Báo cáo Lab 4 – oop

Họ và tên : Nguyễn Thanh Bình

Mssv : 20225792

1.Import the existing project into the workspace of Eclipse

2. Additional requirements of AIMS

3. Tạo lớp Book

```java
package hust.soict.hedspi.aims.media;
import java.util.*;

public class Book {

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

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

    public void setId(int id) {
        this.id = id;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public void setCategory(String category) {
        this.category = category;
    }
    public void setCost(float cost) {
        this.cost = cost;
    }


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

```java

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

    public void removeAuthor(String authorName) {
        if (authors.contains(authorName)) {
            authors.remove(authorName);
        } else {
            System.out.println("No author has been found to remove!");
        }
    }
}
```

4. Tạo lớp trừu tượng Media

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

public abstract class Media {
    private static int nbMedia = 0;
    private int id;

    private String title;
    private String category;
    private float cost;

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

    public int getId() {
        return id;
    }

    public String getTitle() {
        return title;
    }

    public String getCategory() {
        return category;
    }

    public float getCost() {
        return cost;
    }

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

}
```

4.1 Lớp book kế thừa Media

```java
package hust.soict.hedspi.aims.media;
import java.util.*;

public class Book extends Media {

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

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

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

    public void removeAuthor(String authorName) {
        if (authors.contains(authorName)) {
            authors.remove(authorName);
        } else {
            System.out.println("No author has been found to remove!");
        }
    }
}
```

5. Tạo lớp CompactDisc

5.1 Tạo lớp Disc kế thừa Media

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

public class Disc extends Media {

    private String director;
    private int length;


    public String getDirector() {
            return director;
        }
        public int getLength() {
            return length;
        }
    public Disc(String title) {
        super(title);
    }
    public Disc(String title, String category, float cost) {
        super(title, category, cost);
    }
    public Disc(String title, String category, String director, float cost) {
        super(title, category, cost);
        this.director = director;
    }
    public Disc(String title, String category, String director, int length, float cost) {
        super(title, category, cost);
        this.director = director;
        this.length = length;
    }


}
```

5.2 Tạo lớp Track

```java
package hust.soict.hedspi.aims.media;
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;
    }

}
```

5.3 Sửa lớp DigitalVideoDisc

```java
*/
package hust.soict.hedspi.aims.media;

public class DigitalVideoDisc extends Disc{


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


    @Override
    public String toString() {
        return "DVD: " + this.getTitle() +
                " - Category: " + this.getCategory() +
                " - Director: " + this.getDirector() +
                " - DVD length: " + this.getLength() +
                " - Cost: " + this.getCost() + "$";
    }

    public boolean isMatch(String title) {
        return this.getTitle().toLowerCase().contains(title.toLowerCase());
    }

}
```

5.4 CompactDisc kế thừa Media

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

public class CompactDisc extends Media {

    private String artist;
    private ArrayList<Track> tracks;


    public String getArtist() {
        return artist;
    }


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

    public void addTrack(Track track) {
        if (!tracks.contains(track)) {
            tracks.add(track);
        } else {
            System.out.println("Track already exists in CD.");
        }
    }

    public void removeTrack(Track track) {
        if (tracks.contains(track)) {
            tracks.remove(track);
        } else {
            System.out.println("Track does not exist in CD.");
        }
    }

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

}
```

6. Tạo playable interface

6.1 Tạo playable interface

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

public interface Playable {

    public void play();

}
```

6.2 Triển khai interface cho CompactDisc

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

public class CompactDisc extends Media implements Playable {

    private String artist;
    private ArrayList<Track> tracks;


    public String getArtist() {
        return artist;
    }


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

    public void addTrack(Track track) {
        if (!tracks.contains(track)) {
            tracks.add(track);
        } else {
            System.out.println("Track already exists in CD.");
        }
    }

    public void removeTrack(Track track) {
        if (tracks.contains(track)) {
            tracks.remove(track);
        } else {
            System.out.println("Track does not exist in CD.");
        }
    }

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

```

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

6.3 Triển khai interface với DigitalVideoDisc

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

public class DigitalVideoDisc extends Disc implements Playable {


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


    @Override
    public String toString() {
        return "DVD: " + this.getTitle() +
                " - Category: " + this.getCategory() +
                " - Director: " + this.getDirector() +
                " - DVD length: " + this.getLength() +
                " - Cost: " + this.getCost() + "$";
    }

    public boolean isMatch(String title) {
        return this.getTitle().toLowerCase().contains(title.toLowerCase());
    }

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

6.4 Triển khai interface với Track

```java
package hust.soict.hedspi.aims.media;
public class Track implements Playable {

    private String title;
    private int length;

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

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

    public String getTitle() {
        return title;
    }

    public int getLength() {
        return length;
    }

}
```

