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 -LAB01

Họ và tên sv: Vương Quốc Huy

MSSV: 20225637

Lớp: Việt Nhật 07-K67

GVHD: Lê Thị Hoa

HTGD: Đặng Mạnh Cường

Hà Nội 9/2024

BÁO CÁO THỰC HÀNH LAB 4
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Table of Contents

1. Tạo lớp Book

```java
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 int getId() {
        return id;
    }

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

```java
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;
}
```

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


public boolean addAuthor(String authorName) {
    if (authors.contains(authorName)) {
        System.out.println("Author already existed");
        return false;
    } else {
        authors.add(authorName);
        return true;
    }
}


public boolean removeAuthor(String authorName) {
    if (!authors.contains(authorName)) {
        System.out.println("Author doesn't exist");
        return false;
    } else {
        authors.remove(authorName);
        return true;
    }
```

2. Tạo lớp abstract Media

```java
public abstract class Media {
    private int id;
    private String title;
    private String category;
    private float cost;

    public Media(){}

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

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

```
26
27      public boolean equals(Object o){
28          if(!(o instanceof Media)) return false;
29          return ((Media) o).getTitle().equals(this.title);
30      }
31      public int getId(){
32          return id;
33      }
34  }
```

- Kế thừa lớp Media cho Book và DigitalVideoDisc

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  public class Book extends Media {
5      private int id;
6      private String title;
7      private String category;
8      private float cost;
9      private List<String> authors = new ArrayList<String>();
10
```

```
49      public boolean addAuthor(String authorName) {
50          if (authors.contains(authorName)) {
51              System.out.println("Author already existed");
52              return false;
53          } else {
54              authors.add(authorName);
55              return true;
56          }
57      }
58
59
60      public boolean removeAuthor(String authorName) {
61          if (!authors.contains(authorName)) {
62              System.out.println("Author doesn't exist");
63              return false;
64          } else {
65              authors.remove(authorName);
66              return true;
67          }
68      }
69  }
```

```java
public class DigitalVideoDisc extends Media{
  //  private String title;
   // private String category;
    private String director;
    private int length;
  //  private float cost;
    private static int nbDigitalVideoDisc = 0;
    //private int id;
```

3. Tạo lớp CompactDisc

3.1 Tạo lớp Disc

- Tạo lớp Disc kế thừa lớp Media, có thêm 2 thuộc tính length và director và tạo constructor.

```java
public class Disc extends Media{
    private int length;
    private String director;

    public Disc(int id, String title, String category, Float cost, int length, String director){
        super(id, title, category, cost);
        this.length = length;
        this.director = director;
    }

    public Disc(String title, float cost){
        super(title, cost);
    }

    public int getLength(){
        return length;
    }
    public String getDirector(){
        return director;
    }
    public void setDirector(String director){
        this.director = director;
    }
    public void setLength(int length){
        this.length = length;
    }
```

- Sửa đổi để lớp DigitalVideoDisc kế thừa lớp Disc

```java
public class DigitalVideoDisc extends Disc{
    private String title;
   private String category;
    private String director;
    private int length;
    private float cost;
    private static int nbDigitalVideoDisc = 0;
    private int id;
  public DigitalVideoDisc(String title){
    super();
   setTitle(title);
    nbDigitalVideoDisc++;
    setId(nbDigitalVideoDisc);
  }
  public DigitalVideoDisc(String title, float cost){
    super(title, cost);
  }
  public DigitalVideoDisc(String category, String title, float cost){
    this(title);
    setCategory(category);
    setCost(cost);
  }

  public DigitalVideoDisc(String director, String category, String title, float cost){
    this(category, title, cost);
    setDirector(director);
```

```java
  public DigitalVideoDisc(String title, float cost){
    super(title, cost);
  }
  public DigitalVideoDisc(String category, String title, float cost){
    this(title);
    setCategory(category);
    setCost(cost);
  }

  public DigitalVideoDisc(String director, String category, String title, float cost){
    this(category, title, cost);
    setDirector(director);

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

- Tạo lớp CompactDisc kế thừa lớp Disc

