

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BÁO CÁO THỰC HÀNH**  
IT3103-744530-2024.1

**BÀI THỰC HÀNH 4**

Họ và tên sv: Lê Phúc Hưng

MSSV: 20215276

Lớp: CNTT VP 01 K66

GVHD: Lê Thị Hoa

HTGD: Bùi Trọng Dũng

Hà Nội, 11/2024

# BÁO CÁO THỰC HÀNH LAB 4

## THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Contents

BÁO CÁO THỰC HÀNH LAB 4.....	1
THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG.....	1
Danh mục hình ảnh.....	2
1. Create the Book class .....	4
1.1. Mã chương trình cài tiến .....	4
2. Create the abstract Media class .....	5
2.2. Mã nguồn cài tiến lớp DigitalVideoDisc .....	7
2.3. Mã nguồn cài tiến lớp Book .....	7
3. Create the CompactDisc class .....	8
3.1. Tao lớp Disc .....	8
3.2. Tạo lớp Track .....	9
3.3. Tạo lớp CompactDisc.....	10
4. Create the Playable interface .....	12
4.1. Mã nguồn giao diện Playable .....	12
4.2. Triển khai giao diện Playable ở các lớp khác .....	12
5. Update the Cart class to work with Media .....	15
6. Update the Store class to work with Media.....	17
7. Constructor of whole classes and parent classes .....	18
8. Unique item in a list.....	19
9. Polymorphism with <code>toString()</code> method .....	20
10. Sort media in the cart.....	22
11. Create a complete console application in the Aims class.....	23
12. Class diagram .....	29
13. Usecase diagram .....	30
14. Answer questions.....	31

## Danh mục hình ảnh

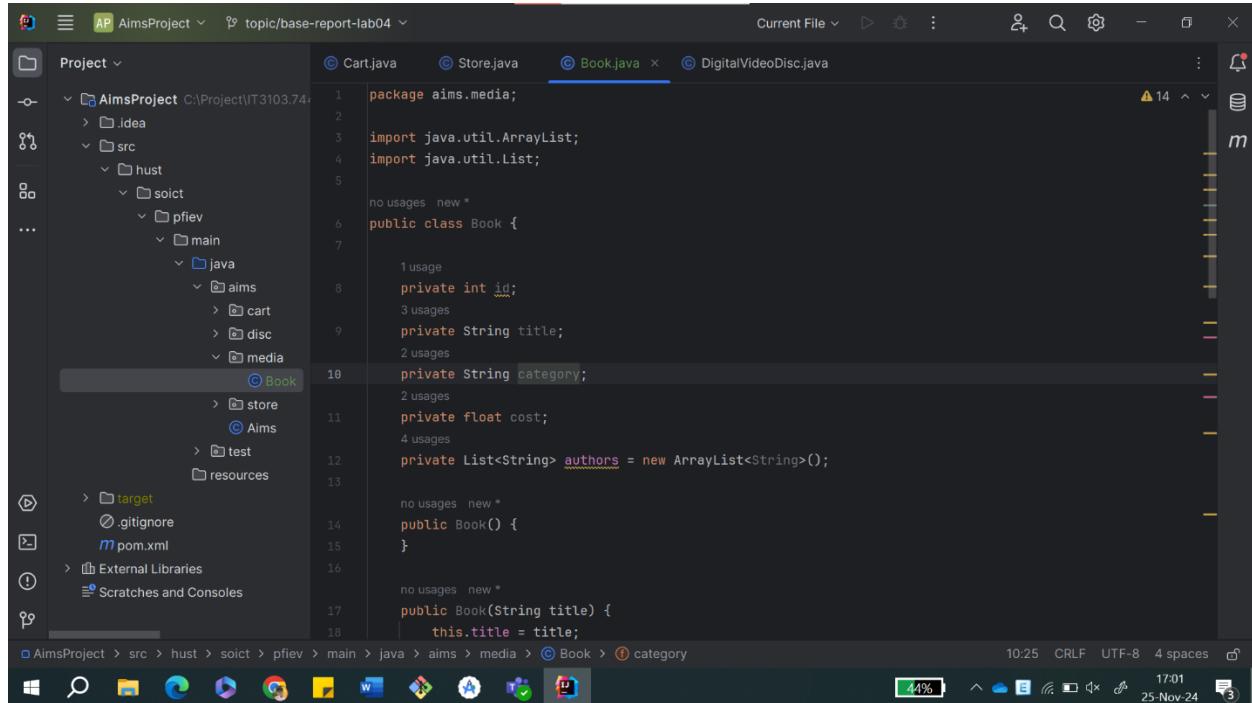
Hình 1.1: Mã nguồn lớp Book phần 1 .....	4
Hình 1.2: Mã nguồn lớp Book phần 2 .....	5
Hình 1.3: Mã nguồn lớp Book phần 3 .....	5
Hình 2.1: Mã nguồn lớp Media phần 1 .....	5
Hình 2.2: Mã nguồn lớp Media phần 2 .....	6
Hình 2.3: Mã nguồn lớp Media phần 3 .....	6
Hình 2.4: Mã nguồn cải tiến lớp DigitalVideoDisc .....	7
Hình 2.5: Mã nguồn cải tiến lớp Book .....	7
Hình 3.1: Mã nguồn lớp Disc phần 1 .....	8
Hình 3.2: Mã nguồn lớp Disc phần 2 .....	8
Hình 3.3: Mã nguồn lớp Track phần 1 .....	9
Hình 3.4: Mã nguồn lớp Track phần 2 .....	9
Hình 3.5: Mã nguồn lớp CompactDisc phần 1 .....	10
Hình 3.6: Mã nguồn lớp CompactDisc phần 2 .....	10
Hình 3.7: Mã nguồn lớp CompactDisc phần 3 .....	11
Hình 3.8: Mã nguồn lớp CompactDisc phần 4 .....	11
Hình 4.1: Mã nguồn giao diện Playable .....	12
Hình 4.2: Mã nguồn cải tiến lớp DigitalVideoDisc phần 1 .....	12
Hình 4.3: Mã nguồn cải tiến lớp DigitalVideoDisc phần 2 .....	13
Hình 4.4: Mã nguồn cải tiến lớp CompactDisc phần 1 .....	13
Hình 4.5: Mã nguồn cải tiến lớp CompactDisc phần 2 .....	14
Hình 4.6: Mã nguồn cải tiến lớp Track phần 1 .....	14
Hình 4.7: Mã nguồn cải tiến lớp Track phần 2 .....	15
Hình 5.1: Mã nguồn cải tiến lớp Cart phần 1 .....	15
Hình 5.2: Mã nguồn cải tiến lớp Cart phần 2 .....	16
Hình 5.3: Mã nguồn cải tiến lớp Cart phần 3 .....	16
Hình 6.1: Mã nguồn cải tiến lớp Store phần 1 .....	17
Hình 6.2: Mã nguồn cải tiến lớp Store phần 2 .....	17
Hình 6.3: Mã nguồn cải tiến lớp Store phần 3 .....	18
Hình 8.1: Mã nguồn cập nhật lớp Media cho phương thức equals .....	19
Hình 8.2: Mã nguồn cập nhật lớp Track cho phương thức equals .....	19
Hình 9.1: Phương thức <code>toString()</code> lớp Book .....	20
Hình 9.2: Phương thức <code>toString()</code> lớp CompactDisc .....	20
Hình 9.3: Phương thức <code>toString()</code> lớp DigitalVideoDisc .....	21
Hình 9.4: Kết quả demo đa hình trong hướng đối tượng .....	21
Hình 10.1: Thêm các thuộc tính dùng để so sánh vào lớp Media .....	22
Hình 10.2: Phương thức <code>MediaComparatorByCostTitle()</code> .....	22
Hình 10.3: Phương thức <code>MediaComparatorByTitleCost()</code> .....	23
Hình 11.1: Giao diện dòng lệnh ban đầu của AIMS .....	23

Hình 11.2: Giao diện chức năng xem sản phẩm trong cửa hàng.....	24
Hình 11.3: Giao diện chức năng xem chi tiết sản phẩm.....	24
Hình 11.4: Giao diện chức năng thêm sản phẩm vào giỏ hàng.....	25
Hình 11.5: Giao diện chức năng phát sản phẩm.....	25
Hình 11.6: Giao diện chức năng xem giỏ hàng .....	26
Hình 11.7: Giao diện chức năng sắp xếp giỏ hàng.....	26
Hình 11.8: Giao diện chức năng đặt hàng .....	27
Hình 11.9: Giao diện chức năng thêm sản phẩm vào cửa hàng .....	27
Hình 11.10: Kết quả thêm sản phẩm vào cửa hàng.....	28
Hình 11.11: Giao diện chức năng thêm sản phẩm vào cửa hàng .....	28
Hình 11.12: Kết quả xóa sản phẩm khỏi cửa hàng.....	29
Hình 12.1: Class diagram AIMS Project.....	30
Hình 13.1: Usecase diagram AIMS Project .....	31