7. Cập nhật cart để hoạt động với Media

```java
package hust.soict.hedspi.aims.cart.Cart;
import java.util.*;

import hust.soict.hedspi.aims.media.Media;

public class Cart {

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

    public int qtyOrdered = 0;

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

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

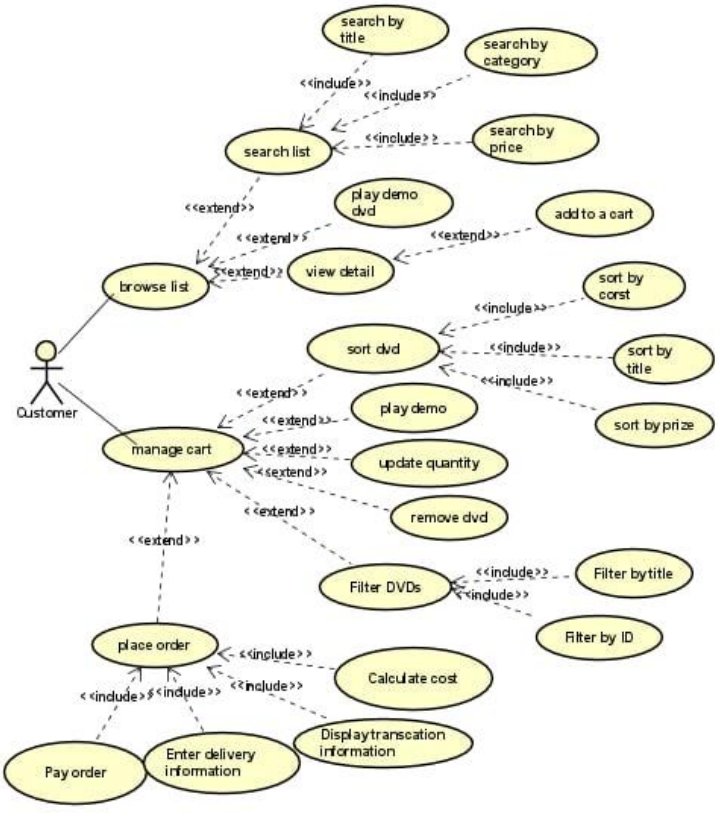
```java
    public void searchByCategory(String category) {
        boolean found = false;
        for (Media media : itemsOrdered) {
            if (media.getCategory().equalsIgnoreCase(category)) {
                System.out.println("Found " + media);
                found = true;
            }
        }
        if (!found) {
            System.out.println("Sorry, no media matching the \"" + category + "\" category were found!");
        }
    }


    public void searchByPrice(float maxCost) {
        boolean matchFound = false;
        for (Media media : itemsOrdered) {
            if (media.getCost() <= maxCost) {
                System.out.println("Found " + media);
                matchFound = true;
            }
        }
        if (!matchFound) {
            System.out.println("Sorry, no media were found that match the maximum cost provided!");
        }
    }

    public void searchByPrice(float minCost, float maxCost) {
        boolean matchFound = false;
        for (Media media : itemsOrdered) {
            if (media.getCost() >= minCost && media.getCost() <= maxCost) {
                System.out.println("Found " + media);
                matchFound = true;
            }
        }
        if (!matchFound) {
            System.out.println("Sorry, no media were found that match the cost range between your specified minimum and maximum!");
        }
    }
```

```
    public void searchByPrice(float minCost, float maxCost) {
        boolean matchFound = false;
        for (Media media : itemsOrdered) {
            if (media.getCost() >= minCost && media.getCost() <= maxCost) {
                System.out.println("Found " + media);
                matchFound = true;
            }
        }
        if (!matchFound) {
            System.out.println("Sorry, no media were found that match the cost range between your specified minimum and
        }
    }


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

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


    public void print() {
        System.out.println("**********CART**********");
        System.out.println("Ordered Items:");
        int i =0;
        for (Media media : itemsOrdered) {
            i +=1;
            System.out.println(i + ". " + media);
        }
        System.out.println("Total cost: " + totalCost());
        System.out.println("**************************************************");
    }

