BÁO CÁO THỰC HÀNH LAB 04
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục nội dung

Mục lục hình ảnh

2. Additional requirements of AIMS

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

import ...

no usages  new *
public class Book extends Media {

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

    no usages  new *
    public Book(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

    no usages  new *
    public void addAuthor(String authorName) {
        for(String author: authors) {
            if(author.equals(authorName)) {
                System.out.println("Author is already added!");
                return ;
            }
        }
        authors.add(authorName);
        System.out.println("Author is added!");
    }
    no usages  new *
    public void removeAuthor(String authorName) {
        for(String author: authors) {
            if(author.equals(authorName)) {
```

Figure 1 class Book

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

2 usages  2 inheritors  new *
public class Disc extends Media {
    2 usages
    String director;
    5 usages
    int length;

    5 usages  new *
    public Disc() {
    }

    1 usage  new *
    public Disc(String director, String category, String title, float cost){
        super(title,category,cost);
        this.setDirector(director);
    }

    no usages  new *
    public Disc(String title, String category, String director, int length, float cost){
        this(director,category,title,cost);
        this.setLength(length);
    }

    no usages  new *
    public Disc(String director, int length) {
        super();
        this.setDirector(director);
        this.setLength(length);
```

Figure 2: Class Disc

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

import java.util.ArrayList;
//import hust.soict.dsai.aims.media.Track;

no usages  new *
public class CompactDisc extends Disc implements Playable{

    2 usages
    private String artist;
    8 usages
    private ArrayList<Track> tracks = new ArrayList<~>();

    no usages  new *
    public CompactDisc() {
        // TODO Auto-generated constructor stub
    }

    no usages  new *
    public CompactDisc(String title, String artist) {
        this(title);
        this.setArtist(artist);
    }


    2 usages  new *
    public CompactDisc(String title) { super(title); }

    1 usage  new *
    public CompactDisc(String title, String category, float cost){
        this(title);
        this.setCategory(category);
```

Figure 3 CD class

```java
package hust.soict.dsai.aims.media;
no usages  new *
public class DigitalVideoDisc extends Disc implements Playable {
    // Khai báo thuộc tính
    8 usages
    private static int nbDigitalVideoDiscs=0;
    4 usages
    private String director;

    // Constructor

    no usages  new *
    public DigitalVideoDisc(String title) {
        super();
        this.title = title;
        nbDigitalVideoDiscs++;
        this.id = nbDigitalVideoDiscs;
    }

    no usages  new *
    public DigitalVideoDisc(String title, String category, float cost) {
        super();
        this.title = title;
        this.category = category;
        this.cost = cost;
        nbDigitalVideoDiscs++;
        this.id = nbDigitalVideoDiscs;
    }

    no usages  new *
    public DigitalVideoDisc(String title, String category, String director, float cost) {
        super();
```

Figure 4 DVD class

3. Creating the Book class

3.1 Book class đã sửa theo Media class

```
package hust.soict.dsai.aims.media;

import java.util.ArrayList;
import java.util.List;

public class Book extends Media {

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

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

   public void addAuthor(String authorName) {
      for(String author: authors) {
         if(author.equals(authorName)) {
            System.out.println("Author is already added!");
            return ;
         }
      }
      authors.add(authorName);
      System.out.println("Author is added!");
   }
   public void removeAuthor(String authorName) {
      for(String author: authors) {
         if(author.equals(authorName)) {
            authors.remove(authorName);
            System.out.println("Author is removed!");
            return ;
         }
      }
      System.out.println("Author is not added yet!");
   }

   public String toString() {
      return ("Book - " + this.getTitle() + " - " + this.getCategory() + ": " + this.getCost() + "$");
   }
}
```

4. Creating the abstract Media class

4.1 Media class

```
package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class Media {

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

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

   public Media(int id, String title, String category, float cost) {
      this(category, title, cost);
      this.setId(id);
   }

   public Media(String title){
      this.setTitle(title);
   }

   public Media(String title, String category, float cost){
      this(title);
      this.setCategory(category);
      this.setCost(cost);
   }

   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;
    }
    public void setCategory(String category) {
        this.category = category;
    }
    public float getCost() {
        return cost;
    }
    public void setCost(float cost) {
        this.cost = cost;
    }

    @Override
    public boolean equals(Object object) {

        if(object instanceof Media) {
            Media media = (Media) object;
            return media.getTitle().equals(this.getTitle());
        }
        return false;
    }