## 1. Create the Book class

### 1.1. Mã chương trình cài tiến

Dưới đây là mã chương trình trong file Book, đây là một lớp cho các đối tượng sách của Aims Project.



```

package aims.media;

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

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

    public Book() {
    }

    public Book(String title) {
        this.title = title;
    }

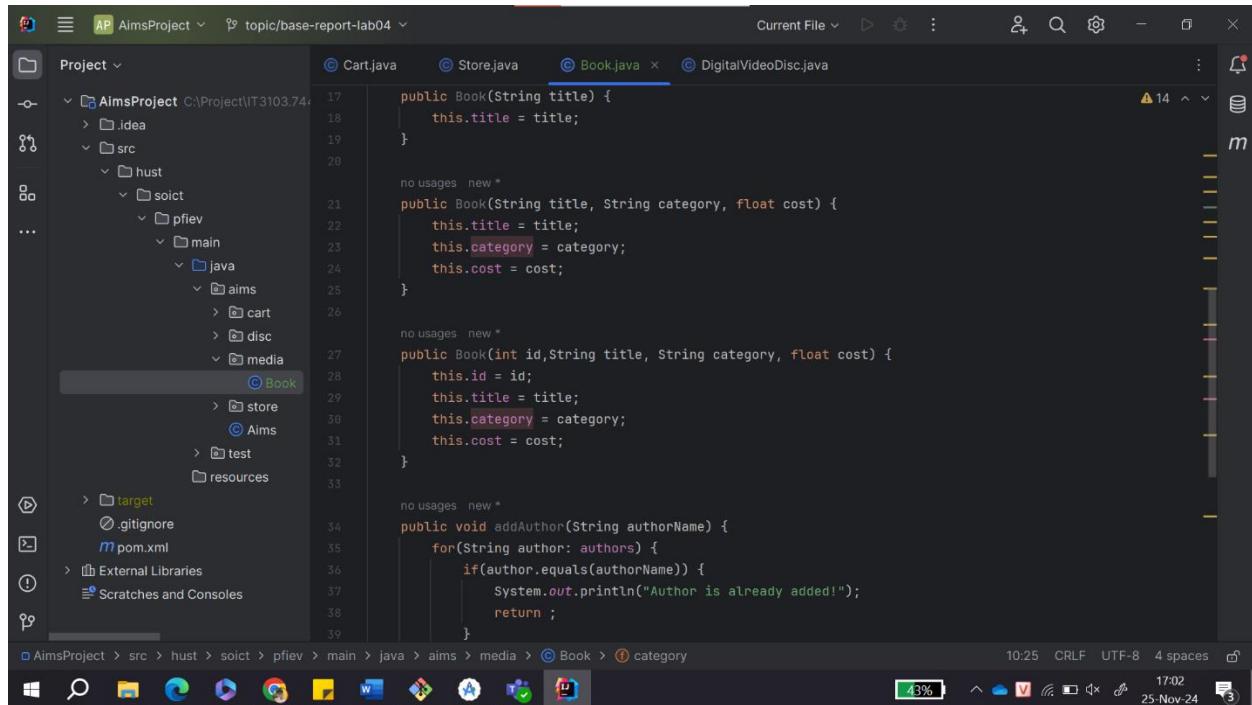
    public Book(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

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

```

Hình 1.1: Mã nguồn lớp Book phần 1

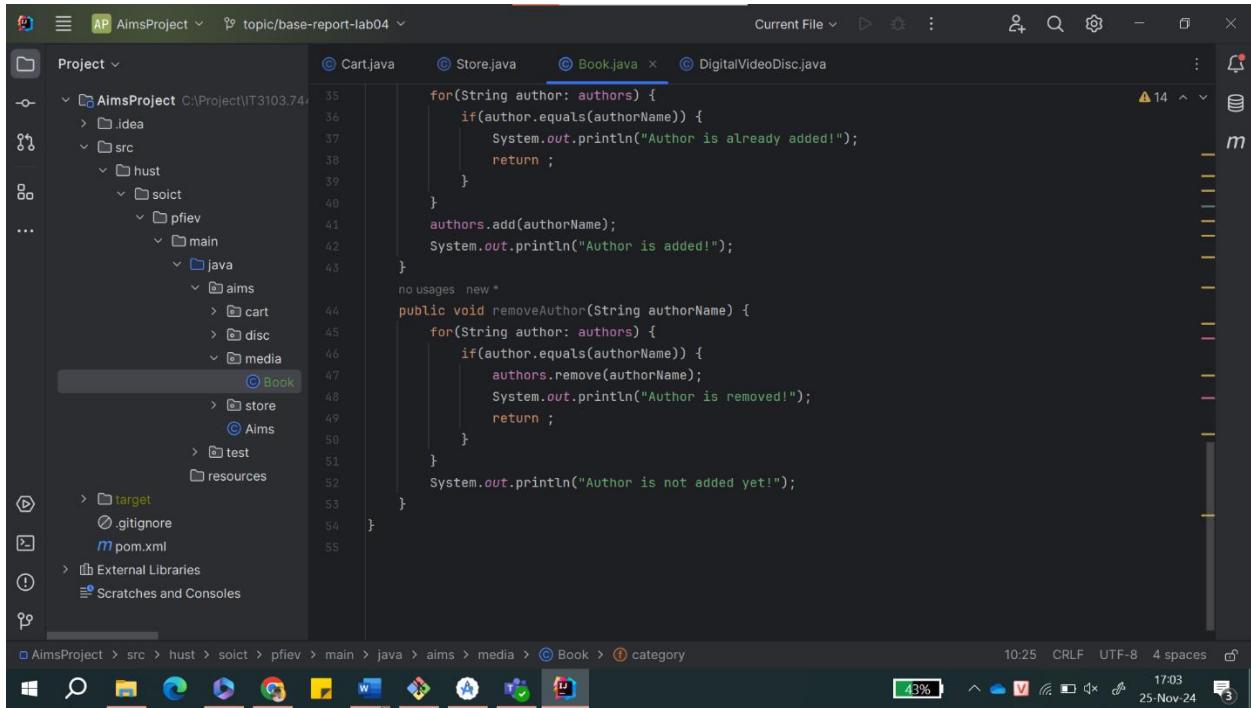


```

public void addAuthor(String authorName) {
    for(String author: authors) {
        if(author.equals(authorName)) {
            System.out.println("Author is already added!");
            return ;
        }
    }
    authors.add(authorName);
}

```

Hình 1.2: Mã nguồn lớp Book phần 2



```

package aims.media;
import java.util.ArrayList;
import java.util.List;

public class Book {
    private List<String> authors = new ArrayList<String>();

    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!");
    }

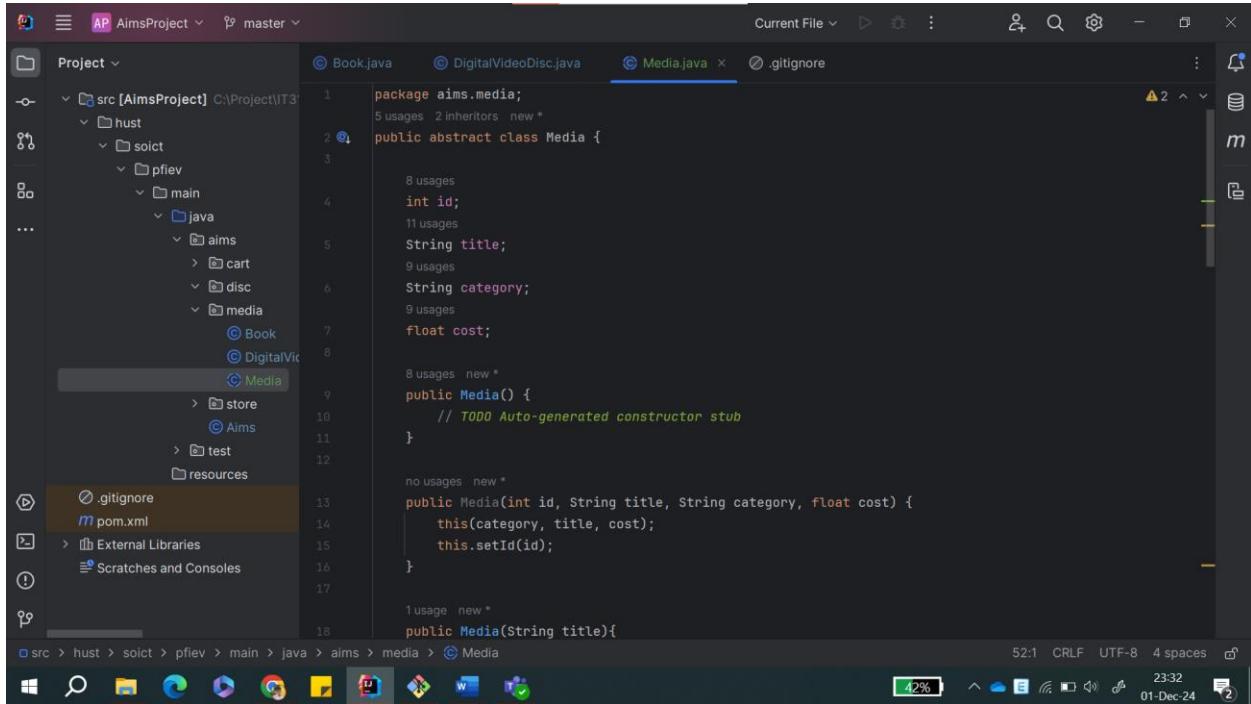
    public void removeAuthor(String authorName) {
        for(String author: authors) {
            if(author.equals(authorName)) {
                authors.remove(authorName);
                System.out.println("Author is removed!");
                return ;
            }
        }
        System.out.println("Author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