```java
public class CompactDisc extends Disc{
    private String artist;
    public CompactDisc(int id, String title, String category, float cost, int length, String director){
        super(id, title, category, cost,length, director);
    }
    public CompactDisc(String title, float cost){
        super(title, cost);
    }
}
```

3.2 Tạo lớp Track

- Tạo lớp Track có 2 thuộc tính title, length và tạo constructor.

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

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

3.3 Chỉnh sửa lớp CompactDisc

- Thêm 2 thuộc tính private artist và tracks, tạo getter cho artist, tạo constructor

```java
import java.util.List;
import java.util.ArrayList;
public class CompactDisc extends Disc{
    private String artist;
    private List<Track> tracks = new ArrayList<>();
    public CompactDisc(int id, String title, String category, float cost, int length, String director){
        super(id, title, category, cost,length, director);
    }
    public CompactDisc(String title, float cost){
        super(title, cost);
    }
    public CompactDisc(int id, String title, String category, float cost, int length, String director, String
        super(id, title, category, cost, length, director);
        this.artist = artist;
        this.tracks.addAll(tracks);
    }
    public boolean addTrack(Track track){
        if (tracks.contains(track)){
            System.out.println("Track already existed");
            return false;
        } else {
            tracks.add(track);
            System.out.println("Track added successfully");
            return true;
        }
    }
```

```java
public boolean removeTrack(Track track){
    if(!tracks.contains(track)){
        System.out.println("No such track found");
        return false;
    } else {
        tracks.remove(track);
        System.out.println("Track removed successfully");
        return true;
    }
}
public int getLength(){
    int length = 0;
    for(Track track : tracks){
        length += track.getLength();
    } return length;
}
    }
```

- Tạo các phương thức addTrack và removeTrack.

```
public boolean addTrack(Track track){
    if (tracks.contains(track)){
        System.out.println("Track already existed");
        return false;
    } else {
        tracks.add(track);
        System.out.println("Track added successfully");
        return true;
    }
}
public boolean removeTrack(Track track){
    if(!tracks.contains(track)){
        System.out.println("No such track found");
        return false;
    } else {
        tracks.remove(track);
        System.out.println("Track removed successfully");
        return true;
    }
}
```

- Tạo phương thức getLength trả về tổng length của tất cả các track có trong CD.

```
public int getLength(){
    int length = 0;
    for(Track track : tracks){
        length += track.getLength();
    } return length;
}
    }
```

4. Tạo interface Playable

- Tạo interface Playable và thêm prototype public void play();

```
1   public interface Playable{
2       public void play();
3   }
4
```

- Implement Playable với các lớp CompactDisc, DigitalVideoDisc và Track

```
public class DigitalVideoDisc extends Disc implements Playable{
    private String title;
   private String category;
```

```java
import java.util.List;
import java.util.ArrayList;
public class CompactDisc extends Disc implements Playable{
    private String artist;
    private List<Track> tracks = new ArrayList<>();
```

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

- Implement phương thức play() cho lớp DigitalVideoDisc và Track

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

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

- Implement phương thức play() cho lớp CompactDisc, play tất cả track có trong CD đó.

```java
public void play(){
    for(Track track : tracks){
        track.play();
    }
}
```

5. Chỉnh sửa lớp Cart để hoạt động với Media

- Thay thế 2 phương thức addDigitalVideoDisc() và removeDigitalVideoDisc() bằng addMedia() và removeMedia(), thay thế mảng itemsOrdered bằng một đối tượng ArrayList

```java
import java.util.List;
import java.util.ArrayList;
public class Cart{
    public static final int MAX_NUMBERS_ODERED = 20;
    private List<Media> itemsOrdered = new ArrayList<>();
    private int qtyOrdered = 0;
```

```java
public boolean addMedia(Media media){
 if(itemsOrdered.size() == MAX_NUMBERS_ODERED){
     System.out.println("The cart is already full");
     return false;
 }
 else{
     itemsOrdered.add(media);
     System.out.println("Item added");
     return true;
 }
}
public boolean removeMedia( Media media){
 if(!itemsOrdered.contains(media)){
     System.out.println("No such item found");
     return false;
 } else{
     itemsOrdered.remove(media);
     System.out.println("Item removed");
     return true;
 }
```

- Chỉnh sửa phương thức totalCost().