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

}
```

5. Creating the CompactDisc class

5.1 Disc class

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

2 usages  2 inheritors  new *
public class Disc extends Media {
    2 usages
    String director;
    5 usages
    int length;

    5 usages  new *
    public Disc() {
    }

    1 usage  new *
    public Disc(String director, String category, String title, float cost){
        super(title,category,cost);
        this.setDirector(director);
    }

    no usages  new *
    public Disc(String title, String category, String director, int length, float cost){
        this(director,category,title,cost);
        this.setLength(length);
    }

    no usages  new *
    public Disc(String director, int length) {
        super();
        this.setDirector(director);
        this.setLength(length);
    }
```

Figure 5 Disc class

5.2 Track class

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

10 usages  new *
public class Track implements Playable{

    3 usages
    private String title;
    3 usages
    private int length;
    no usages  new *
    public Track() {
        // TODO Auto-generated constructor stub
    }
    no usages  new *
    public Track(String title, int length) {
        this.setLength(length);
        this.setTitle(title);
    }

    3 usages  new *
    public String getTitle() { return title; }
    1 usage  new *
    public void setTitle(String title) { this.title = title; }
    3 usages  new *
    public int getLength() { return length; }
    1 usage  new *
    public void setLength(int length) { this.length = length; }

    1 usage  new *
    public void play() {
        System.out.println("Playing track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    }
```

Figure 6 Track class

5.3 CompactDisc class

```
package hust.soict.dsai.aims.media;

import java.util.ArrayList;
//import hust.soict.dsai.aims.media.Track;

public class CompactDisc extends Disc implements Playable{

   private String artist;
   private ArrayList<Track> tracks = new ArrayList<Track>();

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

   public CompactDisc(String title, String artist) {
      this(title);
      this.setArtist(artist);
   }


   public CompactDisc(String title){
      super(title);
   }

   public CompactDisc(String title, String category, float cost){
      this(title);
      this.setCategory(category);
      this.setCost(cost);
   }

   public CompactDisc(String director, String category, String title, float cost){
      this(category,title,cost);
      this.setDirector(director);
   }

   public void addTrack(Track newTrack) {
      if(this.tracks.contains(newTrack)) {
         System.out.println("Track already exists!");
         return ;
      }
      this.tracks.add(newTrack);
```

```
        System.out.println("Track is added!");
    }
    public void removeTrack(Track track) {
        if(this.tracks.contains(track)) {
            this.tracks.remove(track);
            System.out.println("Track is removed!");
            return ;
        }
        System.out.println("Track does not exist!");
    }

    @Override
    public int getLength() {
        int sum = 0;
        for(Track track: tracks) {
            sum += track.getLength();
        }
        return sum;
    }

    public String getArtist() {
        return artist;
    }
    public void setArtist(String artist) {
        this.artist = artist;
    }
//  public ArrayList<Track> getTrack() {
//      return track;
//  }
//  public void setTrack(ArrayList<Track> track) {
//      this.track = track;
//  }
    public void play() {
        System.out.println("CD contains " + this.tracks.size() + " tracks!");
        System.out.println("Tracks:");
        int i=0;
        for(Track track: tracks) {
            System.out.println(i++ + ". " + track.getTitle());
        }
        for(Track track: tracks) {
            track.play();
        }
    }
```

```
    @Override
    public String toString() {
        return ("CD - " + this.getTitle() + " - " + this.getCategory() + " - " + this.getDirector() + " - " + this.getLength() + ": "
+ this.getCost() + "$");
    }


}
```

6. Create the Playable interface

```
package hust.soict.dsai.aims.media;

3 usages  3 implementations  new *
public interface Playable {
    1 usage  3 implementations  new *
    public void play();
}
```

Figure 7 Playable interface

```
//  }
    1 usage  new *
    public void play() {
        System.out.println("CD contains " + this.tracks.size() + " tracks!");
        System.out.println("Tracks:");
        int i=0;
        for(Track track: tracks) {
            System.out.println(i++ + ". " + track.getTitle());
        }
        for(Track track: tracks) {
            track.play();
        }
    }
```

Figure 8 play in CompactDisc

```
    1 usage  new *
    public void play() {
        System.out.println("Playing track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    }

    new *
    @Override
```

Figure 9 play in Track class

```
    1 usage  new *
    public void play() {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
```

Figure 10 play in DVD class

7. Update the Cart class to work with Media

7.1 Cart class