```

Hình 1.3: Mã nguồn lớp Book phần 3

## 2. Create the abstract Media class

### 2.1. Mã nguồn lớp Media



```

package aims.media;
public abstract 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){
    }
}

```

Hình 2.1: Mã nguồn lớp Media phần 1

```

1 usage new *
public Media(String title){
    this.setTitle(title);
}

1 usage new *
public Media(String title, String category, float cost){
    this.title = title;
    this.setCategory(category);
    this.setCost(cost);
}

1 usage 1 override new *
public int getId() {
    return id;
}

1 usage new *
public void setId(int id) {
    this.id = id;
}

9 usages 1 override new *
public String getTitle() {
    return title;
}

2 usages 1 override new *
public void setTitle(String title) {
    this.title = title;
}

```

Hình 2.2: Mô hình lớp Media phần 2

```

2 usages 1 override new *
public void setTitle(String title) {
    this.title = title;
}

1 usage 1 override new *
public String getCategory() {
    return category;
}

1 usage 1 override new *
public void setCategory(String category) {
    this.category = category;
}

2 usages 1 override new *
public float getCost() {
    return cost;
}

1 usage 1 override new *
public void setCost(float cost) {
    this.cost = cost;
}

```

Hình 2.3: Mô hình lớp Media phần 3

## 2.2. Mã nguồn cải tiến lớp DigitalVideoDisc

```

1 package aims.media;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class DigitalVideoDisc extends Media {
7     private String director;
8     private int length;
9     private static int nbDigitalVideoDiscs=0;
10
11    public int getId() { return this.id; }
12
13    public DigitalVideoDisc(String title) {
14        super();
15        this.title = title;
16        nbDigitalVideoDiscs++;
17        this.id = nbDigitalVideoDiscs;
18    }
19
20    public DigitalVideoDisc(String title, String category, float cost) {
21        super();
22        this.title = title;
23        this.category = category;
24        this.cost = cost;
25    }
26
27    public void play() {
28        System.out.println("Playing " + title);
29    }
30
31    public void stop() {
32        System.out.println("Stopping " + title);
33    }
34
35    public void pause() {
36        System.out.println("Pausing " + title);
37    }
38
39    public void resume() {
40        System.out.println("Resuming " + title);
41    }
42
43    public void fastForward() {
44        System.out.println("Fast Forwarding " + title);
45    }
46
47    public void rewind() {
48        System.out.println("Rewinding " + title);
49    }
50
51    public void eject() {
52        System.out.println("Ejecting " + title);
53    }
54
55    public void insert() {
56        System.out.println("Inserting " + title);
57    }
58
59    public void eject() {
60        System.out.println("Ejecting " + title);
61    }
62
63    public void eject() {
64        System.out.println("Ejecting " + title);
65    }
66
67    public void eject() {
68        System.out.println("Ejecting " + title);
69    }
70
71    public void eject() {
72        System.out.println("Ejecting " + title);
73    }
74
75    public void eject() {
76        System.out.println("Ejecting " + title);
77    }
78
79    public void eject() {
80        System.out.println("Ejecting " + title);
81    }
82
83    public void eject() {
84        System.out.println("Ejecting " + title);
85    }
86
87    public void eject() {
88        System.out.println("Ejecting " + title);
89    }
90
91    public void eject() {
92        System.out.println("Ejecting " + title);
93    }
94
95    public void eject() {
96        System.out.println("Ejecting " + title);
97    }
98
99    public void eject() {
100       System.out.println("Ejecting " + title);
101   }
102
103  }

```

Hình 2.4: Mã nguồn cải tiến lớp DigitalVideoDisc

Lớp DigitalVideoDisc kế thừa từ lớp trừu tượng Media, một số thuộc tính được xóa đi.

## 2.3. Mã nguồn cải tiến lớp Book

```

1 package aims.media;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Book extends Media {
7     private List<String> authors = new ArrayList<String>();
8
9     public Book() {
10
11    }
12
13    public Book(String title) {
14        this.title = title;
15    }
16
17    public Book(String title, String category, float cost) {
18        this.title = title;
19        this.category = category;
20        this.cost = cost;
21    }
22
23    public void play() {
24        System.out.println("Playing " + title);
25    }
26
27    public void stop() {
28        System.out.println("Stopping " + title);
29    }
30
31    public void pause() {
32        System.out.println("Pausing " + title);
33    }
34
35    public void resume() {
36        System.out.println("Resuming " + title);
37    }
38
39    public void fastForward() {
40        System.out.println("Fast Forwarding " + title);
41    }
42
43    public void rewind() {
44        System.out.println("Rewinding " + title);
45    }
46
47    public void eject() {
48        System.out.println("Ejecting " + title);
49    }
50
51    public void insert() {
52        System.out.println("Inserting " + title);
53    }
54
55    public void eject() {
56        System.out.println("Ejecting " + title);
57    }
58
59    public void eject() {
60        System.out.println("Ejecting " + title);
61    }
62
63    public void eject() {
64        System.out.println("Ejecting " + title);
65    }
66
67    public void eject() {
68        System.out.println("Ejecting " + title);
69    }
70
71    public void eject() {
72        System.out.println("Ejecting " + title);
73    }
74
75    public void eject() {
76        System.out.println("Ejecting " + title);
77    }
78
79    public void eject() {
80        System.out.println("Ejecting " + title);
81    }
82
83    public void eject() {
84        System.out.println("Ejecting " + title);
85    }
86
87    public void eject() {
88        System.out.println("Ejecting " + title);
89    }
90
91    public void eject() {
92        System.out.println("Ejecting " + title);
93    }
94
95    public void eject() {
96        System.out.println("Ejecting " + title);
97    }
98
99    public void eject() {
100       System.out.println("Ejecting " + title);
101   }
102
103  }

```

Hình 2.5: Mã nguồn cải tiến lớp Book

Tương tự, Lớp Book kế thừa từ lớp trừu tượng Media, một số thuộc tính được xóa đi.

### 3. Create the CompactDisc class

#### 3.1. Tao lớp Disc

```

package aims.media;

public class Disc extends Media{
    String director;
    int length;

    public Disc() {
    }

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

    public Disc(String title, String category, String director, int length, float cost){
        this(director,category,title,cost);
        this.setLength(length);
    }
}

```

Hình 3.1: Mã nguồn lớp Disc phần 1

```

public Disc(String director, int length) {
    super();
    this.setDirector(director);
    this.setLength(length);
}

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

public String getDirector() {
    return director;
}

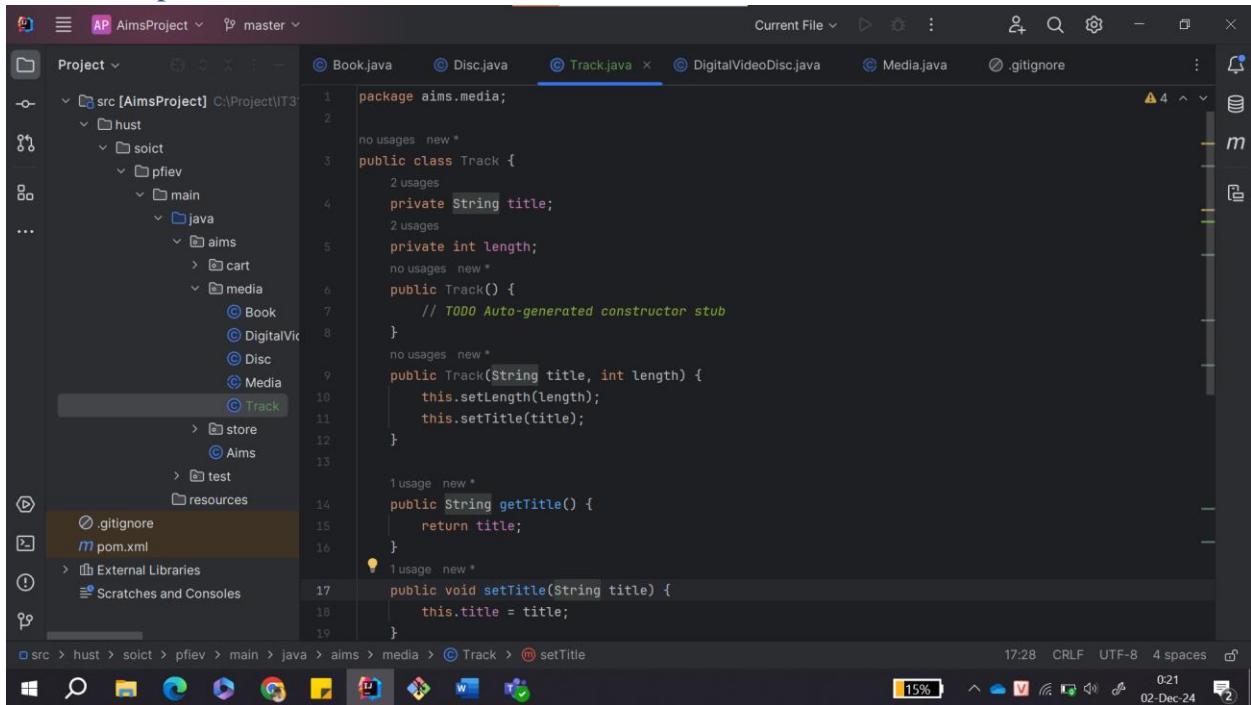
public void setDirector(String director) {
    this.director = director;
}

public int getLength() {
    return length;
}

