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

BÁO CÁO THỰC HÀNH
IT3103-744527-2024.1

BÀI THỰC HÀNH - LAB04

Họ tên SV: Lý Công Tiến
MSSV: 20225934
Lớp: Việt-Nhật 04
GVHD: Lê Thị Hoa
HTGD: Đặng Mạnh Cường

Hà Nội 9/2024

BÁO CÁO THỰC HÀNH LAB4 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

1 Create the Book class

```java
package hust.soict.dsai.aims.media;
import java.util.List;
import java.util.ArrayList;
public class BookTienLC extends MediaTienLC {
    private List<String> authors = new ArrayList<String>();
    public BookTienLC(String title, String category, float cost){
        super(title, category, cost);
    }
    public void addAuthorTienLC(String authorName){
        if(authors.contains(authorName)){
            System.out.println("Author is already in the list.");
        }
        else{
            authors.add(authorName);
            System.out.println("added new Author successfully.");
        }
    }
    public void removeAuthorTienLC(String authorName){
        if(!authors.contains(authorName)){
            System.out.println("Not found the author.");
        }
        else{
            authors.remove(authorName);
            System.out.println("Deleted Author successfully.");
        }
    }
```

Figure 1.1: Book Class 1

```java
    @Override
    public int getLength(){
        return -1;
    }
    @Override
    public String getDirector(){
        return "Not Applicable";
    }
}
```

Figure 1.2: Book Class 2

2. Creating the abstract Media class

Đây sẽ là lớp cha để các lớp DigitalVideoDisc, Book kế thừa.

```
package hust.soict.dsai.aims.media;
public abstract class MediaTienLC {
    private static int idCounter = 0;
    private int id;
    private String title;
    private String category;
    private float cost;
    //Constructor
    public MediaTienLC(String title, String category, float cost){
        this.id = ++idCounter;
        this.title = title;
        this.category = category;
        this.cost = 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;
    }
```

Figure 2.1: Media Class 1

```
    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 obj){
        if(this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        MediaTienLC media = (MediaTienLC) obj;
        return this.title != null && this.title.equals(media.title);
    }
    @Override
    public String toString(){
        return "ID: " + id + "\nTitle: " + title + "\nCategory: " + category + "\nCost: " + cost;
    }
    public abstract int getLength();
    public abstract String getDirector();
}
```

Figure 2.2: Media Class 2

3. Creating the CompactDisc class

3.1 Create the Disc class extending the Media Class

```
package hust.soict.dsai.aims.media;
public class DiscTienLC extends MediaTienLC {
    private int length;
    private String director;
    //Constructor
    public DiscTienLC(String title, String category, float cost, String director, int length){
        super(title, category, cost);
        this.director = director;
        this.length = length;
    }
    @Override
    public int getLength() {
        return length;
    }
    public void setLength(int length){
        this.length = length;
    }
    @Override
    public String getDirector() {
        return director;
    }
    public void setDirector(String director){
        this.director = director;
    }
}
```

Figure 3.1: Disc Class

```java
package hust.soict.dsai.aims.media;
public class DigitalVideoDiscTienLC extends DiscTienLC implements PlayableTienLC {
    @Override
    public void play(){
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
    //Constructor day du
    public DigitalVideoDiscTienLC(String title, String category, String director, int length, float cost){
        super(title, category, cost, director, length);
    }
    public DigitalVideoDiscTienLC(String title, String category, float cost){
        super(title, category, cost, "", 0);
    }
    public DigitalVideoDiscTienLC(String title){
        super(title, "", 0, "", 0);
    }
}
```

Figure 3.2: DigitalVideoDisc Class

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

public class CompactDiscTienLC extends DiscTienLC implements PlayableTienLC {
    private final String artist;
    private final ArrayList<TrackTienLC> tracks;

    //Constructor cua CompactDisc
    public CompactDiscTienLC(String title, String category, float cost, String artist){
        super(title, category, cost, "", 0);
        this.artist = artist;
        this.tracks = new ArrayList<>(); //Khoi tao danh sach tracks
    }
    @Override
    public void play(){
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("Artist: " + this.artist);
        for(TrackTienLC track : tracks){
            track.play();
        }
    }
    //Getter (only artist)
    public String getArtist(){
        return artist;
    }
```

Figure 3.3: CompactDisc Class 1