```
package hust.soict.dsai.aims.cart;

import java.util.ArrayList;

import hust.soict.dsai.aims.media.CompactDisc;
import hust.soict.dsai.aims.media.DigitalVideoDisc;
import hust.soict.dsai.aims.media.Media;

public class Cart {
   private ArrayList<Media> itemsOrdered = new ArrayList<Media>();
   public void addMedia(Media d) {
      itemsOrdered.add(d);
      System.out.println("The media has been added!");
   }
   public void addMedia(Media [] dList) {
      for (Media disc: dList) {
         addMedia(disc);
      }
   }
   public void addMedia(Media d1,Media d2) {
      addMedia(d1);
      addMedia(d2);
   }

   public void removeMedia (Media d) {
      try {
         itemsOrdered.remove(d);
         System.out.println("The media is removed!");
      }
      catch (Exception e) {
         System.out.println("Media has not been added yet!");
         return ;
      }
   }
   public float totalCost() {
      float total = 0;
      for(int i= 0; i< itemsOrdered.size(); i++) {
         total += itemsOrdered.get(i).getCost();
      }
      return total;
   }
```

```
    // Method overloading
    // Print()
    public void print() {
        System.out.println("**********CART**********");
        for (int i = 0; i < this.itemsOrdered.size(); i++ ) {
            System.out.println(i+1 + ". " + itemsOrdered.get(i));
        }
        System.out.println("Total cost: " + totalCost() + "$");
        System.out.println("**************************************************");
    }
    public ArrayList<Media> getItemsOrdered() {
        return itemsOrdered;
    }

    // Search
    public void searchId(int id) {
        for(Media item: itemsOrdered) {
            if(item.getId() == id) {
                System.out.println(item);
                return ;
            }
        }
        System.out.println("No media is matched!");
    }
    public void searchTitle(String title) {
        for(Media item: itemsOrdered) {
            if(item.getTitle().equals(title)) {
                System.out.println(item);
                return ;
            }
        }
        System.out.println("No media is matched!");
    }
}
```

8. Update the Store class to work with Media

8.1 Store class updated

```
package hust.soict.dsai.aims.store;

import java.util.ArrayList;
```

```
import hust.soict.dsai.aims.media.Book;
import hust.soict.dsai.aims.media.CompactDisc;
import hust.soict.dsai.aims.media.DigitalVideoDisc;
import hust.soict.dsai.aims.media.Media;

public class Store {
   private ArrayList<Media> itemsInStore = new ArrayList<Media>();
   private int nbItems = 0;
   public void addMedia(Media d) {
      itemsInStore.add(d);
      nbItems += 1;
      System.out.println("The media is added to store!");
   }
   public void removeMedia(Media d) {
      try {
         itemsInStore.remove(d);
         nbItems -= 1;
         System.out.println("The media is removed from store!");
      }
      catch (Exception e) {
         System.out.println("Media doesn't exist!");
         return ;
      }
   }
   public void print() {
      int num = 1;
      System.out.println("**********STORE**********");
      for(Media media: this.itemsInStore) {
         if (media instanceof DigitalVideoDisc) {
            DigitalVideoDisc dvd = (DigitalVideoDisc) media;
            System.out.println(num + ". " +"DVD"+ ". " +  dvd.getTitle() + ". " + dvd.getCategory() + ". " +
dvd.getDirector() + ". " + dvd.getLength() + ". " + dvd.getCost());
         }
         else if (media instanceof CompactDisc) {
            CompactDisc cd = (CompactDisc) media;
            System.out.println(num  + ". " + "CD" + ". " + cd.getTitle() + ". " + cd.getCategory() + ". " + cd.getArtist() + ". "
+ cd.getLength() + ". " + cd.getCost());
         }
         else if (media instanceof Book) {
            Book book = (Book) media;
            System.out.println(num  + ". " + "Book" + ". " + book.getTitle() + ". " + book.getCategory() + ". " +
book.getCost());
         }
         num++;
```

