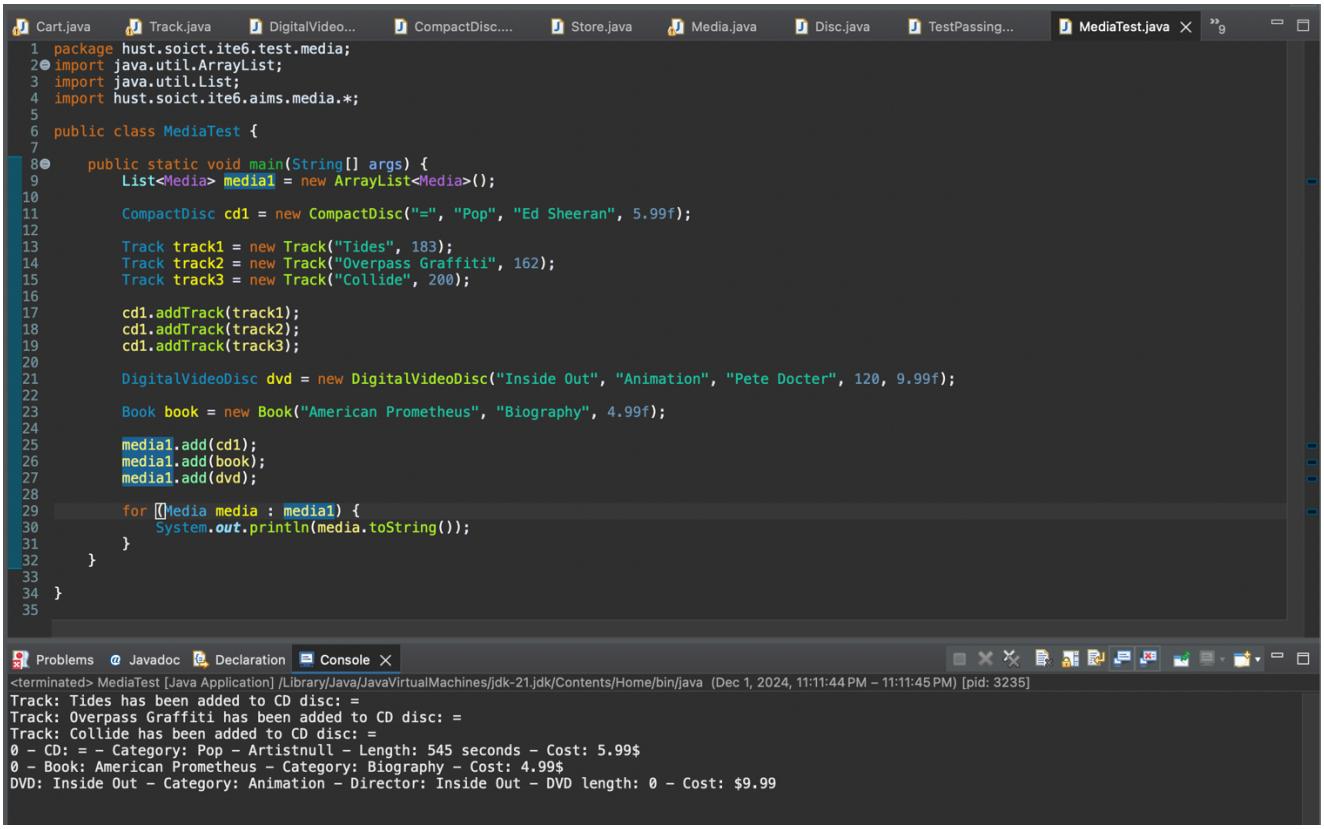


Figure 0-30 Create a new test class 'MediaTest' to test the toString() method

Section 12: Sort media in the cart

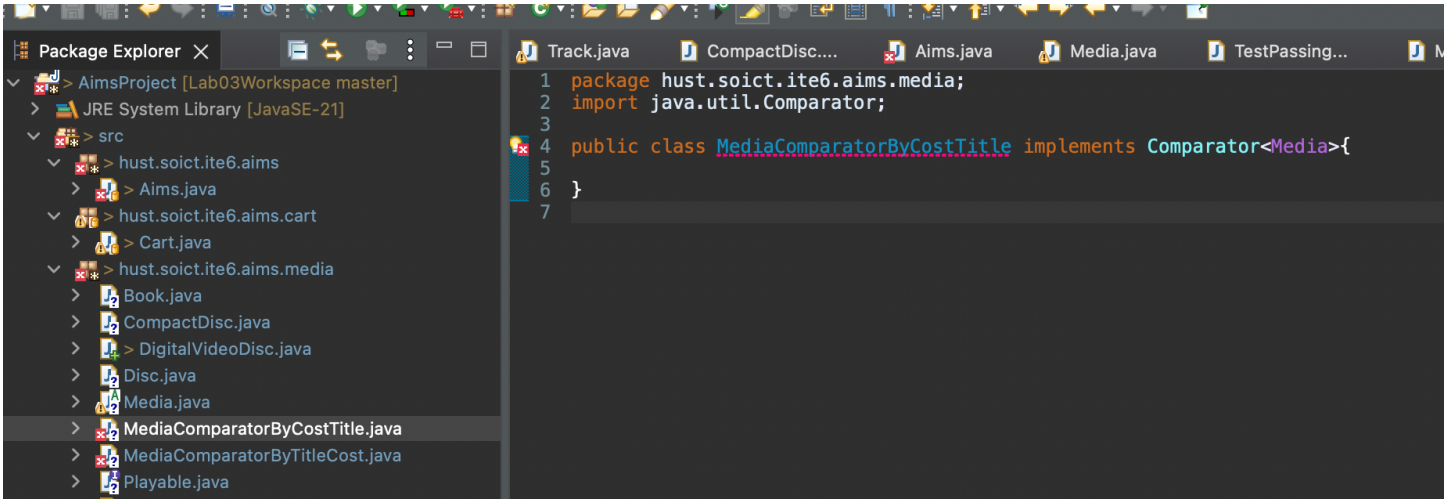


Figure 0-31 Create 2 classes of comparators

Track.java | CompactDisc.... | Aims.java | Media.java | TestPassing... | MediaTest.java | MediaCompar

```java
package hust.soict.ite6.aims.media;
import java.util.Comparator;

public abstract class Media  implements Comparator<Media>{

    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();
    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();

    private static int nbMedia = 0;