```
28      public void addTrackTienLC(TrackTienLC track){
29          if(!tracks.contains(track)){
30              tracks.add(track);
31          }
32          else{
33              System.out.println("Track already exists.");
34          }
35      }
36      //remove track
37      public void removeTrackTienLC(TrackTienLC track){
38          if(tracks.contains(track)){
39              tracks.remove(track);
40          }
41          else{
42              System.out.println("Track not found.");
43          }
44      }
45      //total track sum
46      @Override
        public int getLength(){
48          int totalLength = 0;
49          for(TrackTienLC track : tracks){
50              totalLength += track.getLength();
51          }
52          return totalLength;
53      }
54  }
```

Figure 3.4: CompactDisc Class 2

3.2 Create the Track class which models a track on a compact disc and will store information incuding the title and length of the track.

```java
package hust.soict.dsai.aims.media;
import java.util.Objects;
public class TrackTienLC implements PlayableTienLC {
    private final String title;
    private final int length;
    //Constructor
    public TrackTienLC(String title, int length){
        super();
        this.title = title;
        this.length = length;
    }
    @Override
    public void play(){
        System.out.println("Playing track: " + this.title);
        System.out.println("Track length: " + this.length);
    }
    public String getTitle(){
        return title;
    }
    public int getLength(){
        return length;
    }
```

Figure 3.5: Track Class

```java
    @Override
    public int hashCode(){
        return Objects.hash(length, title);
    }
    @Override
    public boolean equals(Object obj) {
        // Kiểm tra đối tượng truyền vào có phải là đối tượng Track hay không
        if (this == obj) return true;  // So sánh tham chiếu
        if (obj == null || getClass() != obj.getClass()) return false;  // Kiểm tra null và kiểu đối tượng
        TrackTienLC track = (TrackTienLC) obj;  // Ép kiểu đối tượng
        return this.title != null && this.title.equals(track.title) && this.length == track.length;  // So sánh title và length
        }
}
```

Figure 3.6: Track Class 2

3.3 Open the CompactDisc class

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

public class CompactDiscTienLC extends DiscTienLC implements PlayableTienLC {
    private final String artist;
    private final ArrayList<TrackTienLC> tracks;

    //Constructor cua CompactDisc
    public CompactDiscTienLC(String title, String category, float cost, String artist){
        super(title, category, cost, "", 0);
        this.artist = artist;
        this.tracks = new ArrayList<>(); //Khoi tao danh sach tracks
    }
    @Override
    public void play(){
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("Artist: " + this.artist);
        for(TrackTienLC track : tracks){
            track.play();
        }
    }
    //Getter (only artist)
    public String getArtist(){
        return artist;
    }
    public void addTrackTienLC(TrackTienLC track){
        if(!tracks.contains(track)){
            tracks.add(track);
        }
        else{
            System.out.println("Track already exists.");
        }
    }
    //remove track
    public void removeTrackTienLC(TrackTienLC track){
        if(tracks.contains(track)){
            tracks.remove(track);
        }
        else{
            System.out.println("Track not found.");
        }
    }
    //total track sum
    @Override
    public int getLength(){
        int totalLength = 0;
        for(TrackTienLC track : tracks){
            totalLength += track.getLength();
        }
        return totalLength;
    }
}
```

Figure 3.7: CompactDisc Class

4 Create the Playable interface

```
2   package hust.soict.dsai.aims.media;
3
I   public interface PlayableTienLC {
I       public void play();
6   }
7
```

Figure 4.1: Playable interface

Implement play() cho các class DigitalVideoDisc, Track, CompactDisc

```
5       @Override
        public void play(){
7           System.out.println("Playing DVD: " + this.getTitle());
8           System.out.println("DVD length: " + this.getLength());
9       }
```

Figure 4.2: Method play() của DigitalVideoDisc

```
13      @Override
        public void play(){
15          System.out.println("Playing track: " + this.title);
16          System.out.println("Track length: " + this.length);
17      }
```

Figure 4.3: Method play() của Track

```
15      @Override
        public void play(){
17          System.out.println("Playing CD: " + this.getTitle());
18          System.out.println("Artist: " + this.artist);
19          for(TrackTienLC track : tracks){
20              track.play();
21          }
22      }
```