```
        }
        System.out.println("**************************************************");
    }
    public int getNbItems() {
        return nbItems;
    }
    public ArrayList<Media> getItemsInStore() {
        return itemsInStore;
    }
}
```

9. Constructors of whole classes and parent classes

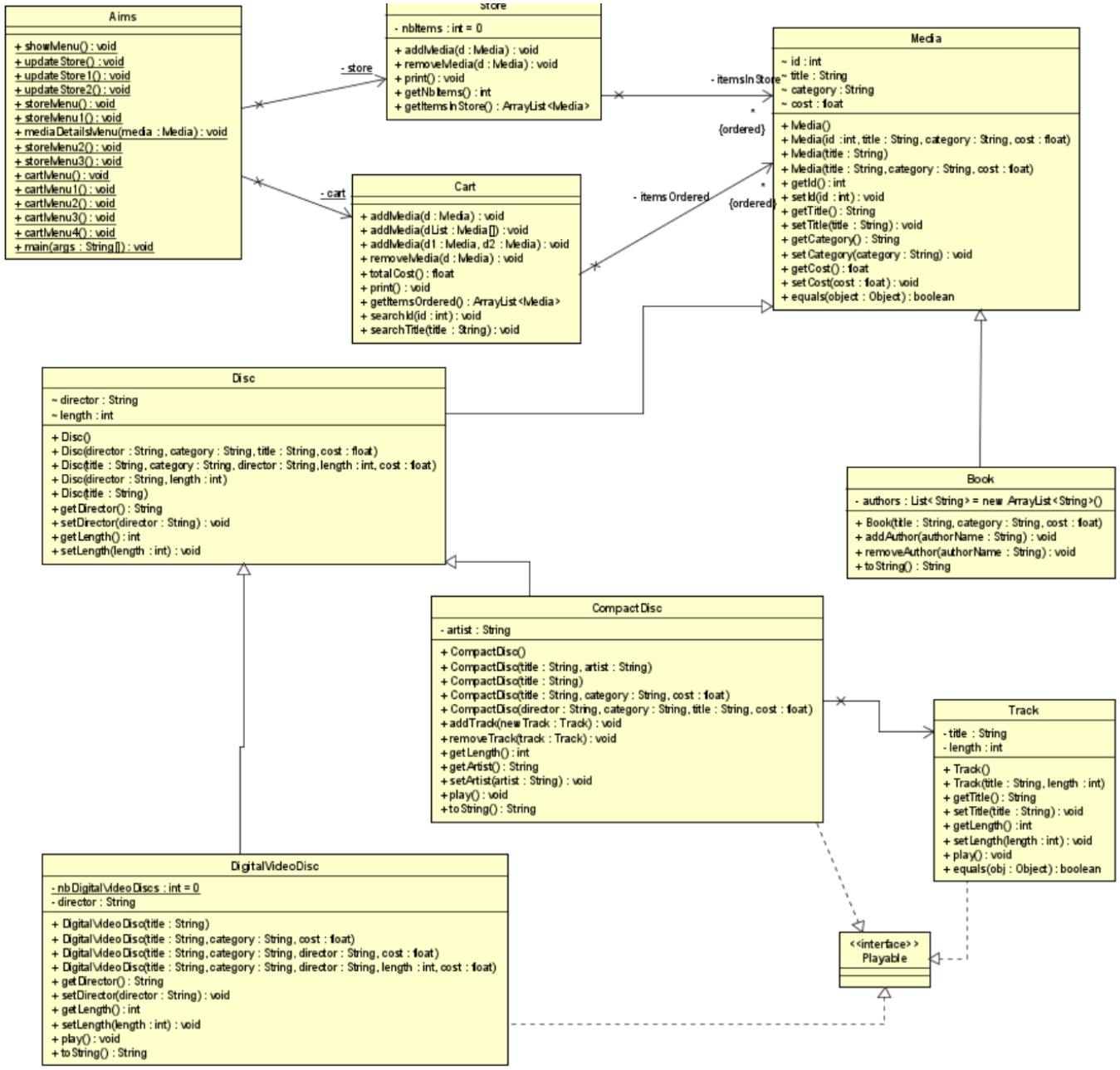


Figure 11 Update class-diagrams

10. Unique item in a list

```java
@Override
public boolean equals(Object obj) {

    if (obj instanceof Track) {
        Track track = (Track) obj;
        if ((title == track.getTitle()) && (length == track.getLength())) {
            return true;
        }
    }
    return false;
}
```

Figure 12 Override in track class