}
```

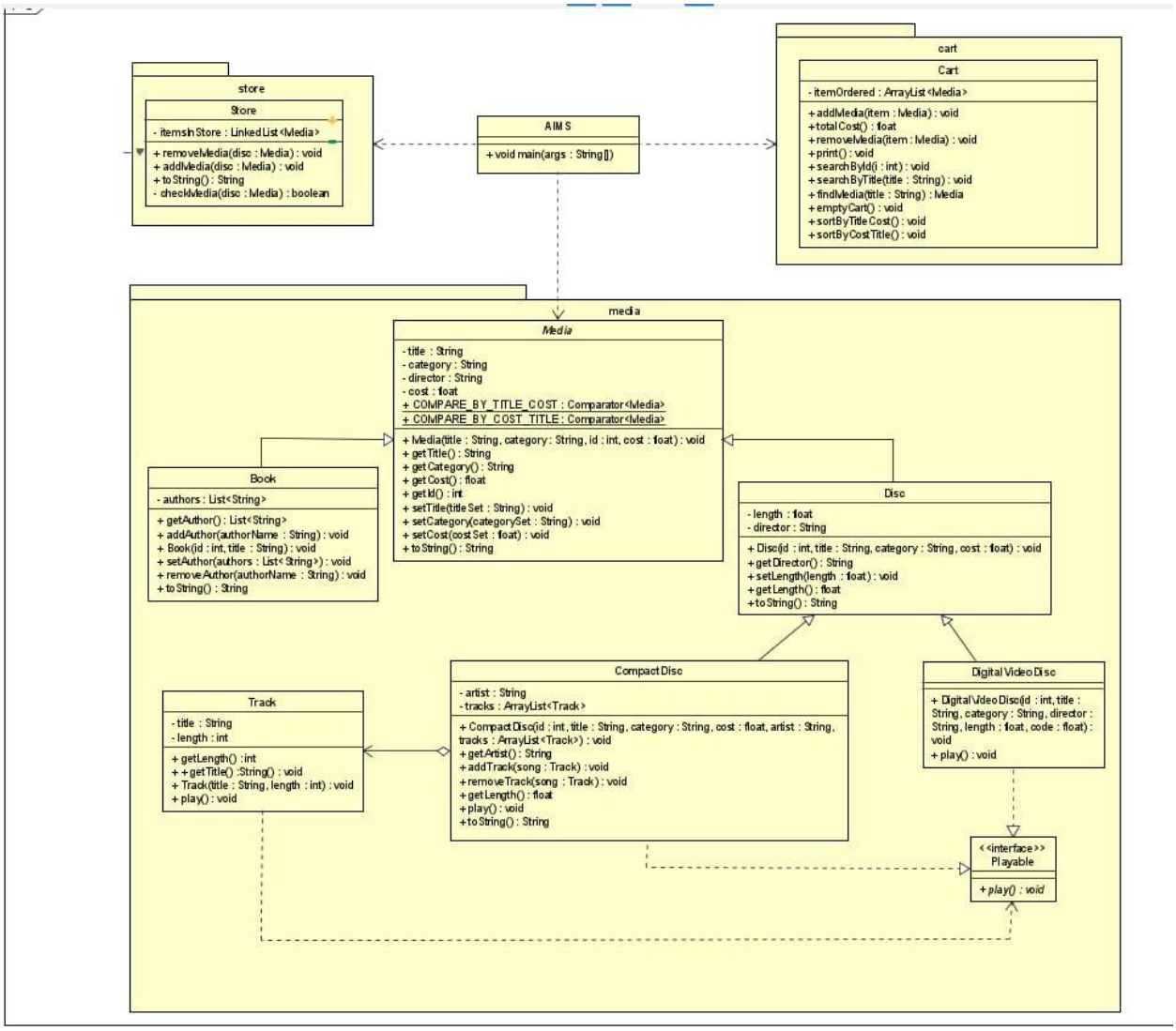
8. Cập nhật Store để hoạt động với Media

```java
package hust.soict.hedspi.aims.store.Store;

import java.util.ArrayList;

import hust.soict.hedspi.aims.media.Media;

public class Store {

    private ArrayList<Media> itemsInStore = new ArrayList<Media>();

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

    public void print() {
        if (itemsInStore.size() == 0) {
            System.out.println("The store is empty!");
        } else {
            System.out.println("**********STORE INVENTORY**********");
            int i = 0;
            for (Media media : itemsInStore) {
                i +=1;
                System.out.println(i +" - " + media);
            }
            System.out.println("**********************************************************");
        }
    }

}
```

9. Các hàm tạo của toàn bộ lớp và lớp cha



- Class Diagram



10. Unique item in a list

10.1 Thêm phương thức equals cho lớp Media

```java
    // Getter method
    public int getId() {
        return id;
    }
    public String getTitle() {
        return title;
    }
    public String getCategory() {
        return category;
    }
    public float getCost() {
        return cost;
    }

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

    // Check is title match
    public boolean isMatch(String title) {
        return this.getTitle().toLowerCase().contains(title.toLowerCase());
    }

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

10.2 Thêm phương thức equals cho lớp Track

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

public class Track implements Playable {

    private String title;
    private int length;

    // Constructor
    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }

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

    // Getter method
    public String getTitle() {
        return title;
    }
    public int getLength() {
        return length;
    }