Figure 4.4: Method play() của CompactDisc

5 Update the Cart class to work with Media

Lớp Cart bây giờ cần có khả năng tương tác với các đối tượng DVD, CD và Book. Vì các lớp DVD, CD và Book đều kế thừa từ lớp Media, nên thay vì làm việc trực tiếp với từng lớp con, lớp cart chỉ cần giao tiếp với lớp Media là có thể hoạt động được với tất cả.

```
2   package hust.soict.dsai.aims.cart;
3
4   import java.util.ArrayList;
5   import java.util.Collections;
6   import java.util.Comparator;
7   import hust.soict.dsai.aims.media.MediaTienLC;
8
9   public class CartTienLC {
10      public static final int MAX_NUMBERS_ORDERED = 20;
11      private final ArrayList<MediaTienLC> itemsOrdered = new ArrayList<>();
12
13      public void addMediaTienLC(MediaTienLC media){
14          if(itemsOrdered.size() == MAX_NUMBERS_ORDERED){
15              System.out.println("The cart is full. Can't add more items.");
16          }
17          else if(itemsOrdered.contains(media)){
18              System.out.println(media.getTitle() + " is already in the cart.");
19          }
20          else{
21              itemsOrdered.add(media);
22              System.out.println("The item \"" + media.getTitle() + "\" has been added.");
23          }
24      }
```

Figure 5.1: Cart Class 1

```
26      //ham them mot danh sach cac doi tuong media vao cart
27      public int addMediaTienLC(MediaTienLC... mediaArray){
28          int addedCount = 0;
29          for(MediaTienLC media : mediaArray){
30              if(itemsOrdered.size() == MAX_NUMBERS_ORDERED){
31                  System.out.println("The cart is full. Can't add more items.");
32              }
33              else if(itemsOrdered.contains(media)){
34              System.out.println(media.getTitle() + " is already in the cart.");
35              }
36              else{
37                  itemsOrdered.add(media);
38                  System.out.println("The item \"" + media.getTitle() + "\" has been added.");
39                  addedCount++;
40              }
41          }
42          return addedCount;
43      }
44      //Xoa media
45      public void removeMediaTienLC(MediaTienLC media){
46          if(itemsOrdered.isEmpty()){
47              System.out.println("Your cart is empty.");
48          }
49          if(itemsOrdered.contains(media)){
50              itemsOrdered.remove(media);
```

Figure 5.2: Cart Class 2

```
51              System.out.println("Removed \"" + media.getTitle() + "\" successfully.");
52          }
53          else{
54              System.out.println("Item you want to remove not found.");
55          }
56      }
57      //update total cost
58      public float totalCostTienLC(){
59          float sum = 0.00f;
60          for(MediaTienLC media : itemsOrdered){
61              sum += media.getCost();
62          }
63          return sum;
64      }
65       public void print() {
66          StringBuilder output = new StringBuilder("**********CART********** \nOrdered items: \n");
67          for (int i = 0; i < itemsOrdered.size(); i++) {
68              MediaTienLC media = itemsOrdered.get(i);
                output.append(i + 1 + ".[" + media.getTitle() + "] - [" + media.getCategory() + "] - ["
70                      + media.getDirector() + "] - [" + media.getLength() + "]: "
71                      + media.getCost() + " $\n");
72          }
73          output.append("total: ").append(totalCostTienLC()).append(" $\n");
74          output.append("**************************************************\n");
```

Figure 5.3: Cart Class 3

```
75              System.out.println(output);
76      }
77  // Tìm kiếm theo ID (index)
78      public void searchById(int i) {
79          if (i > itemsOrdered.size() || i <= 0) {
80              System.out.println("No match found!");
81          } else {
82              MediaTienLC media = itemsOrdered.get(i - 1);
83              System.out.println("Result: " + "[" + media.getTitle() + "] - [" + media.getCategory() + "] - ["
84                      + media.getDirector() + "] - [" + media.getLength() + "]: " + media.getCost() + " $\n");
85          }
86      }
87
88      // Tìm kiếm theo tiêu đề
89      public void searchByTitle(String title) {
90          for (MediaTienLC media : itemsOrdered) {
91              if (media.getTitle().equals(title)) {
92                  System.out.println("Result: " + "[" + media.getTitle() + "] - [" + media.getCategory() + "] - ["
93                          + media.getDirector() + "] - [" + media.getLength() + "]: " + media.getCost() + " $\n");
94                  return;
95              }
96          }
97          System.out.println("No match found!");
98      }
```