```java
@Override
public boolean equals(Object object) {

    if(object instanceof Media) {
        Media media = (Media) object;
        return media.getTitle().equals(this.getTitle());
    }
    return false;
}
```

Figure 13 Override in Media class

11. Polymorphism with toString() method

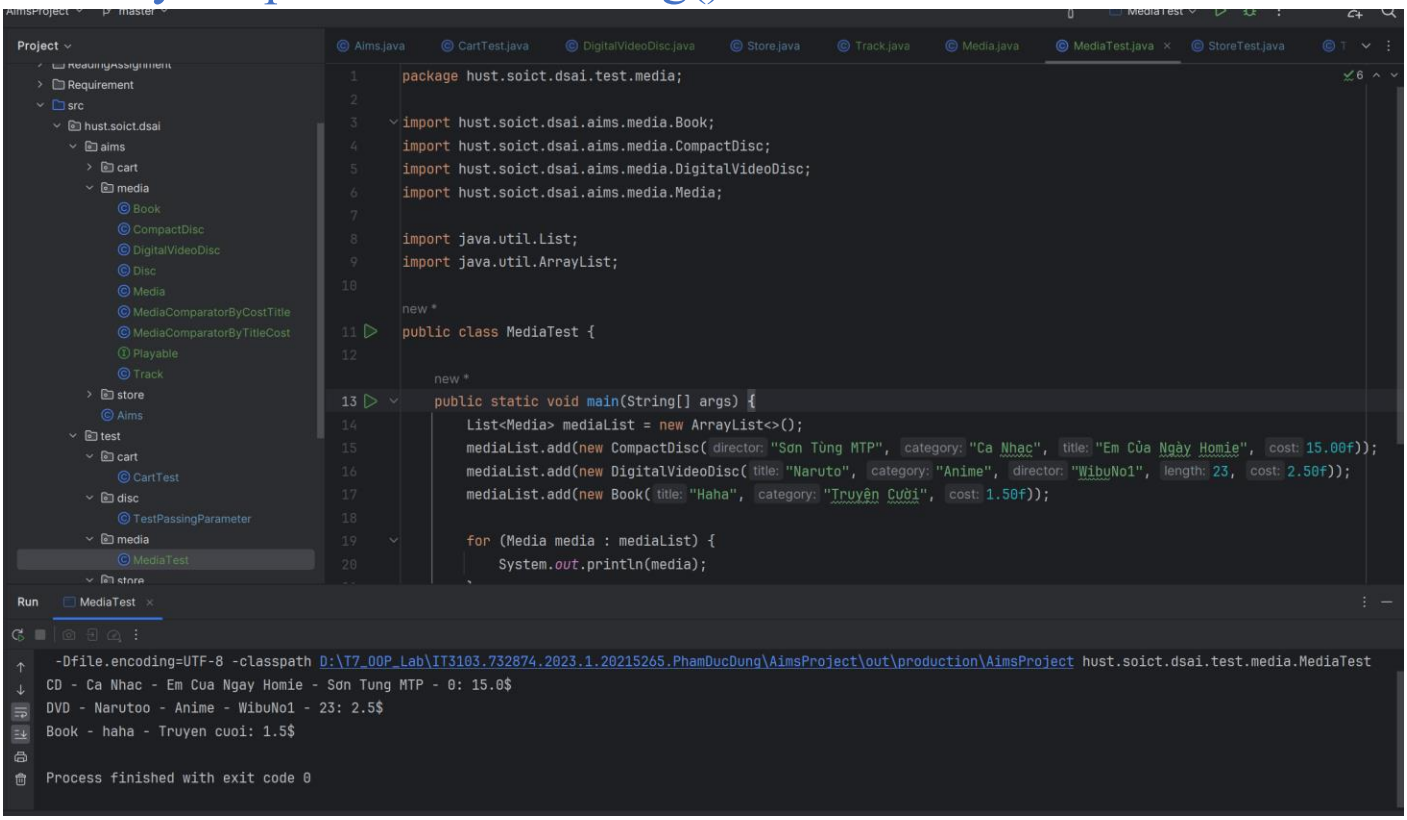


Figure 14 Media Test

Giải thích về Đa hình (Polymorphism): Đa hình được thể hiện qua việc sử dụng phương thức toString(). Mỗi lớp (CompactDisc, DigitalVideoDisc, Book) đều có một cách triển khai riêng cho phương thức toString(). Khi duyệt qua danh sách mediaList và in từng phần tử, phương thức toString() tương ứng với kiểu thực tế của mỗi đối tượng (CD, DVD, hoặc Book) sẽ được gọi. Điều này chứng minh tính đa hình: một phương thức duy nhất (toString()) được sử dụng theo nhiều cách khác nhau tùy thuộc vào đối tượng đang được tham chiếu.

12. Sort media in the cart

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

}
```

Figure 15 In class Media

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

import java.util.Comparator;