```java
public float totalCost(){
 float sum = 0f;
 for(Media item : itemsOrdered){
     sum += item.getCost();
 } return sum;
}
```

6. Chỉnh sửa lớp Store để hoạt động với Media

- Thay thế mảng itemsInStore bằng đối tượng ArrayList. Thay đổi các phương thức làm việc với DigitalVideoDisc ban đầu thành làm việc với Media.

```java
import java.util.List;
import java.util.ArrayList;
public class Store {
    public static final int MAX_NUMBERS_DVD = 100;
    private List<Media> itemsInStore = new ArrayList<>();
 //   private int qtyOrdered = 0;

    public boolean addMedia(Media media){
    if(itemsInStore.size() == MAX_NUMBERS_DVD){
        System.out.println("Store is already full");
        return false;
    }
    else{
        itemsInStore.add(media);
        System.out.println("Item added successfully");
        return true;
    }
   }
   public boolean removeMedia( Media media){
    if(!itemsInStore.contains(media)){
        System.out.println("Item not found");
        return false;
    } else{
        itemsInStore.remove(media);
        System.out.println("Item removed successfully");
        return true;
```

```java
  }
  public boolean removeMedia(String title){
   for(Media item : itemsInStore){
       if(item.getTitle().equals(title)){
           itemsInStore.remove(item);
           System.out.println("Item removed");
           return true;
       }
   }
   System.out.println("item not found");
   return false;
  }
```

7. Các hàm tạo của toàn lớp và lớp cha

8. Item duy nhất trong một list

- Ghi đè phương thức equals() của lớp Media.

```java
    @Override
    public boolean equals(Object o){
        if(!(o instanceof Media)) return false;
        return ((Media) o).getTitle().equals(this.title);
    }
```

- Ghi đè phương thức equals() của lớp Track.

```java
    @Override
public boolean equals(Object o) {
    if (!(o instanceof Track)) return false;
    Track other = (Track) o;
    return other.getTitle().equals(title) && other.getLength() == length;
}
```

- Câu hỏi: Khi ghi đè phương thức equals() của lớp Object, chúng ta phải cast tham số o thành kiểu đang cần xử lí. Ví dụ, đối với lớp Media, chúng ta cần cast o thành kiểu Media. Nếu truyền vào một đối tượng không phải instance của Media, điều gì sẽ xảy ra? → Nếu không xử lí các kiểu khác với Media, quá trình biên dịch sẽ xảy ra lỗi incompatible type.

9. Test đa hình với phương thức toString()

```java
public class ToStringTest {
    Run main | Debug main
    public static void main(String[] args) {
        List<Media> ms = new ArrayList<>();
        ms.add(new DigitalVideoDisc("Doraemon", "Anime", "Fujio", 15f));
        ms.add(new Book(0, "Why we sleep", "Scientific", 9f));
        ms.add(new CompactDisc(1, "Gao ranger", "Super sentai", 20f, 2, "Peter"));

        for(Media m : ms) {
            System.out.println(m.toString());
        }

    }
}
```

10. Sắp xếp media trong giỏ hàng

10.1 Tạo các class so sánh

MediaComparatorByCostTitle.java > ...

```java
import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media>{
    @Override
    public int compare(Media a, Media b) {
        return Comparator.comparingDouble(Media::getCost)
                .reversed()
                .thenComparing(Media::getTitle)
                .compare(a, b);
    }
}

```

MediaComparatorByTitleCost.java > MediaComparatorByTitleCost

```java
import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {
    @Override
    public int compare(Media a, Media b) {
        return Comparator.comparing(Media::getTitle)
                .thenComparingDouble((o) -> -o.getCost())
                .compare(a, b);
    }
}
```

10.2 Test với Collections.sort()

ComparatorTest.java > ...