Figure 5.4: Cart Class 4

```
100         // Lọc các món trong giỏ hàng theo tiêu đề
101         public void filterByTitle(String title) {
102             for (MediaTienLC media : itemsOrdered) {
103                 if (media.getTitle().contains(title)) {
104                     System.out.println("Result: " + "[" + media.getTitle() + "] - [" + media.getCategory() + "] - ["
105                             + media.getDirector() + "] - [" + media.getLength() + "]: " + media.getCost() + " $\n");
106                 }
107             }
108         }
109 
110         // Sắp xếp giỏ hàng theo tiêu đề
111         public void sortByTitle() {
112             Collections.sort(itemsOrdered, new Comparator<MediaTienLC>() {
113                 public int compare(MediaTienLC m1, MediaTienLC m2) {
114                     int titleComparison = m1.getTitle().compareTo(m2.getTitle());
115                     if (titleComparison != 0) {
116                         return titleComparison;
117                     }
118                     return Float.compare(m2.getCost(), m1.getCost()); // Nếu tiêu đề giống nhau, sắp xếp theo giá
119                 }
120             });
121             System.out.println("Cart sorted by title and cost!");
122         }
```

Figure 5.5: Cart Class 5

```
124         // Sắp xếp giỏ hàng theo giá
125         public void sortByCost() {
126             Collections.sort(itemsOrdered, new Comparator<MediaTienLC>() {
127                 public int compare(MediaTienLC m1, MediaTienLC m2) {
128                     int costComparison = Float.compare(m2.getCost(), m1.getCost());
129                     if (costComparison != 0) {
130                         return costComparison;
131                     }
132                     return m1.getTitle().compareTo(m2.getTitle()); // Nếu giá giống nhau, sắp xếp theo tiêu đề
133                 }
134             });
135             System.out.println("Cart sorted by cost!");
136         }
137 
138         // Làm sạch giỏ hàng khi đặt hàng
139         public void clear() {
140             itemsOrdered.clear();
141             System.out.println("Your cart has been cleared.");
142         }
143     }
```

Figure 5.6: Cart Class 6

6 Update the Store class to work with Media

```java
package hust.soict.dsai.aims.store;
import hust.soict.dsai.aims.media.MediaTienLC;
import java.util.ArrayList;
public class StoreTienLc {
    private final ArrayList<MediaTienLC> itemsInStore = new ArrayList<MediaTienLC>();
    private boolean checkMediaTienLC(MediaTienLC media){
        for(MediaTienLC m : itemsInStore){
            if (m.equals(media)){
                return true;
            }
        }
        return false;
    }
    public void addMediaTienLC(MediaTienLC media){
        if(!checkMediaTienLC(media)){
            itemsInStore.add(media);
            System.out.println(media.getTitle() + " 've been added to the store !");
        }
        else{
            System.out.println(media.getTitle() + " 'already exists in the store.");
        }
    }

```

Figure 6.1: Store Class 1

```java
    public void removeMediaTienLC(MediaTienLC media){
        if(checkMediaTienLC(media)){
            itemsInStore.remove(media);
            System.out.println(media.getTitle() + " 've been deleted from the store.");
        }
        else{
            System.out.println("There is no "+ media.getTitle() + " in the store.");
        }
    }

    public void printStore(){
        System.out.println("\n--- Store ---");
        if(itemsInStore.isEmpty()){
            System.out.println("No Dvds in store.");
        }
        else{
            itemsInStore.forEach(dvd -> System.out.println(dvd));
        }
        System.out.println("-------------");
    }
```

Figure 6.2: Store Class 2

```java
    @Override //Dinh nghia lai phuong thuc trong lop Object cua thu vien java.lang
    public String toString() {
        StringBuilder string = new StringBuilder("***************STORE***************\nitems in the store: \n");
        if(itemsInStore.isEmpty()) string.append("There is no media items in the store.\n");
        else {
            for (MediaTienLC media : itemsInStore) {
                string.append(media.getTitle()).append(" - ").append(media.getCost()).append(" $\n");
            }
        }
        string.append("***************************************");
        return string.toString();
    }
}