    private int id;
    private String title;
    private String category;
    private float cost;
```

Figure 0-32 Add comparators as attributes in 'Media' class

Questions:

- *What class should implement the Comparable interface?*

  The 'Media' class should implement the 'Comparable' interface.

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

  The implementation of the 'compareTo()' method can be found in the 'Media' class.

- 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?

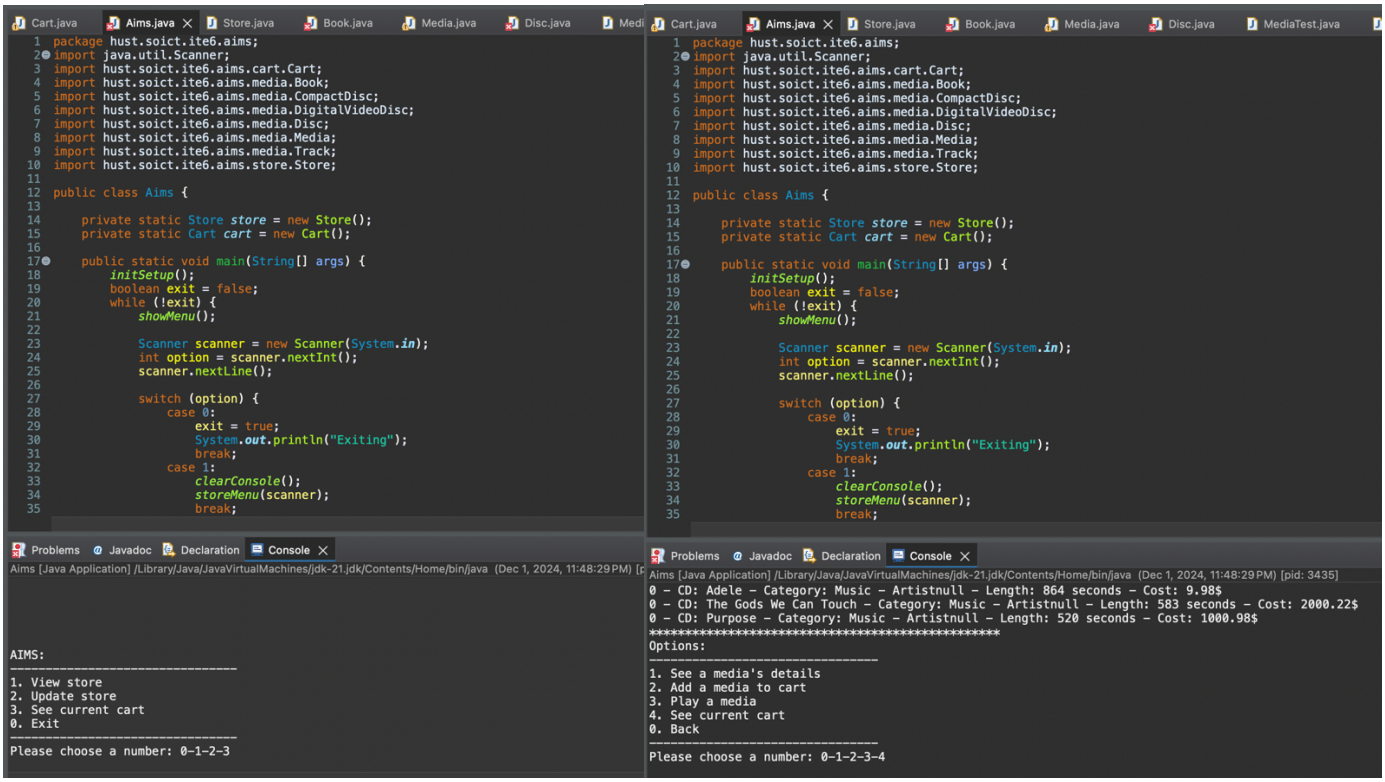
No, we cannot. The 'Comparable' interface supports only one natural ordering for the objects it compares.