```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class ComparatorTest {
    Run main | Debug main
    public static void main(String[] args) {
        List<Media> m = new ArrayList<>();
        m.add(new Book(0, "B-Book", "History book", 10f));
        m.add(new Book(1, "C-Book", "Music book", 18f));
        m.add(new Book(1, "C-Book", "Cooking book", 20f));
        m.add(new Book(2, "A-Book", "Cartoon", 3f));

        System.out.println("Sort by cost then title: ");
        Collections.sort(m, Media.COMPARE_BY_COST_TITLE);
        m.forEach(System.out::println);

        System.out.println("\nSort by title then cost: ");
        Collections.sort(m, Media.COMPARE_BY_TITLE_COST);
        m.forEach(System.out::println);
    }
}
```

11. Tạo console app hoàn chỉnh

```
public class AimsProject{
    public static void main(String[] args){
    //    System.out.println(anOrder.totalCost());
    //    anOrder.listID();
    public static void showMenu() {
        while (true) {
            System.out.println("Vuong Quoc Huy 20225637");
            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("--------------------------------------------------------");

            int choice = scanner.nextInt();
            switch (choice) {
                case 0:
                    break;
                case 1:
                    storeMenu();
                    continue;
                case 2:
                    updateStoreMenu();
                    continue;
                case 3:
                    cartMenu();
                    continue;
                default:
                    System.out.println("Invalid choice, please try again");
                    break;
            }
            break;
```

```
    public static void updateStoreMenu() {
        while (true) {
            System.out.println("Options:");
            System.out.println("--------------------------------------------------------");
            System.out.println("1. Add a media");
            System.out.println("2. Remove a media");
            System.out.println("0. Back");
            System.out.println("--------------------------------------------------------");
            System.out.println("---> ");

            int choice = scanner.nextInt();
            switch (choice) {
                case 0:
                    break;
                case 1:
                    Media newItem = createNewMedia();
                    store.addMedia(newItem);
                    continue;
                case 2:
                    System.out.println("Enter the media title: ");
                    scanner.nextLine();
                    String title = scanner.nextLine();
                    store.removeMedia(title);
                    continue;
                default:
                    System.out.println("Invalid choice, please try again");
                    break;
            }
```

```java
public static Media createNewMedia() {
    Media item = null;
    System.out.println("Enter the Media info");
    System.out.println("Title: ");
    scanner.nextLine();
    String title = scanner.nextLine();
    System.out.println("Cost: ");
    float price = scanner.nextFloat();

    while (true) {
        System.out.println("Chosse a media type: ");
        System.out.println("1. Book");
        System.out.println("2. DVD");
        System.out.println("3. CD");
        System.out.println("0. Back");

        int choice = scanner.nextInt();

        switch (choice) {
            case 0:
                break;
            case 1:
                item = new Book(title, price);
                break;
            case 2:
                item = new DigitalVideoDisc(title, price);
                break;
            case 3:
```

```
            case 3:
                item = new CompactDisc(title, price);
                break;
            default:
                System.out.println("Invalid choice, please try again");
                break;
        }
        break;
    }
    return item;
}

public static void storeMenu() {
    while (true) {
            store.printItemsInStore();
            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 media");
            System.out.println("4. See current cart");
            System.out.println("0. Exit");
            System.out.println("--------------------------------------------------------");
            System.out.println("Please choose a number: 0, 1, 2, 3, 4");
            System.out.println("--->");

            String title;
            Media item;
            int choice = scanner.nextInt();
            switch (choice) {
```