```

Figure 6.3: Store Class 3

7 Constructors of whole classes and parent classes

```
7       //Constructor
8       public TrackTienLC(String title, int length){
9           super();
10          this.title = title;
11          this.length = length;
12      }
```

Figure 7.1: Constructor Track Class

```
9   //Constructor cua CompactDisc
10  public CompactDiscTienLC(String title, String category, float cost, String artist){
11      super(title, category, cost, "", 0);
12      this.artist = artist;
13      this.tracks = new ArrayList<>(); //Khoi tao danh sach tracks
14  }
```

Figure 7.2: Constructor CompactDisc Class

Lớp Disc kế thừa lớp Media, khi đó lớp Media là lớp cha, lớp Disc là lớp con.

```
9       //Constructor
10      public MediaTienLC(String title, String category, float cost){
11          this.id = ++idCounter;
12          this.title = title;
13          this.category = category;
14          this.cost = cost;
15      }
```

Figure 7.3: Constructor Media Class

```
6   //Constructor
7   public DiscTienLC(String title, String category, float cost, String director, int length){
8       super(title, category, cost);
9       this.director = director;
10      this.length = length;
11  }
```

Figure 7.4: Constructor Disc Class

8 Unique item in a list

Để tránh trùng lặp các phần tử media trong giỏ hàng hoặc các track trong một đĩa CD, chúng ta có thể ghi đè lại phương thức equals() mặc định kế thừa từ lớp Object. Việc này cho phép so sánh bản chất thay vì so sánh vị trí ô nhớ của các đối tượng, qua đó ngăn chặn thêm các phần tử bị trùng lắp vào danh sách.

```
40      @Override
        public boolean equals(Object obj){
42          if(this == obj) return true;
43          if (obj == null || getClass() != obj.getClass()) return false;
44          MediaTienLC media = (MediaTienLC) obj;
45          return this.title != null && this.title.equals(media.title);
46      }
```

Figure 8.1: Override equals in Media Class

```java
    @Override
    public boolean equals(Object obj) {
        // Kiểm tra đối tượng truyền vào có phải là đối tượng Track hay không
        if (this == obj) return true;  // So sánh tham chiếu
        if (obj == null || getClass() != obj.getClass()) return false;  // Kiểm tra null và kiểu đối tượng
        TrackTienLC track = (TrackTienLC) obj;  // Ép kiểu đối tượng
        return this.title != null && this.title.equals(track.title) && this.length == track.length;  // So sánh title và length
        }
}
```

Figure 8.2: Override equals in Track Class

9 Polymorphism with toString() method

```java
package hust.soict.dsai.aims.media;
import java.util.ArrayList;
public class TestMediaTienLC {
    public static void main(String[] args) {
        // Tạo một ArrayList chứa các đối tượng Media
        ArrayList<MediaTienLC> mediaList = new ArrayList<>();

        // Thêm các đối tượng CD, DVD và Book vào danh sách
        mediaList.add(new DigitalVideoDiscTienLC("Star Wars", "Science Fiction", "George Lucas", 120, 24.95f));
        mediaList.add(new CompactDiscTienLC("Thriller", "Pop", 30.99f, "Michael Jackson"));
        mediaList.add(new BookTienLC("Harry Potter", "Fantasy", 19.95f));

        // Duyệt qua danh sách và gọi phương thức toString() để in thông tin
        for (MediaTienLC media : mediaList) {
            System.out.println(media.toString());  // Gọi phương thức toString() của từng đối tượng
        }
    }
}
```

Figure 9.1: Code mô phỏng Polymorphism

```java
    @Override
    public String toString(){
        return "ID: " + id + "\nTitle: " + title + "\nCategory: " + category + "\nCost: " + cost;
    }
    public abstract int getLength();
    public abstract String getDirector();
}
```