1 usage  new *
public class MediaComparatorByCostTitle implements Comparator<Media> {
    1 usage  new *
    public MediaComparatorByCostTitle() {
        // TODO Auto-generated constructor stub
    }

    new *
    @Override
    public int compare(Media md1, Media md2) {
        if(md1.getCost() != md2.getCost()) {
            return Float.compare(md1.getCost(), md2.getCost());
        }

        return md1.getTitle().compareTo(md2.getTitle());
    }
}
```

Figure 16 MediaComparatorByCostTitle

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

import java.util.Comparator;

1 usage  new *
public class MediaComparatorByTitleCost implements Comparator<Media> {
    1 usage  new *
    public MediaComparatorByTitleCost() {
        // TODO Auto-generated constructor stub
    }

    new *
    @Override
    public int compare(Media md1, Media md2) {
        if(md1.getTitle().equals(md2.getTitle())) {
            return Float.compare(md1.getCost(),md2.getCost());
        }

        return md1.getTitle().compareTo(md2.getTitle());
    }
}
```

Figure 17 MediaComparatorByTitleCost

13. Create a complete console application in the Aims class

13.1 Aims class full

```
package hust.soict.dsai.aims;

import hust.soict.dsai.aims.media.DigitalVideoDisc;
import hust.soict.dsai.aims.media.Media;
import hust.soict.dsai.aims.media.CompactDisc;
import hust.soict.dsai.aims.media.Book;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.store.Store;

import java.util.Collections;
```

```
import java.util.Scanner;

public class Aims{

    private static Cart cart = new Cart();
    private static Store store = new Store();
    private static Scanner input = new Scanner(System.in);

    public static void showMenu() {
        System.out.println("AIMS: ");
        System.out.println("--------------------------------");
        System.out.println("1. View store");
        System.out.println("2. Update store");
        System.out.println("3. See the current cart");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3");

        int n = input.nextInt();
        input.nextLine();
        if(n == 1) {
            storeMenu();
        }
        else if (n == 2) {
            updateStore();
        }
        else if (n == 3) {
            cartMenu();
        }
        else if (n == 0) {
            System.out.println("Exit successfully!");
        }
        else {
            System.out.println("Error!");
            showMenu();
        }
    }
    public static void updateStore() {
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. Add media");
        System.out.println("2. Remove media");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2");
```

```
        int n = input.nextInt();
        input.nextLine();
        if(n == 1) {
            updateStore1();
        }
        else if(n == 2) {
            updateStore2();
        }
        else if (n == 0) {
            System.out.println("Exit successfully!");
            showMenu();
        }
        else {
            System.out.println("Error!");
            showMenu();
        }
    }
    public static void updateStore1() {
        System.out.print("Enter title of the media: ");
        String title = input.nextLine();
        System.out.print("Enter category of the media: ");
        String category = input.nextLine();
        System.out.print("Enter cost of the media: ");
        float cost = input.nextFloat();
        input.nextLine();

        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. DigitalVideoDisc");
        System.out.println("2. CompactDisc");
        System.out.println("3. Book");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3");

        int n = input.nextInt();
        if (n == 1) {
            DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, cost);
            store.addMedia(dvd);
            updateStore();
        }
        else if (n == 2) {
            CompactDisc cd = new CompactDisc(title, category, cost);
            store.addMedia(cd);
```

```
            updateStore();
        }
        else if (n == 3) {
            Book book = new Book(title, category, cost);
            store.addMedia(book);
            updateStore();
        }
        else if (n == 0) {
            System.out.println("Exit successfully!");
            updateStore();
        }
        else {
            System.out.println("Error!");
            showMenu();
        }
    }
    public static void updateStore2() {
        System.out.print("Enter the title of the media:");
        String title = input.nextLine();
        for(Media media: store.getItemsInStore()) {
            if(media.getTitle().equals(title)) {
                store.removeMedia(media);
                updateStore();
                return ;
            }
        }
        System.out.println("Media does not exist!");
        updateStore();
    }

    public static void storeMenu() {
        store.print();
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. See a media's details");
        System.out.println("2. Add a media to cart");