- 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?

The compareTo() method in the Disc class can be overridden to implement the specific ordering rule.

Section 13: Create a complete console application in the Aims class

For full implementation please check the source code.



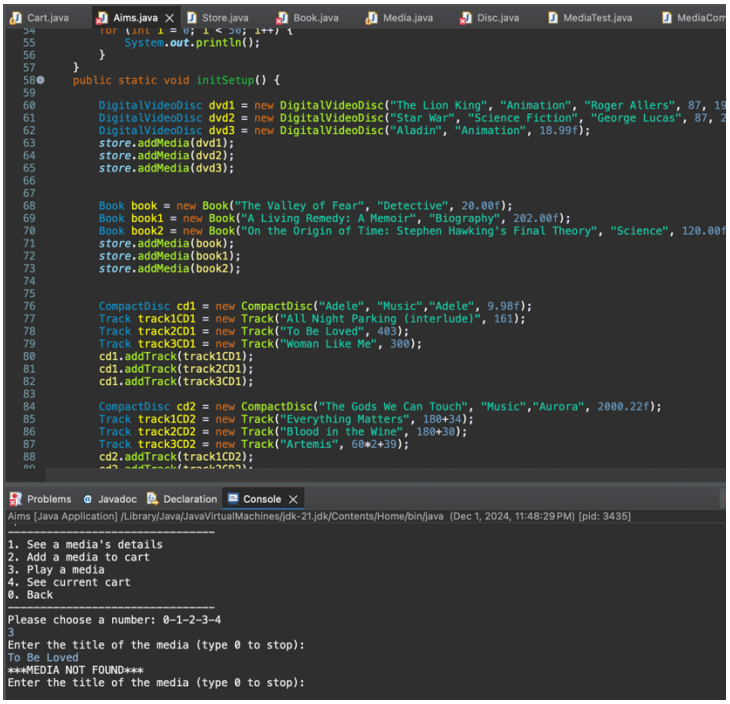
```
Cart.java  Aims.java  Store.java  Book.java  Media.java  Disc.java  Medi
1  package hust.soict.ite6.aims;
2  import java.util.Scanner;
3  import hust.soict.ite6.aims.cart.Cart;
4  import hust.soict.ite6.aims.media.Book;
5  import hust.soict.ite6.aims.media.CompactDisc;
6  import hust.soict.ite6.aims.media.DigitalVideoDisc;
7  import hust.soict.ite6.aims.media.Disc;
8  import hust.soict.ite6.aims.media.Media;
9  import hust.soict.ite6.aims.media.Track;
10 import hust.soict.ite6.aims.store.Store;
11
12 public class Aims {
13
14     private static Store store = new Store();
15     private static Cart cart = new Cart();
16
17     public static void main(String[] args) {
18         initSetup();
19         boolean exit = false;
20         while (!exit) {
21             showMenu();
22
23             Scanner scanner = new Scanner(System.in);
24             int option = scanner.nextInt();
25             scanner.nextLine();
26
27             switch (option) {
28                 case 0:
29                     exit = true;
30                     System.out.println("Exiting");
31                     break;
32                 case 1:
33                     clearConsole();
34                     storeMenu(scanner);
35                     break;

Problems  Javadoc  Declaration  Console
Aims [Java Application] /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java (Dec 1, 2024, 11:48:29 PM) [

AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3
```

```
Cart.java  Aims.java  Store.java  Book.java  Media.java  Disc.java  MediaTest.java
1  package hust.soict.ite6.aims;
2  import java.util.Scanner;
3  import hust.soict.ite6.aims.cart.Cart;
4  import hust.soict.ite6.aims.media.Book;
5  import hust.soict.ite6.aims.media.CompactDisc;
6  import hust.soict.ite6.aims.media.DigitalVideoDisc;
7  import hust.soict.ite6.aims.media.Disc;
8  import hust.soict.ite6.aims.media.Media;
9  import hust.soict.ite6.aims.media.Track;
10 import hust.soict.ite6.aims.store.Store;
11
12 public class Aims {
13
14     private static Store store = new Store();
15     private static Cart cart = new Cart();
16
17     public static void main(String[] args) {
18         initSetup();
19         boolean exit = false;
20         while (!exit) {
21             showMenu();
22
23             Scanner scanner = new Scanner(System.in);
24             int option = scanner.nextInt();
25             scanner.nextLine();
26
27             switch (option) {
28                 case 0:
29                     exit = true;