Figure 9.2: Override toString() in Media Class

Kết quả

```
run:
ID: 1
Title: Star Wars
Category: Science Fiction
Cost: 24.95
ID: 2
Title: Thriller
Category: Pop
Cost: 30.99
ID: 3
Title: Harry Potter
Category: Fantasy
Cost: 19.95
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 9.3: Result demo Polymorphism

Lớp Media là lớp cơ sở được kế thừa bởi các lớp cụ thể hơn là CompactDisc, DigitalVideoDisc và Book. Khi

khởi tạo các đối tượng cd, dvd, book thuộc lớp con rồi gán chúng cho biến kiểu Media, ta áp dụng kỹ thuật gọi là upcasting.

Việc thêm chúng vào danh sách media và duyệt danh sách để in ra thông tin mỗi phần tử bằng phương thức toString() là ví dụ điển hình cho tính đa hình động. Mỗi lớp con có thể cài đặt riêng toString() nên kết quả sẽ khác nhau dựa theo loại đối tượng, mà không cần quan tâm đến kiểu cụ thể của từng phần tử.

10 Sort media in the cart

Sắp xếp các media trong giỏ hàng theo hai tiêu chí:
Bằng title: Hiển thị tất cả các media theo thứ tự bảng chữ cái. Trong trường hợp cùng title, media có cost cao hơn sẽ được hiển thị trước.
Bằng cost: Hiển thị theo thứ tự cost giảm dần. Trong trường hợp cost như nhau, sắp xếp media theo thứ tự bảng chữ cái

```
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 10.1: Add the comparators as attributes of the Media class

```
2  package hust.soict.dsai.aims.media;
3  import java.util.Comparator;
4
5  public class CompareByCostTitle implements Comparator<MediaTienLC> {
6      @Override
7      public int compare(MediaTienLC m1, MediaTienLC m2) {
8          // So sánh theo chi phí (chi phí cao hơn ưu tiên trước)
9          int costComparison = Float.compare(m2.getCost(), m1.getCost());
10
11         // Nếu chi phí giống nhau, so sánh theo tiêu đề
12         if (costComparison == 0) {
13             return m1.getTitle().compareTo(m2.getTitle());  // So sánh theo tiêu đề (theo thứ tự chữ cái)
14         }
15
16         return costComparison;  // Nếu chi phí khác nhau, sắp xếp theo chi phí
17     }
18 }
```

Figure 10.2: MediaComparatorByCostTitle Class

```
2  package hust.soict.dsai.aims.media;
3  import java.util.Comparator;
4
5  public class CompareByTitleCost implements Comparator<MediaTienLC> {
6      @Override
7      public int compare(MediaTienLC m1, MediaTienLC m2) {
8          // So sánh theo tiêu đề
9          int titleComparison = m1.getTitle().compareTo(m2.getTitle());
10
11         // Nếu tiêu đề giống nhau, so sánh theo chi phí (chi phí cao hơn ưu tiên trước)
12         if (titleComparison == 0) {
13             return Float.compare(m2.getCost(), m1.getCost());  // So sánh theo chi phí giảm dần
14         }
15
16         return titleComparison;  // Nếu tiêu đề khác nhau, sắp xếp theo tiêu đề
17     }
18 }
19
```

Figure 10.3: MediaComparatorByTitleCost Class

11 Create a complete console application in the Aims Class

```
run:
Thriller 've been added to the store !
Star Wars 've been added to the store !
Harry Potter 've been added to the store !
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3
```

Figure 11.1: Màn hình chính

```java
package hust.soict.dsai.aims;
import java.util.Scanner;

import hust.soict.dsai.aims.cart.CartTienLC;
import hust.soict.dsai.aims.media.BookTienLC;
import hust.soict.dsai.aims.media.CompactDiscTienLC;
import hust.soict.dsai.aims.media.DigitalVideoDiscTienLC;
import hust.soict.dsai.aims.media.MediaTienLC;
import hust.soict.dsai.aims.media.PlayableTienLC;
import hust.soict.dsai.aims.store.StoreTienLc;

public class Aims {

    private static StoreTienLc store = new StoreTienLc(); // Cửa hàng
    private static CartTienLC cart = new CartTienLC(); // Giỏ hàng
    private static Scanner scanner = new Scanner(System.in); // Scanner để nhập dữ liệu

    public static void main(String[] args) {
        // Thêm một số media vào cửa hàng (ví dụ)
        store.addMediaTienLC(new CompactDiscTienLC("Thriller", "Pop", 15.99f, "Michael Jackson"));
        store.addMediaTienLC(new DigitalVideoDiscTienLC("Star Wars", "Science Fiction", "George Lucas", 120, 24.95f));
        store.addMediaTienLC(new BookTienLC("Harry Potter", "Fantasy", 19.95f));
        showMenu();
    }