        System.out.println("3. Play a media");
        System.out.println("4. See current cart");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3-4");

        int n = input.nextInt();
        input.nextLine();
```

```
        if (n == 1) {
            storeMenu1();
        }
        else if (n == 2) {
            storeMenu2();
        }
        else if (n == 3) {
            storeMenu3();
        }
        else if (n == 4) {
            cartMenu();
        }
        else if (n == 0) {
            showMenu();
        }
        else {
            System.out.println("Error!");
            showMenu();
        }
    }

    public static void storeMenu1() {
        System.out.println("Enter the title of the media");
        String title = input.nextLine();
        for(Media media: store.getItemsInStore()) {
            if(media.getTitle().equals(title)) {
                media.toString();
                mediaDetailsMenu(media);
                return ;
            }
        }
        System.out.println("Media does not exist!");
        storeMenu();
    }

    public static void mediaDetailsMenu(Media media) {
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. Add to cart");
        System.out.println("2. Play");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2");
```

```
        int n = input.nextInt();
        input.nextLine();
        if (n == 1) {
            cart.addMedia(media);
            storeMenu();
        }
        else if (n == 2) {
            if (media instanceof DigitalVideoDisc) {
                DigitalVideoDisc dvd = (DigitalVideoDisc) media;
                dvd.play();
            }
            else if (media instanceof CompactDisc) {
                CompactDisc cd = (CompactDisc) media;
                cd.play();
            }
            else if (media instanceof Book) {
                System.out.println("Can not be played!");
            }
            System.out.println("Do you want to add to cart?");
            System.out.println("1. Yes");
            System.out.println("0. No");
            int opt = input.nextInt();
            input.nextLine();
            if(opt == 1) {
                cart.addMedia(media);
                storeMenu();
            }
            else if(opt == 0) {
                storeMenu();
            }
            else {
                System.out.println("Error!");
                showMenu();
            }
        }
        else if (n == 0) {
            storeMenu();
        }
        else {
            System.out.println("Error!");
            showMenu();
        }
    }
```

```
public static void storeMenu2() {
    System.out.print("Enter the title of the media: ");
    String title = input.nextLine();
    for(Media media: store.getItemsInStore()) {
        if(media.getTitle().equals(title)) {
            cart.addMedia(media);
            storeMenu();
            return ;
        }
    }
    System.out.println("Media does not exist!");
    storeMenu();
}
public static void storeMenu3() {
    System.out.print("Enter the title of the media: ");
    String title = input.nextLine();
    for(Media media: store.getItemsInStore()) {
        if(media.getTitle().equals(title)) {
            if (media instanceof DigitalVideoDisc) {
                DigitalVideoDisc dvd = (DigitalVideoDisc) media;
                dvd.play();
            }
            else if (media instanceof CompactDisc) {
                CompactDisc cd = (CompactDisc) media;
                cd.play();
            }
            else if (media instanceof Book) {
                System.out.println("Can not be played!");
            }
            storeMenu();
            return ;
        }
    }
    System.out.println("Media does not exist!");
    storeMenu();
}

public static void cartMenu() {
    cart.print();
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. Filter medias in cart");
    System.out.println("2. Sort medias in cart");
    System.out.println("3. Remove media from cart");
```

```
        System.out.println("4. Play a media");
        System.out.println("5. Play order");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3-4-5");
        int n = input.nextInt();
        input.nextLine();
        if (n == 1) {
            cartMenu1();
        }
        else if (n == 2) {
            cartMenu2();
        }
        else if (n == 3) {
            cartMenu3();
        }
        else if (n == 4) {
            cartMenu4();
        }
        else if (n == 5) {
            System.out.println("Order is placed!");
            cart.getItemsOrdered().clear();
            cart.print();
        }
        else if (n == 0) {
            showMenu();
        }
        else {
            System.out.println("Error!");
            showMenu();
        }
    }

    public static void cartMenu1() {
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. Filter by id");
        System.out.println("2. Filter by title");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2");
        int n = input.nextInt();
        input.nextLine();
        if(n == 1) {
```