```
        switch (choice) {
            case 0:
                break;
            case 1:
                System.out.println("Enter the media title: ");
                scanner.nextLine();
                title = scanner.nextLine();
                item = store.searchByTitle(title);
                if(item != null) {
                    mediaDetailMenu(item);
                }
                continue;
            case 2:
                System.out.println("Enter the media title: ");
                scanner.nextLine();
                title = scanner.nextLine();
                item = store.searchByTitle(title);
                if(item != null) {
                    cart.addMedia(item);
                }
                continue;
            case 3:
                System.out.println("Enter the media title: ");
                scanner.nextLine();
                title = scanner.nextLine();
                item = store.searchByTitle(title);
                if(item instanceof Playable) {
                    ((Playable) item).play();
                } else if (item != null) {
                    System.out.println("This item is not playable");
```

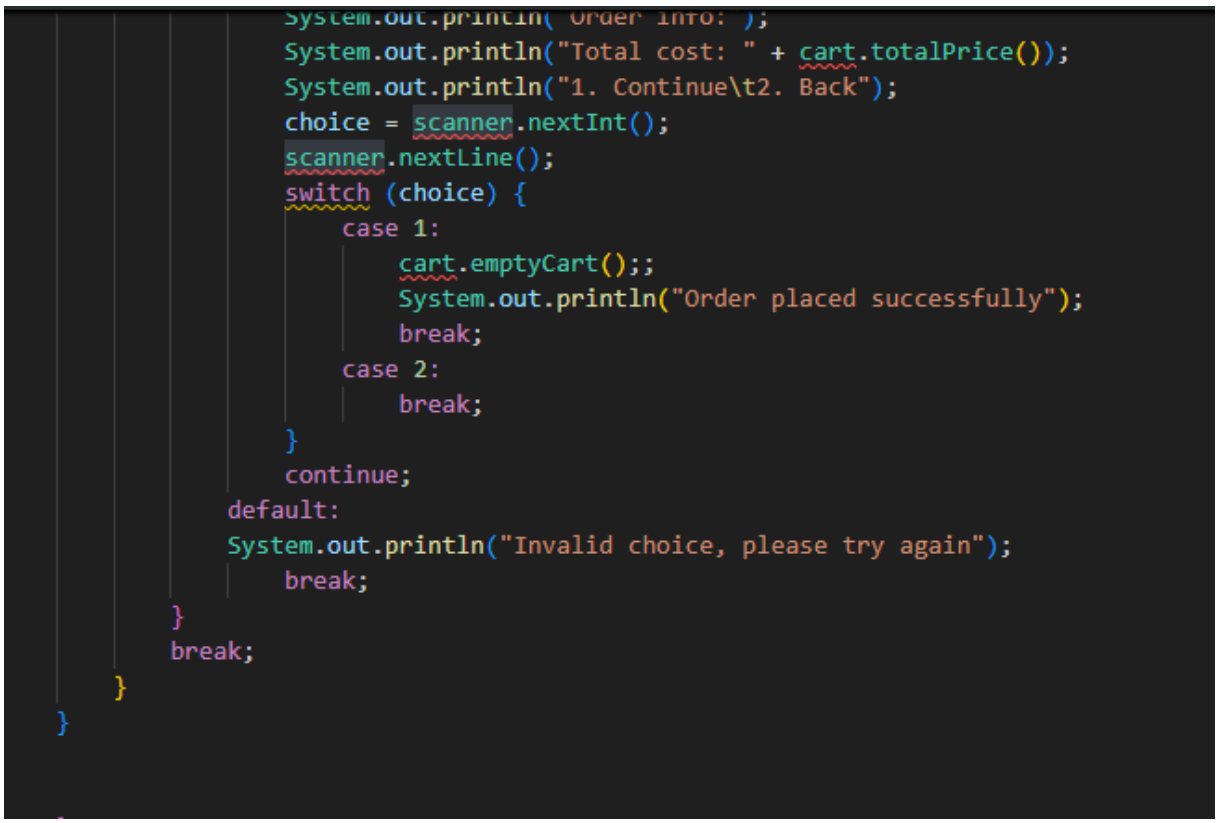
```
                        System.out.println("This item is not playable");
                    }
                    continue;
                case 4:
                    cart.printItemsInCart();
                    continue;
                default:
                    System.out.println("Invalid choise, please try again");
                    continue;
            }
            break;
        }
    }

public static void mediaDetailMenu(Media item) {
    while (true) {
        System.out.println(item.toString());
        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 chosse a number: 0, 1, 2");
        System.out.println("--->");

        int choice = scanner.nextInt();
        switch (choice) {
            case 0:
```