30                     System.out.println("Exiting");
31                     break;
32                 case 1:
33                     clearConsole();
34                     storeMenu(scanner);
35                     break;

Problems  Javadoc  Declaration  Console
Aims [Java Application] /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java (Dec 1, 2024, 11:48:29 PM) [pid: 3435]
0 - CD: Adele - Category: Music - Artistnull - Length: 864 seconds - Cost: 9.98$
0 - CD: The Gods We Can Touch - Category: Music - Artistnull - Length: 583 seconds - Cost: 2000.22$
0 - CD: Purpose - Category: Music - Artistnull - Length: 520 seconds - Cost: 1000.98$
**************************************************
Options:
--------------------------------
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4
```



```
Cart.java  Aims.java  Store.java  Book.java  Media.java  Disc.java  MediaTest.java  MediaCom
55             System.out.println();
56         }
57     }
58     public static void initSetup() {
59
60         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19
61         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star War", "Science Fiction", "George Lucas", 87, 2
62         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
63         store.addMedia(dvd1);
64         store.addMedia(dvd2);
65         store.addMedia(dvd3);
66
67
68         Book book = new Book("The Valley of Fear", "Detective", 20.00f);
69         Book book1 = new Book("A Living Remedy: A Memoir", "Biography", 202.00f);
70         Book book2 = new Book("On the Origin of Time: Stephen Hawking's Final Theory", "Science", 120.00f
71         store.addMedia(book);
72         store.addMedia(book1);
73         store.addMedia(book2);
74
75
76         CompactDisc cd1 = new CompactDisc("Adele", "Music","Adele", 9.98f);
77         Track track1CD1 = new Track("All Night Parking (interlude)", 161);
78         Track track2CD1 = new Track("To Be Loved", 403);
79         Track track3CD1 = new Track("Woman Like Me", 300);
80         cd1.addTrack(track1CD1);
81         cd1.addTrack(track2CD1);
82         cd1.addTrack(track3CD1);
83
84         CompactDisc cd2 = new CompactDisc("The Gods We Can Touch", "Music","Aurora", 2000.22f);
85         Track track1CD2 = new Track("Everything Matters", 180+34);
86         Track track2CD2 = new Track("Blood in the Wine", 180+30);
87         Track track3CD2 = new Track("Artemis", 60*2+39);
88         cd2.addTrack(track1CD2);

Problems  Javadoc  Declaration  Console
Aims [Java Application] /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java (Dec 1, 2024, 11:48:29 PM) [pid: 3435]
--------------------------------
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4
3
Enter the title of the media (type 0 to stop):
To Be Loved
***MEDIA NOT FOUND***
Enter the title of the media (type 0 to stop):
```