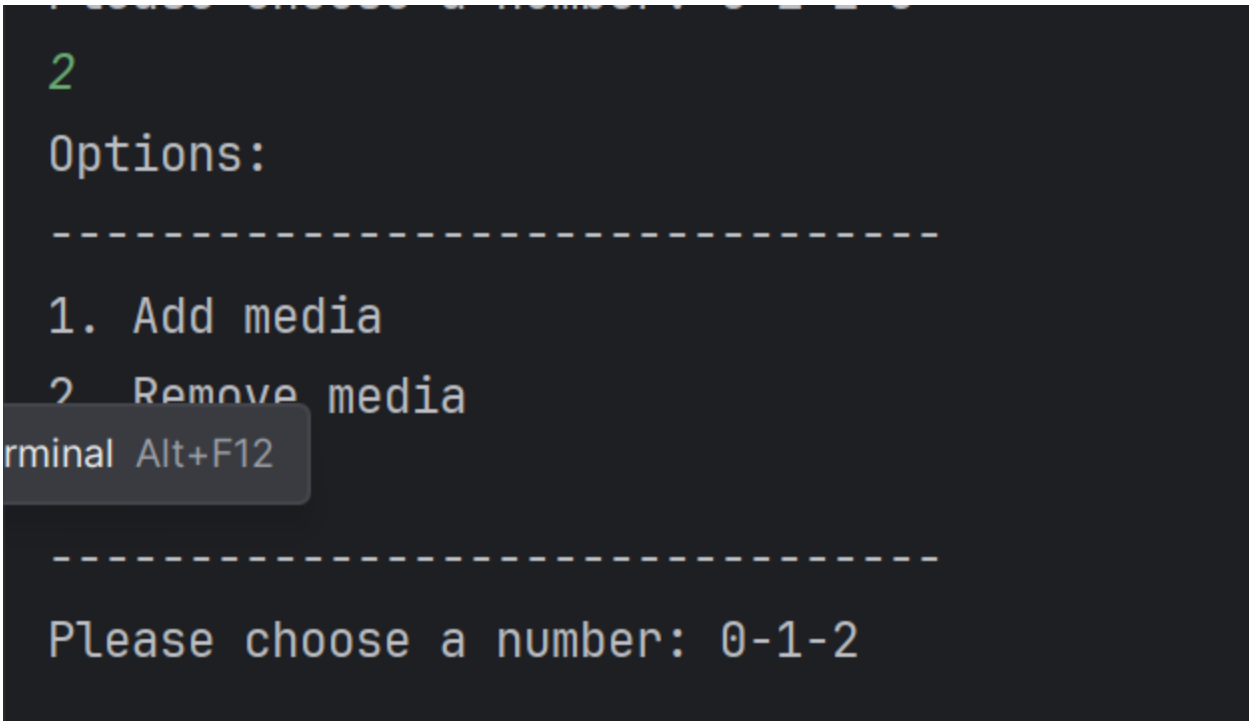
```
            System.out.print("Enter id: ");
            int id = input.nextInt();
            input.nextLine();
            cart.searchId(id);
            cartMenu();
        }
        else if(n == 2) {
            System.out.print("Enter title: ");
            String title = input.nextLine();
            cart.searchTitle(title);
            cartMenu();
        }
        else if(n == 0) {
            cartMenu();
        }
        else {
            System.out.println("Error!");
            cartMenu();
        }
    }
    public static void cartMenu2() {
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. Sort by title");
        System.out.println("2. Sort by cost");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2");
        int n = input.nextInt();
        input.nextLine();
        if (n == 1) {
            Collections.sort(cart.getItemsOrdered(),Media.COMPARE_BY_TITLE_COST);
            cart.print();
            cartMenu();
        }
        if (n == 2) {
            Collections.sort(cart.getItemsOrdered(),Media.COMPARE_BY_COST_TITLE);
            cart.print();
            cartMenu();
        }
        if (n == 0) {
            cartMenu();
        }
        else {
```

```
            System.out.println("Error!");
            cartMenu();
        }
    }
    public static void cartMenu3() {
        System.out.print("Enter the title of the media to remove: ");
        String title = input.nextLine();
        for(Media media: cart.getItemsOrdered()) {
            if(media.getTitle().equals(title)) {
                cart.removeMedia(media);
                cartMenu();
                return ;
            }
        }
        System.out.println("Media does not exist!");
        cartMenu();
    }
    public static void cartMenu4() {
        System.out.print("Enter the title of the media to play: ");
        String title = input.nextLine();
        for (Media media: store.getItemsInStore()) {
            if (media.getTitle().equals(title)) {
                if (media instanceof CompactDisc){
                    CompactDisc cd = (CompactDisc) media;
                    cd.play();
                    cartMenu();
                    return;
                } else if (media instanceof DigitalVideoDisc){
                    DigitalVideoDisc dvd = (DigitalVideoDisc) media;
                    dvd.play();
                    cartMenu();
                    return;
                } else if (media instanceof Book) {
                    System.out.println("Cannot play a book");
                    cartMenu();
                    return ;
                }
            }
        }
        System.out.println("Media does not exist!");
        cartMenu();
    }
    public static void main(String[] args) {
        showMenu();
```

```
    }
}
```

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

```
1
************************STORE************************
*****************************************************
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
```


2
Options:
1. Add media
rminal Alt+F12
Please choose a number: 0-1-2

```
3
**********CART**********
Total cost: 0.0$
**************************************************
Options:
--------------------------------
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Play order
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4-5
```

Figure 18 Preview All