```

Figure 11.2: Aims Class 1

```
    // Hiển thị menu chính
    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 current cart");
        System.out.println("0. Exit");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3");
    }


    // Hiển thị menu cửa hàng
    public static void storeMenu() {
        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");
    }

    // Menu chi tiết media
    public static void mediaDetailsMenu(MediaTienLC media) {
        while (true) {
            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 choice = scanner.nextInt();
            switch (choice) {
                case 1: // Add to cart
                    cart.addMediaTienLC(media);
                    System.out.println("Media added to cart.");
                    break;
                case 2: // Play
                    if (media instanceof PlayableTienLC) {
                        ((PlayableTienLC) media).play();
                    } else {
                        System.out.println("This media cannot be played.");
                    }
                    break;
                case 0: // Back
                    return;
```

Figure 11.3: Aims Class 2

```
                case 0: // Back
                    return;
                default:
                    System.out.println("Invalid choice! Please choose again.");
            }
        }
    }
    // Hiển thị menu giỏ hàng
    public static void cartMenu() {
        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. Place order");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3-4-5");
    }
    // Sắp xếp media trong giỏ hàng
    public static void sortMediasInCart() {
        System.out.println("Choose sorting criteria: ");
        System.out.println("1. By title");
        System.out.println("2. By cost");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                cart.sortByTitle();
                break;
            case 2:
                cart.sortByCost();
                break;
            default:
                System.out.println("Invalid choice! Please choose again.");
        }
    }
    // Đặt hàng
    public static void placeOrder() {
        System.out.println("Order placed!");
        cart.clear();
    }
    // Cập nhật cửa hàng (thêm/xóa media)
    public static void updateStore() {
        // Chức năng thêm/xóa media trong cửa hàng có thể thêm vào sau nếu cần thiết
    }
}
```