```

Hình 3.2: Mã nguồn lớp Disc phần 2

### 3.2. Tao lớp Track



```

package aims.media;

public class Track {
    private String title;
    private int length;

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

    public Track(String title, int length) {
        this.setTitle(title);
        this.setLength(length);
    }

    public String getTitle() {
        return title;
    }

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

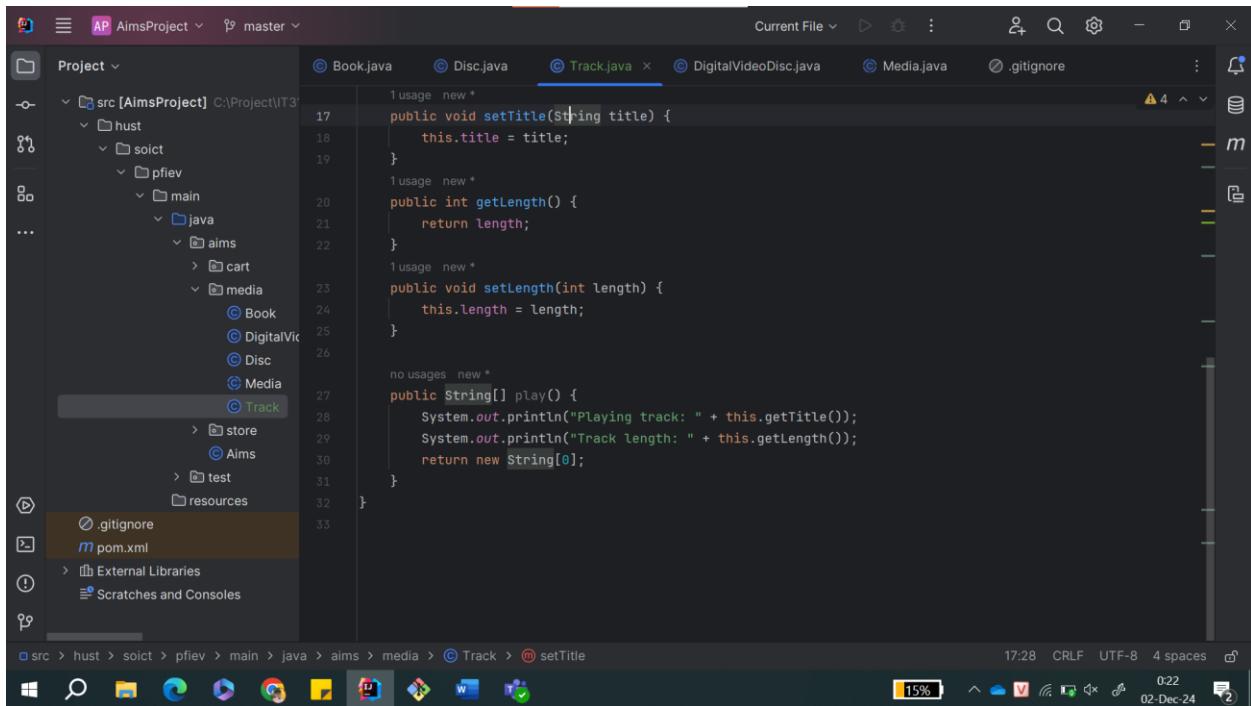
    public int getLength() {
        return length;
    }

    public void setLength(int length) {
        this.length = length;
    }

    public String[] play() {
        System.out.println("Playing track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
        return new String[0];
    }
}

```

Hình 3.3: Mã nguồn lớp Track phần 1



```

public String[] play() {
    System.out.println("Playing track: " + this.getTitle());
    System.out.println("Track length: " + this.getLength());
    return new String[0];
}

```

Hình 3.4: Mã nguồn lớp Track phần 2

### 3.3. Tạo lớp CompactDisc

The screenshot shows an IDE interface with a dark theme. The left sidebar displays a project structure for 'AimsProject' with packages like 'hust' and 'soict'. The 'src' folder contains Java files: 'Book.java', 'Disc.java', 'Track.java', 'CompactDisc.java' (which is currently selected), and 'DigitalVideoDisc.java'. Below these are 'Media.java' and 'Media.java'. The 'test' and 'resources' folders are also listed. The bottom of the sidebar shows file navigation history and system status. The main editor area shows the code for the 'CompactDisc' class, which extends the 'Disc' class. It includes constructors for 'CompactDisc' and 'CompactDisc(String title, String artist)'. The code uses an ArrayList to store tracks. The status bar at the bottom indicates the current file is 'CompactDisc.java', the time is 12:51, and the date is 02-Dec-24.

```
package aims.media;
import java.util.ArrayList;

public class CompactDisc extends Disc {
    private String artist;
    private ArrayList<Track> tracks = new ArrayList<Track>();

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

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

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

Hình 3.5: Mã nguồn lớp *CompactDisc* phần 1

The screenshot shows a Java project structure in the left sidebar. The current file is `CompactDisc.java`, which contains the following code:

```
17
18     usage new *
19     public CompactDisc(String title, String category, float cost){
20         this(title);
21         this.setCategory(category);
22         this.setCost(cost);
23     }
24
25     no usages new *
26     public CompactDisc(String director, String category, String title, float cost){
27         this(category,title,cost);
28         this.setDirector(director);
29     }
30
31     no usages new *
32     public void addTrack(Track newTrack) {
33         if(this.tracks.contains(newTrack)) {
34             System.out.println("Track already exists!");
35             return ;
36         }
37         this.tracks.add(newTrack);
38         System.out.println("Track is added!");
39     }
40
41     no usages new *
42     public void removeTrack(Track track) {
43         if(this.tracks.contains(track)) {
```

The status bar at the bottom shows the file path as `src > hust > soict > pfiev > main > java > aims > media > CompactDisc`, and the system status includes battery level (8%), signal strength, and the date (02-Dec-24).

Hình 3.6: Mă nguồn lớp *CompactDisc* phần 2

```

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!");
}

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;
}

@Override
public String toString() {
    return "CD - " + this.getTitle() + " - " + this.getCategory() + " - " + this.getDirector() + " - " +
}

```

Hình 3.7: Mã nguồn lớp CompactDisc phần 3

```

    return sum;
}

no usages new
public String getArtist() {
    return artist;
}

1usage new
public void setArtist(String artist) {
    this.artist = artist;
}

new *
@Override
public String toString() {
    return "CD - " + this.getTitle() + " - " + this.getCategory() + " - " + this.getDirector() + " - " +
}

```

Hình 3.8: Mã nguồn lớp CompactDisc phần 4

## 4. Create the Playable interface

### 4.1. Mã nguồn giao diện Playable

```

package aims.media;

public interface Playable {
    public void play();
}

```

Hình 4.1: Mã nguồn giao diện Playable

### 4.2. Triển khai giao diện Playable ở các lớp khác

```

package aims.media;

public class DigitalVideoDisc extends Disc implements Playable {
    private String director;
    private int length;
    private static int nbDigitalVideoDiscs=0;

    public int getId() { return this.id; }

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

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

```

Hình 4.2: Mã nguồn cài tiến lớp DigitalVideoDisc phần 1

```

src [AimsProject] C:\Project\IT3103.744
  src
    hust
      soict
        pfiev
          main
            java
              aims
                cart
                media
                  Book
                  CompactDisc
                  DigitalVideoDisc
                  Disc
                  Media
                  Playable
                  Track
                store
                Aims
              test
            resources
          .gitignore
        pom.xml
      External Libraries
      Scratchers and GrepLocs
  src > hust > soict > pfiev > main > java > aims > media > DigitalVideoDisc > play
  94:6 CRLF UTF-8 4 spaces
  22% 055 02-Dec-24 055 02-Dec-24

  package aims.media;
  import java.util.ArrayList;

  public class DigitalVideoDisc implements Playable {
    private float cost;
    private String title;
    private int length;

    public void setLength(int length) { this.length = length; }

    public float getCost() { return cost; }

    public void setCost(float cost) { this.cost = cost; }

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

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

```

Hình 4.3: Mã nguồn cài tiến lớp DigitalVideoDisc phần 2