```
            case 0:
                break;
            case 1:
                cart.addMedia(item);
                continue;
            case 2:
                if(item instanceof Book) {
                    System.out.println("Books are not playable");
                } else if (item instanceof Playable) {
                    ((Playable) item).play();
                }
                continue;
            default:
                System.out.println("Invalid choice, please try again");
                break;
        }
        break;
    }
}

public static void cartMenu() {
    cart.printItemsInCart();
    while (true) {
        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 media");
        System.out.println("5. Place order");
```

```
        System.out.println("5. Place order");
        System.out.println("0. Back");
        System.out.println("Please choose a number: 0, 1, 2, 3, 4, 5");
        System.out.println("--->");

        int choice = scanner.nextInt();
        scanner.nextLine();
        String title = null;
        int id;
        Media item = null;
        switch (choice) {
            case 0:
                break;
            case 1:
                System.out.println("1. Filter by ID");
                System.out.println("2. Filter by title");
                System.out.print("Choose an option: ");
                int filterChoice = scanner.nextInt();
                scanner.nextLine(); // Xử lý ký tự xuống dòng
                switch (filterChoice) {
                    case 1:
                        System.out.print("Enter the ID: ");
                        id = scanner.nextInt();
                        scanner.nextLine(); // Xử lý ký tự xuống dòng
                        System.out.println(cart.searchById(id));
                        break;
                    case 2:
                        System.out.print("Enter the title: ");
                        title = scanner.nextLine();
                        System.out.println(cart.searchByTitle(title));
```

```
                System.out.println(cart.searchByTitle(title));
                break;
            default:
                System.out.println("Invalid filter option");
                break;
        }
        break;
    case 2:
        System.out.println("1. Sort by cost");
        System.out.println("2. Sourt by title");
        System.out.println("Choose an option: ");
        int sortChoice = scanner.nextInt();
        scanner.nextLine();
        switch (sortChoice) {
            case 1:
                cart.sortByCost();
                cart.printItemsInCart();
                break;
            case 2:
                cart.sortByTitle();
                cart.printItemsInCart();
                break;
            default:
                System.out.println("Invalid sort option");
                break;
        }
    case 3:
        System.out.println("Enter the title: ");
        title = scanner.nextLine();
        item  = cart.searchByTitle(title);
```

```
            item  = cart.searchByTitle(title);
            if(item != null) {
                cart.removeMedia(item);
            }
            cart.printItemsInCart();
            continue;
        case 4:
            System.out.print("Enter a title: ");
            title = scanner.nextLine();
            item = cart.searchByTitle(title);

            if (item == null) {
                System.out.println("Item not found.");
                continue;
            }

            if (item instanceof Playable) {
                ((Playable) item).play();
            } else {
                System.out.println("Item not playable.");
            }
            continue;
        case 5:
            System.out.println("Order info:");
            System.out.println("Total cost: " + cart.totalPrice());
            System.out.println("1. Continue\t2. Back");
            choice = scanner.nextInt();
            scanner.nextLine();
            switch (choice) {
                case 1:
```


System.out.println("Order Info: ");
System.out.println("Total cost: " + cart.totalPrice());
System.out.println("1. Continue\t2. Back");
choice = scanner.nextInt();
scanner.nextLine();
switch (choice) {
case 1:
cart.emptyCart();;
System.out.println("Order placed successfully");
break;
case 2:
break;
}
continue;
default:
System.out.println("Invalid choice, please try again");
break;
}
break;
}
}

12. Class Diagram