Figure 11.4: Aims Class 3

12 Class Diagram

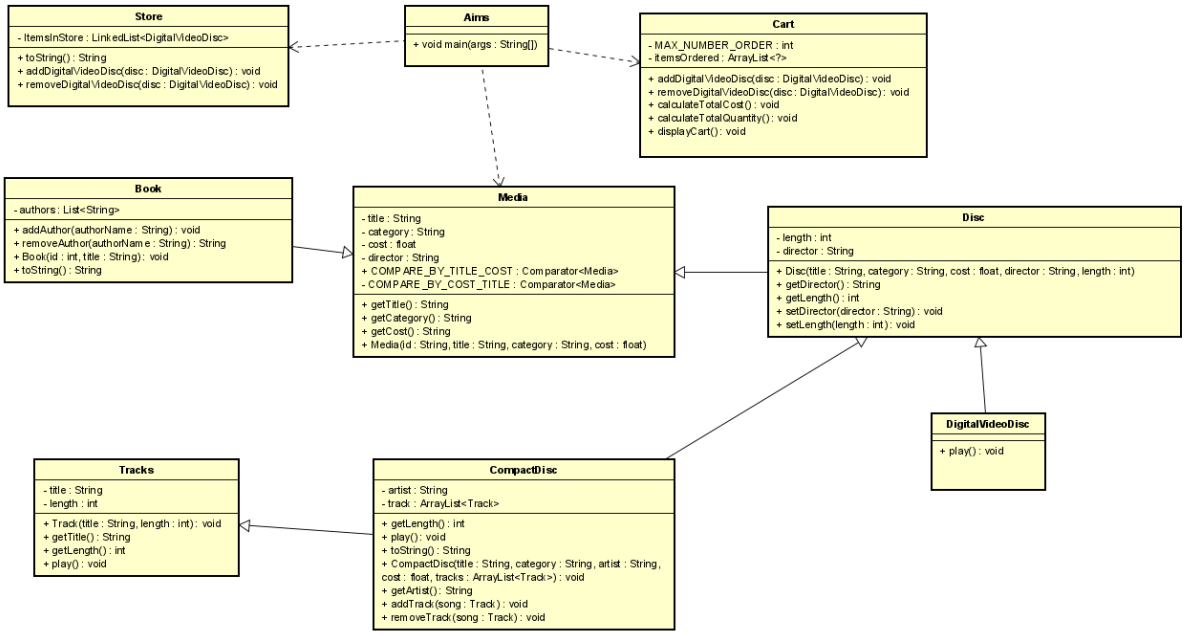

Figure 12.1: Class Diagram

13 UseCase Diagram

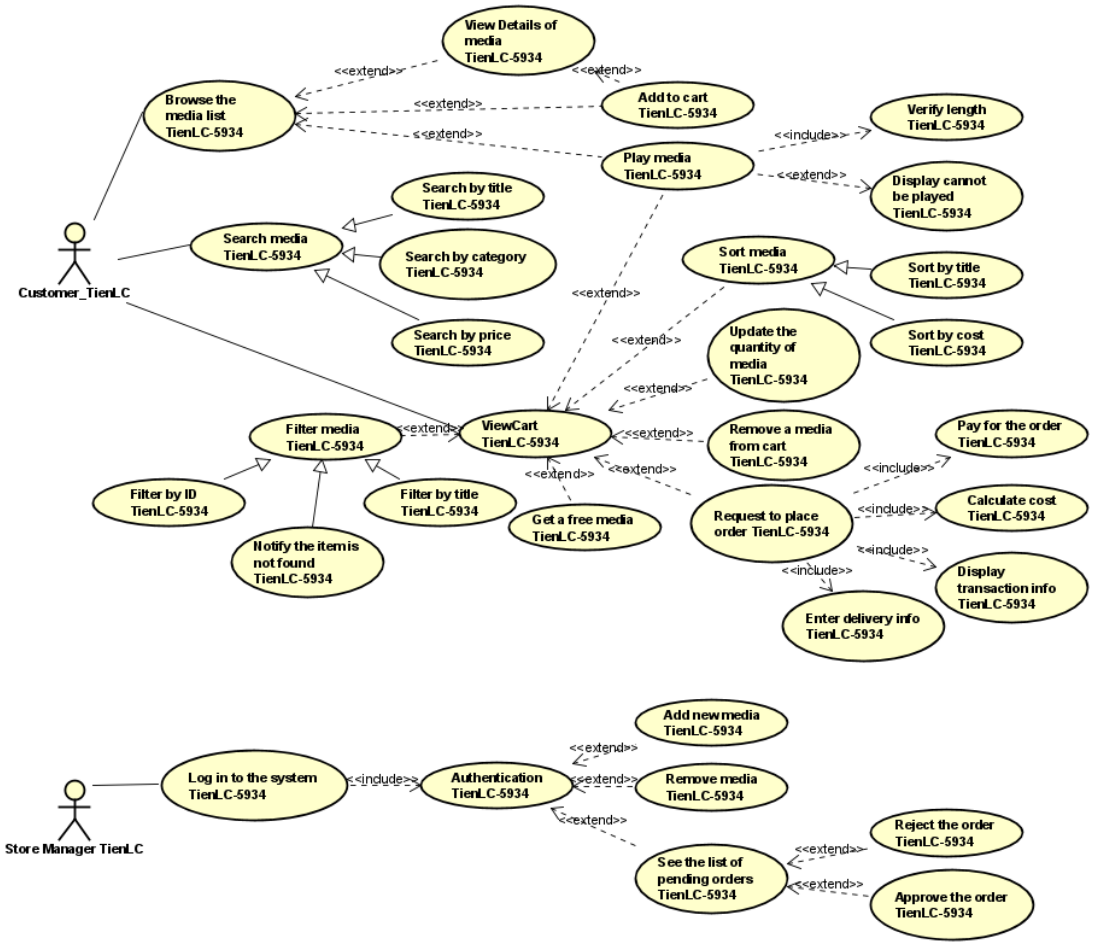

Figure 13.1: Use Case Diagram

14 Answer Questions

- Thay vì sử dụng Comparator để so sánh các mục trong giỏ hàng, chúng ta có thể sử dụng giao diện Comparable và ghi đè phương thức compareTo().

Giả sử chúng ta đang áp dụng Comparable interface approach. Lớp nào nên triển khai giao diện Comparable?

-> Lớp nên triển khai giao diện Comparable là lớp chứa đối tượng mà bạn muốn so sánh, trong trường hợp này, là abstract class “Media”.

- Liệu có thể có hai quy tắc sắp xếp (theo title sau đó là cost và theo cost sau đó là title) nếu sử dụng cách tiếp cận này với giao diện Comparable?

-> Không, với giao diện Comparable, bạn chỉ có thể có một quy tắc sắp xếp cho mỗi lớp. Điều này là do phương thức compareTo() chỉ trả về một giá trị int.