```

src [AimsProject] C:\Project\IT3103.744
  src
    hust
      soict
        pfiev
          main
            java
              aims
                cart
                media
                  Book
                  CompactDisc
                  DigitalVideoDisc
                  Disc
                  Media
                  Playable
                  Track
                store
                Aims
              test
            resources
          .gitignore
        pom.xml
      External Libraries
      Scratchers and GrepLocs
  src > hust > soict > pfiev > main > java > aims > media > CompactDisc > play
  72:6 CRLF UTF-8 4 spaces
  23% 056 02-Dec-24 056 02-Dec-24

  package aims.media;
  import java.util.ArrayList;

  public class CompactDisc extends Disc implements Playable {
    private String artist;
    private ArrayList<Track> tracks = new ArrayList<Track>();

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

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

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

```

Hình 4.4: Mã nguồn cài tiến lớp CompactDisc phần 1

```

public class CompactDisc implements Playable {
    private String artist;
    private List<Track> tracks;

    public CompactDisc(String artist) {
        this.artist = artist;
    }

    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() + " - ");
    }
}

```

Hình 4.5: Mã nguồn cài tiến lớp CompactDisc phần 2

```

package aims.media;

public class Track implements Playable {
    private String title;
    private int length;

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

    public Track(String title, int length) {
        this.setTitle(title);
        this.setLength(length);
    }

    public String getTitle() {
        return title;
    }

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

```

Hình 4.6: Mã nguồn cài tiến lớp Track phần 1

```

16
17
18    public void setTitle(String title) {
19        this.title = title;
20    }
21
22    public int getLength() {
23        return length;
24    }
25
26    public void setLength(int length) {
27        this.length = length;
28    }
29
30    public void play() {
31        System.out.println("Playing track: " + this.getTitle());
32        System.out.println("Track length: " + this.getLength());
33    }

```

Hình 4.7: Mã nguồn cài tiến lớp Track phần 2

## 5. Update the Cart class to work with Media

```

1 package aims.cart;
2
3 import aims.media.Media;
4 import java.util.ArrayList;
5
6 public class Cart {
7     private ArrayList<Media> itemsOrdered = new ArrayList<Media>();
8
9     public void addMedia(Media d) {
10         itemsOrdered.add(d);
11         System.out.println("The media has been added!");
12     }
13
14     public void addMedia(Media[] dList) {
15         for (Media disc: dList) {
16             addMedia(disc);
17         }
18     }
19
20     public void addMedia(Media d1, Media d2) {
21         addMedia(d1);
22         addMedia(d2);
23     }

```

Hình 5.1: Mã nguồn cài tiến lớp Cart phần 1

```

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 ;
    }
}

2 usages ▲ lephuchung *
public float totalCost() {
    float total = 0;
    for(int i = 0; i < itemsOrdered.size(); i++) {
        total += itemsOrdered.get(i).getCost();
    }
    return total;
}

1 usage ▲ lephuchung
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("*****CART*****");
}

```

Hình 5.2: Mã nguồn cài tiến lớp Cart phần 2

```

no usages ▲ lephuchung *
public ArrayList<Media> getItemsOrdered() {
    return itemsOrdered;
}

1 usage ▲ lephuchung *
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!");
}

1 usage ▲ lephuchung *
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!");
}

```

Hình 5.3: Mã nguồn cài tiến lớp Cart phần 3

## 6. Update the Store class to work with Media

```

1 package aims.store;
2
3 import aims.media.Book;
4 import aims.media.CompactDisc;
5 import aims.media.DigitalVideoDisc;
6 import aims.media.Media;
7
8 import java.util.ArrayList;
9
10 public class Store {
11     private ArrayList<Media> itemsInStore = new ArrayList<Media>();
12     private int nbItems = 0;
13
14     public void addMedia(Media d) {
15         itemsInStore.add(d);
16         nbItems += 1;
17         System.out.println("The media is added to store!");
18     }
19
20     public void removeMedia(Media d) {
21         try {
22             itemsInStore.remove(d);
23             nbItems -= 1;
24             System.out.println("The media is removed from store!");
25         } catch (Exception e) {
26             System.out.println("Media doesn't exist!");
27             return ;
28         }
29     }
30
31     public void print() {
32         int num = 1;
33         System.out.println("*****STORE*****");
34         for(Media media: this.itemsInStore) {
35             if (media instanceof DigitalVideoDisc) {
36                 DigitalVideoDisc dvd = (DigitalVideoDisc) media;
37                 System.out.println(num + ". " + "DVD" + ". " + dvd.getTitle() + ". " + dvd.getCategory());
38             }
39             else if (media instanceof CompactDisc) {
40                 CompactDisc cd = (CompactDisc) media;
41                 System.out.println(num + ". " + "CD" + ". " + cd.getTitle() + ". " + cd.getCategory());
42             }
43             else if (media instanceof Book) {
44                 Book book = (Book) media;
45                 System.out.println(num + ". " + "Book" + ". " + book.getTitle() + ". " + book.getCategory());
46             }
47             num++;
48         }
49     }
50 }

```

Hình 6.1: Mã nguồn cài tiến lớp Store phần I

```

51
52     public void print() {
53         int num = 1;
54         System.out.println("*****STORE*****");
55         for(Media media: this.itemsInStore) {
56             if (media instanceof DigitalVideoDisc) {
57                 DigitalVideoDisc dvd = (DigitalVideoDisc) media;
58                 System.out.println(num + ". " + "DVD" + ". " + dvd.getTitle() + ". " + dvd.getCategory());
59             }
60             else if (media instanceof CompactDisc) {
61                 CompactDisc cd = (CompactDisc) media;
62                 System.out.println(num + ". " + "CD" + ". " + cd.getTitle() + ". " + cd.getCategory());
63             }
64             else if (media instanceof Book) {
65                 Book book = (Book) media;
66                 System.out.println(num + ". " + "Book" + ". " + book.getTitle() + ". " + book.getCategory());
67             }
68             num++;
69         }
70     }
71 }

```

Hình 6.2: Mã nguồn cài tiến lớp Store phần 2

```

38     System.out.println(num + ". " + "CD" + ". " + cd.getTitle() + ". " + cd.get
39 }
40 else if (media instanceof Book) {
41     Book book = (Book) media;
42     System.out.println(num + ". " + "Book" + ". " + book.getTitle() + ". " + book.get
43 }
44 num++;
45 }
46 System.out.println("*****");
47 }
48 4 usages  lephuchung *
49 public int getNumItems() {
50     return nbItems;
51 }
52 no usages  new *
53 public ArrayList<Media> getItemsInStore() {
54     return itemsInStore;
55 }

```

Hình 6.3: Mã nguồn cải tiến lớp Store phần 3

## 7. Constructor of whole classes and parent classes

Sơ đồ lớp được cập nhật, sơ đồ cuối lab04 sẽ được cập nhật tại mục 12.

- Lớp Media có các lớp con là Disc và Book
- Lớp Disc có các lớp con là DigitalVideoDisc và CompactDisc

## 8. Unique item in a list

```

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;
}

```

Hình 8.1: Mã nguồn cập nhật lớp Media cho phương thức equals

```

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

@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;
}

```

Hình 8.2: Mã nguồn cập nhật lớp Track cho phương thức equals

## 9. Polymorphism with `toString()` method

```

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!");
}

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

```

Hình 9.1: Phương thức `toString()` lớp `Book`

```

1 usage ʌ lephuchung
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();
    }
}

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

```

Hình 9.2: Phương thức `toString()` lớp `CompactDisc`

```

public float getCost() { return cost; }

public void setCost(float cost) { this.cost = cost; }

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

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

```

Hình 9.3: Phương thức `toString()` lớp `DigitalVideoDisc`

```

anOrder.addMedia(dvd2, dvd3);
System.out.println("-----");
System.out.println("The total cost is: ");
System.out.println(anOrder.totalCost());
System.out.println("-----");
System.out.println("check id dvd2: " + dvd2.getId());
CompactDisc cd1 = new CompactDisc( title: "CDhehe", category: "chill", cost: 100);
Book book1 = new Book( title: "Bookhehe", category: "chill", cost: 100);
System.out.println("-----");
System.out.println(dvd1.toString());
System.out.println(cd1.toString());
System.out.println(book1.toString());

```

Output:

```

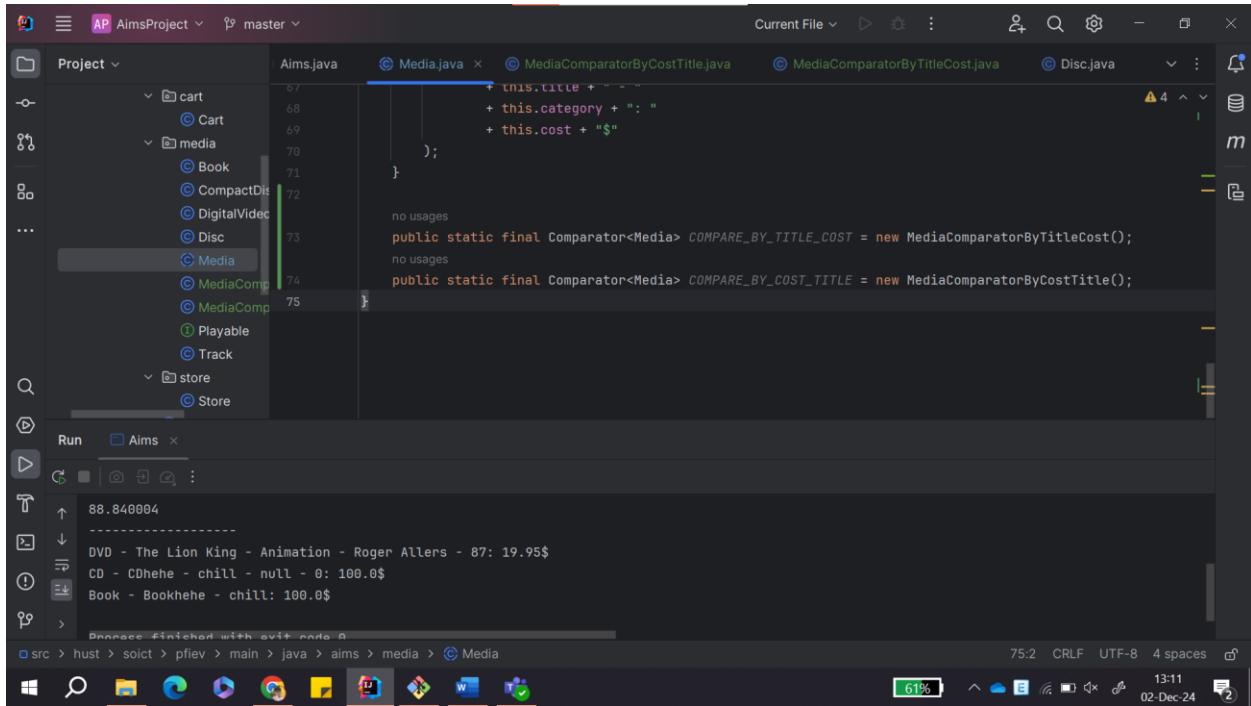
88.840004
-----
DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
CD - CDhehe - chill - null - 0: 100.0$
Book - Bookhehe - chill: 100.0$

Process finished with exit code 0

```

Hình 9.4: Kết quả demo đa hình trong hướng đối tượng

## 10. Sort media in the cart



```

package aims.media;

import java.util.Comparator;

public class Media {
    private String title;
    private String category;
    private float cost;

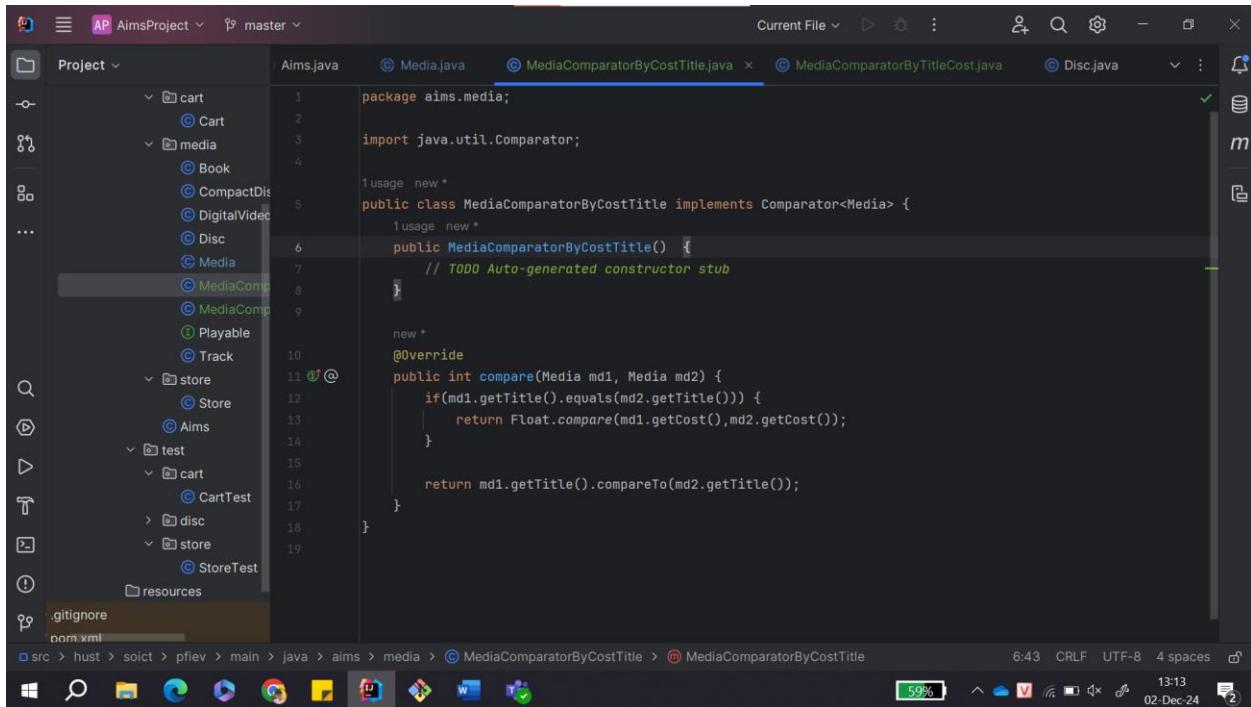
    public Media(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

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

class MediaComparatorByTitleCost implements Comparator<Media> {
    @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());
    }
}

```

Hình 10.1: Thêm các thuộc tính dùng để so sánh vào lớp Media



```

package aims.media;

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> {
    @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());
    }
}

```

Hình 10.2: Phương thức MediaComparatorByCostTitle()

```

package aims.media;

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {
    @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());
    }
}

```

Hình 10.3: Phương thức MediaComparatorByTitleCost()

## 11. Create a complete console application in the Aims class

Giao diện dòng lệnh ban đầu:

```

The media is added to store!
AIMS:
-----
1. View store
2. Update store
3. See the current cart
-----
Please choose a number: 0-1-2-3

```

Hình 11.1: Giao diện dòng lệnh ban đầu của AIMS

Chọn chức năng view store:

```

3. See the current cart
-----
Please choose a number: 0-1-2-3
1
*****
1. DVD. The Lion King. Animation. Roger Allers. 87. 19.95
2. DVD. Star Wars. Science Fiction. Geoger Lucas. 87. 24.95
3. DVD. Aladin. Animation. null. 0. 18.99
4. Book. Bookhehe. chill. null. 0. 100.0
5. CD. CDhehe. chill. null. 0. 100.0
*****
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
|
```

Hình 11.2: Giao diện chức năng xem sản phẩm trong cửa hàng

Chọn chức năng xem chi tiết sản phẩm:

```

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
1
Enter the title of the media
Bookhehe

Book - Bookhehe - chill: 100.0$
Options:
-----
1. Add to cart
2. Play
0. Back
-----
Please choose a number: 0-1-2
|
```

Hình 11.3: Giao diện chức năng xem chi tiết sản phẩm

Chọn chức năng thêm sản phẩm vào giỏ hàng:

```

Please choose a number: 0-1-2-3-4
2
Enter the title of the media: CDhehe
The media has been added!
*****
1. DVD. The Lion King. Animation. Roger Allers. 87. 19.95
2. DVD. Star Wars. Science Fiction. Geoger Lucas. 87. 24.95
3. DVD. Aladin. Animation. null. 0. 18.99
4. Book. Bookhehe. chill. 100.0
5. CD. CDhehe. chill. null. 0. 100.0
*****
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

```

Hình 11.4: Giao diện chức năng thêm sản phẩm vào giỏ hàng

Chọn chức năng phát sản phẩm:

```

3
Enter the title of the media: The Lion King
Playing DVD: The Lion King
DVD length: 87
*****
1. DVD. The Lion King. Animation. Roger Allers. 87. 19.95
2. DVD. Star Wars. Science Fiction. Geoger Lucas. 87. 24.95
3. DVD. Aladin. Animation. null. 0. 18.99
4. Book. Bookhehe. chill. 100.0
5. CD. CDhehe. chill. null. 0. 100.0
*****
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

```

Hình 11.5: Giao diện chức năng phát sản phẩm

Chọn chức năng xem giỏ hàng hiện tại:

```

4. See current cart
0. Back
-----
Please choose a number: 0-1-2-3-4
4
*****
1. Book - Bookhehe - chill: 100.0$
2. CD - CDhehe - chill - null - 0: 100.0$
Total cost: 200.0$
*****
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5
|
```

Hình 11.6: Giao diện chức năng xem giỏ hàng

Chọn chức năng sắp xếp giỏ hàng theo tiêu đề:

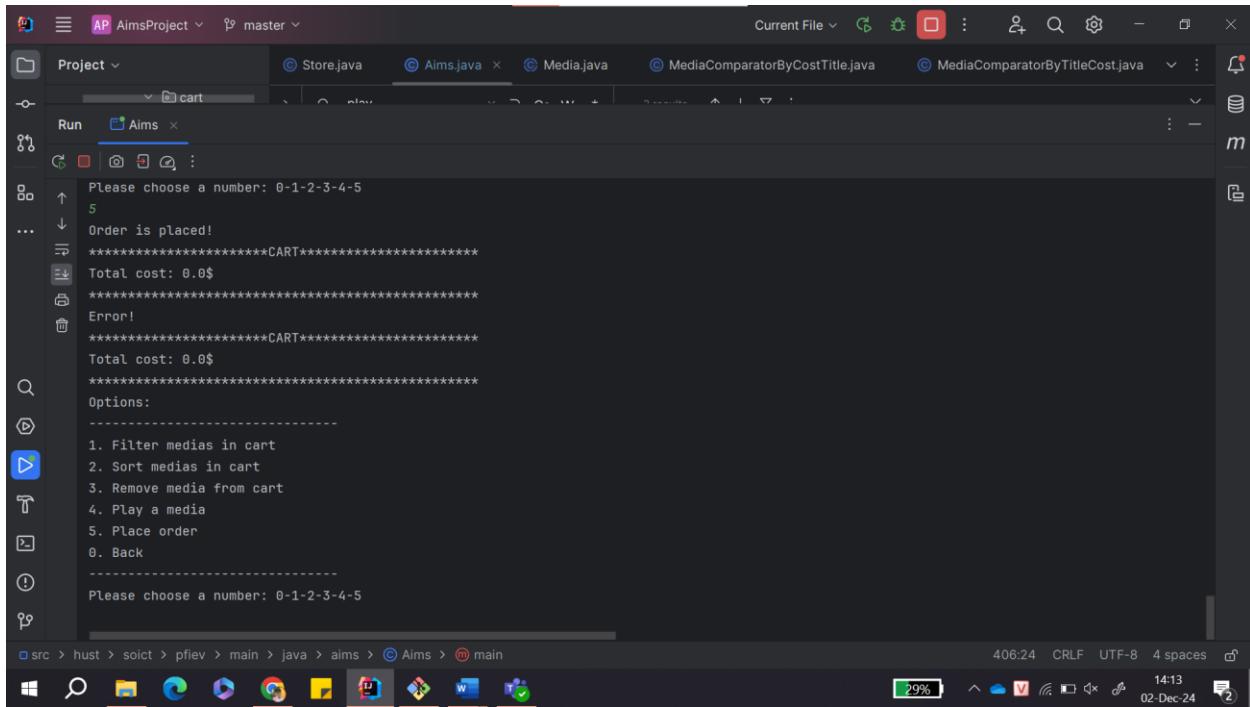
```

1. Book - Bookhehe - chill: 100.0$
2. CD - CDhehe - chill - null - 0: 100.0$
Total cost: 200.0$
*****
1. Book - Bookhehe - chill: 100.0$
2. CD - CDhehe - chill - null - 0: 100.0$
Total cost: 200.0$
*****
Options:
-----
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
-----
Please choose a number: 0-1-2-3-4-5
|
```

Hình 11.7: Giao diện chức năng sắp xếp giỏ hàng

Vì trong trường hợp này, các sản phẩm đã ngẫu nhiên rơi vào trường hợp sắp xếp theo đúng thứ tự, nên việc sắp xếp không gây ra khác biệt. Điều này tương tự sắp xếp theo id.

Chọn chức năng đặt hàng:



The screenshot shows a Java application window titled 'AimsProject' running on IntelliJ IDEA. The code editor displays a class named 'Aims'. The terminal output pane shows the following interaction:

```

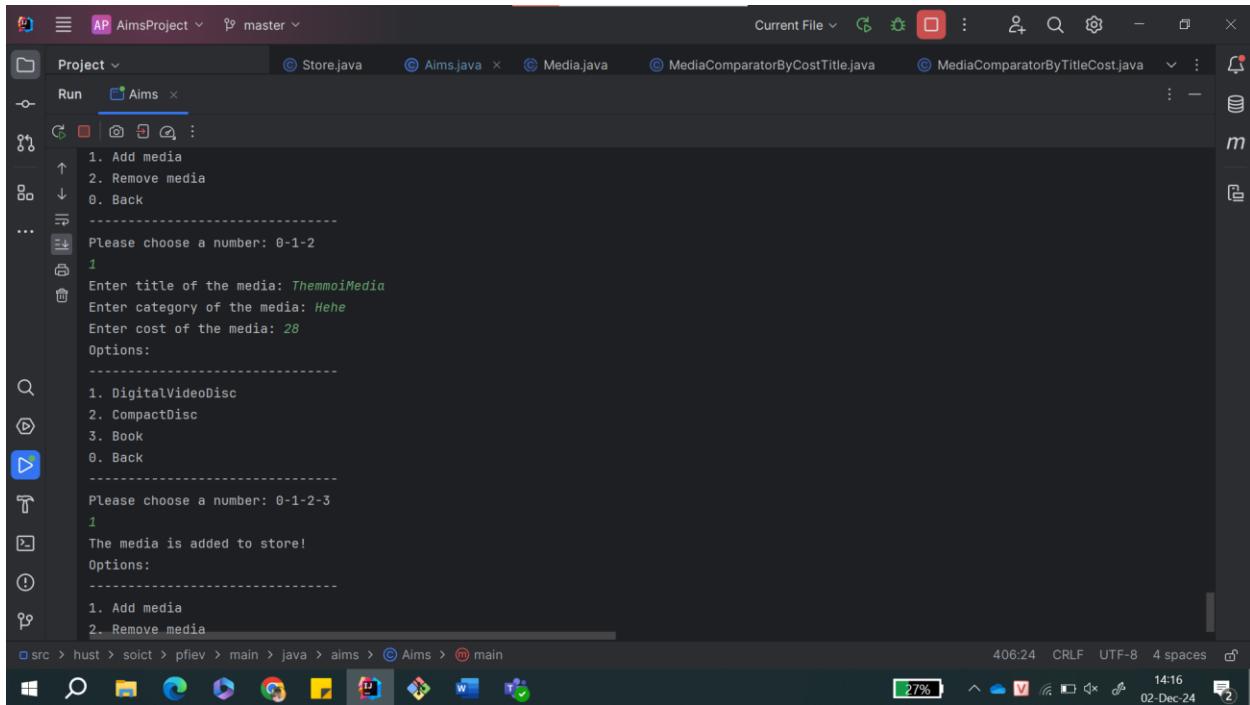
Please choose a number: 0-1-2-3-4-5
5
Order is placed!
Total cost: 0.0$
Error!
Total cost: 0.0$
Options:
1. Filter medias in cart
2. Sort medias in cart
3. Remove media from cart
4. Play a media
5. Place order
0. Back
Please choose a number: 0-1-2-3-4-5

```

The status bar at the bottom indicates the file is 406:24, CRLF, UTF-8, 4 spaces, and the system date and time are 02-Dec-24 14:13.

Hình 11.8: Giao diện chức năng đặt hàng

Giỏ hàng đã được làm trống. Chọn chức năng thêm sản phẩm vào store:



The screenshot shows the same Java application window. The terminal output pane shows the following interaction:

```

1. Add media
2. Remove media
0. Back
Please choose a number: 0-1-2
1
Enter title of the media: ThemmoiMedia
Enter category of the media: Hehe
Enter cost of the media: 28
Options:
1. DigitalVideoDisc
2. CompactDisc
3. Book
0. Back
Please choose a number: 0-1-2-3
1
The media is added to store!
Options:
1. Add media
2. Remove media

```

The status bar at the bottom indicates the file is 406:24, CRLF, UTF-8, 4 spaces, and the system date and time are 02-Dec-24 14:16.

Hình 11.9: Giao diện chức năng thêm sản phẩm vào cửa hàng

## Kết quả:

```

Current File: Aims.java
Run: Aims
Project: AimsProject
File: master

3. See the current cart
Please choose a number: 0-1-2-3
1
*****
*****STORE*****
1. DVD. The Lion King. Animation. Roger Allers. 87. 19.95
2. DVD. Star Wars. Science Fiction. Geoger Lucas. 87. 24.95
3. DVD. Aladin. Animation. null. 0. 18.99
4. Book. Bookhehe. chill. 100.0
5. CD. CDhehe. chill. null. 0. 100.0
6. DVD. ThemmoiMedia. Hehe. null. 0. 28.0
*****
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

```

Hình 11.10: Kết quả thêm sản phẩm vào cửa hàng

Chọn chức năng xóa hàng khỏi store:

```

Current File: Aims.java
Run: Aims
Project: AimsProject
File: master

3. See the current cart
Please choose a number: 0-1-2-3
2
Options:
1. Add media
2. Remove media
0. Back
Please choose a number: 0-1-2
2
Enter the title of the media:ThemmoiMedia
The media is removed from store!
Options:
1. Add media
2. Remove media
0. Back
Please choose a number: 0-1-2

```

Hình 11.11: Giao diện chức năng thêm sản phẩm vào cửa hàng

Kết quả:

```

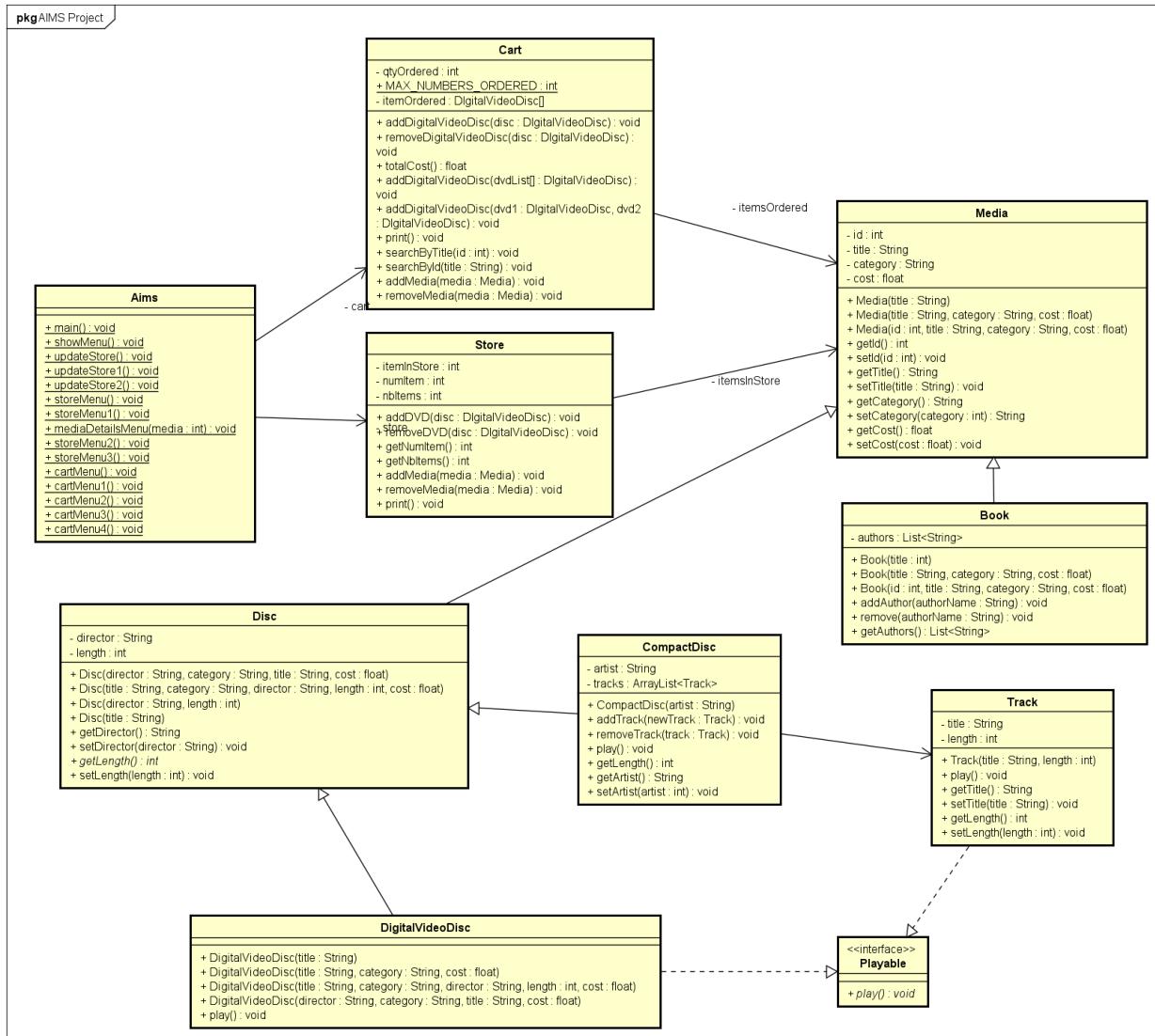
AP AimsProject master
Project Run Aims
Run Aims x
2. Update store
3. See the current cart
-----
Please choose a number: 0-1-2-3
1
*****
*****STORE*****
1. DVD. The Lion King. Animation. Roger Allers. 87. 19.95
2. DVD. Star Wars. Science Fiction. Geoger Lucas. 87. 24.95
3. DVD. Aladin. Animation. null. 0. 18.99
4. Book. Bookhehe. chill. 100.0
5. CD. CDhehe. chill. null. 0. 100.0
*****
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

```

Hình 11.12: Kết quả xóa sản phẩm khỏi cửa hàng

## 12. Class diagram

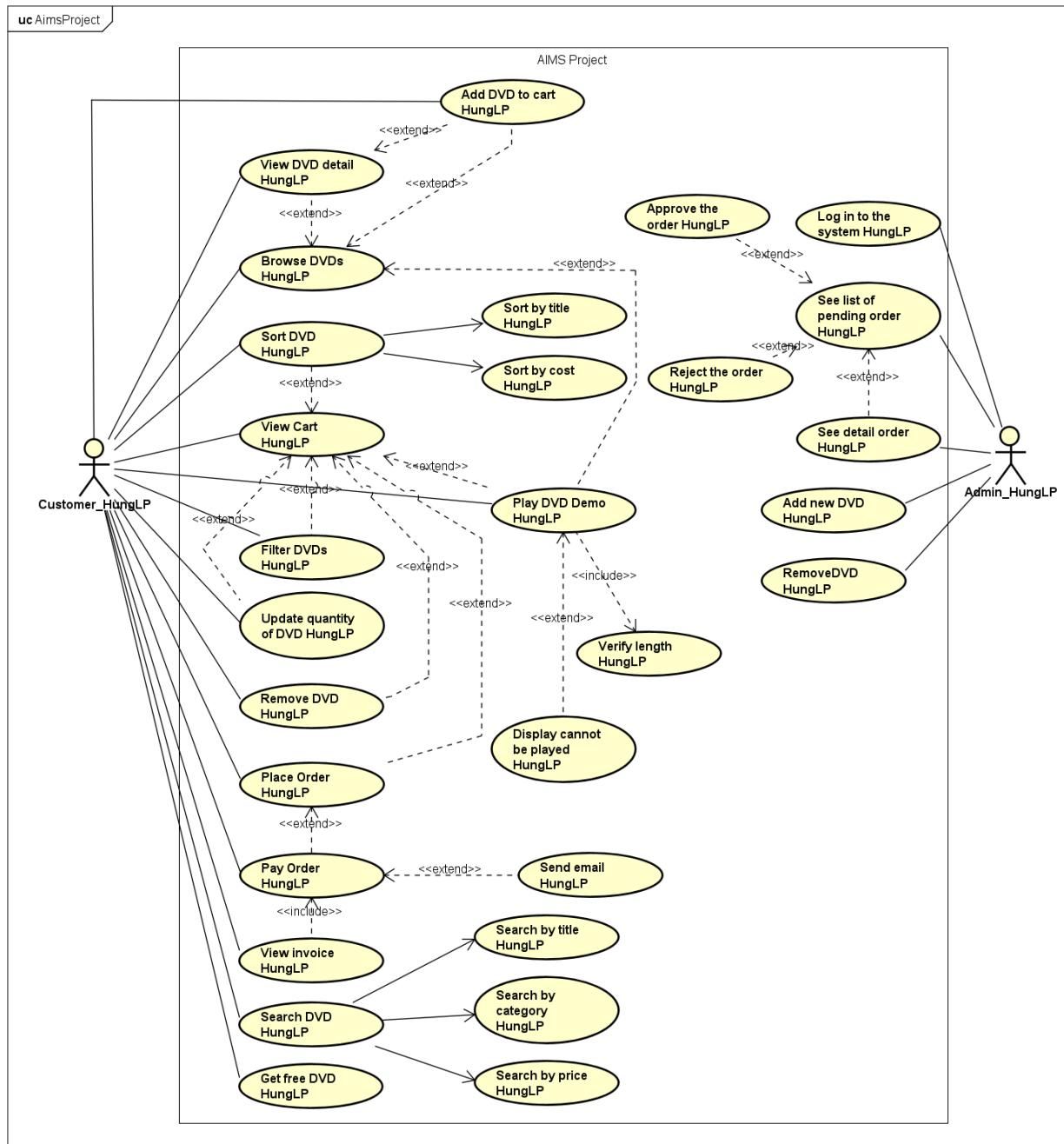
Dưới đây là class diagram đã cập nhật:



Hình 12.1: Class diagram AIMS Project

### 13. Usecase diagram

Usecase không có gì cần cập nhật thêm:



Hình 13.1: Usecase diagram AIMS Project

## 14. Answer questions

- What class should implement the Comparable interface?

The Media class should implement the Comparable interface because it is the base class for all media types (e.g., DVDs, CDs) and holds the shared properties title and cost.

- How should you implement the compareTo() method to reflect the ordering that we want?

If the primary sorting rule is "by title then cost", the compareTo() method should:

- Compare the media items by their title in alphabetical order.
- If the titles are the same, compare the cost in descending order.

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

No, the Comparable interface only allows a single natural ordering to be defined, implemented via the compareTo() method. If two or more sorting rules are required, Comparator is a better choice because it allows for defining multiple custom comparison logic independently of the compareTo() method.

4. How would you modify your code to allow DVDs to have a different ordering rule?

If DVDs need a unique sorting rule (e.g., by title, then decreasing length, then cost), you could override the compareTo() method in a subclass specifically for DVDs.