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

11. Đa hình với phương thức toString()

```java
package hust.soict.hedspi.test;

import java.util.*;

import hust.soict.hedspi.aims.media.*;

public class polymorphism {

    public static void main(String[] args) {
        List<Media> mediae = new ArrayList<Media>();

        DigitalVideoDisc dvd = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        Book book = new Book("Tam Quoc Dien Nghia", "Novel", 20.00f);


        CompactDisc cd = new CompactDisc("Hay trao cho anh", "Music","Son Tung MTP", 1600.91f);
        Track track1 = new Track("We don't talk anymore", 161);
        Track track2 = new Track("Despacito", 403);
        Track track3 = new Track("Shape of you", 300);

        cd.addTrack(track1);
        cd.addTrack(track2);
        cd.addTrack(track3);

        mediae.add(cd);
        mediae.add(book);
        mediae.add(dvd);

        for (Media media : mediae) {
            System.out.println(media.toString());
        }
    }
```

- Kết quả :

```
run:
Track: We don't talk anymore has been added to CD: Hay trao cho anh
Track: Despacito has been added to CD: Hay trao cho anh
Track: Shape of you has been added to CD: Hay trao cho anh
3 - CD: Hay trao cho anh - Category: Music - ArtistSon Tung MTP - Length: 864 seconds - Cost: 1600.91$
2 - Book: Tam Quoc Dien Nghia - Category: Novel - Cost: 20.0$
1 - DVD: The Lion King - Category: Animation - Director: Roger Allers - DVD length: 87 - Cost: 19.95$
BUILD SUCCESSFUL (total time: 0 seconds)
```

12. Sort media in the cart

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

import java.util.Comparator;

public abstract class Media implements Comparable<Media> {

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


    private static int nbMedia = 0;
    private int id;

    private String title;
    private String category;
    private float cost;

    // Constructor
```

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

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media>{

    @Override
    public int compare(Media o1, Media o2) {
        // Compare by cost
        int costComparison = Double.compare(o2.getCost(), o1.getCost());
        if (costComparison != 0) {
            return costComparison;
        }
        // Compare by title
        return o1.getTitle().compareTo(o2.getTitle());
    }
}
```

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

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {
    @Override
    public int compare(Media o1, Media o2) {
        // Compare by title
        int titleComparison = o1.getTitle().compareTo(o2.getTitle());
        if (titleComparison != 0) {
            return titleComparison;
        }
        // Compare by cost
        return Double.compare(o2.getCost(), o1.getCost());
    }
```

13. Tạo 1 ứng dụng bảng điều chỉnh trong Aims

```java
package hust.soict.hedspi.aims.Aims;
import hust.soict.hedspi.aims.cart.Cart.Cart;
import hust.soict.hedspi.aims.media.*;
import hust.soict.hedspi.aims.store.Store.Store;
import java.util.*;

public class Aims {
    private static Store store = new Store();
    private static Cart cart = new Cart();

    public static void main(String[] args) {

        // Init add media to the store
        initSetup();

        boolean exit = false;
        // CLI
        while (!exit) {

            showMenu();

            Scanner scanner = new Scanner(System.in);
            int option = scanner.nextInt();
            scanner.nextLine();

            switch (option) {
                case 0:
                    exit = true;
                    System.out.println("Good bye!");
                    break;
                case 1:
                    clearConsole();
                    storeMenu(scanner);
                    break;
                case 2:
                    clearConsole();
                    updateStoreMenu(scanner);
                    break;
                case 3:
                    clearConsole();
                    cartMenu(scanner);
                    break;
                default:
                    clearConsole();
                    System.out.println("Invalid option, please choose again.");
                    break;
            }

        }
```

```java
public static void clearConsole() {
    for (int i = 0; i < 50; i++) {
        System.out.println();
    }
}
// init store setup
public static void initSetup() {

    DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
    DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star War", "Science Fiction", "George Lucas", 87, 24.95f);
    DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
    store.addMedia(dvd1);
    store.addMedia(dvd2);
    store.addMedia(dvd3);


    Book book = new Book("The Valley of Fear", "Detective", 20.00f);
    Book book1 = new Book("A Living Remedy: A Memoir", "Biography", 202.00f);
    Book book2 = new Book("On the Origin of Time: Stephen Hawking's Final Theory", "Science", 120.00f);
    store.addMedia(book);
    store.addMedia(book1);
    store.addMedia(book2);


    CompactDisc cd1 = new CompactDisc("Adele - 30", "Music","Adele", 1500.98f);
    Track track1CD1 = new Track("All Night Parking (interlude)", 161);
    Track track2CD1 = new Track("To Be Loved", 403);
    Track track3CD1 = new Track("Woman Like Me", 300);
    cd1.addTrack(track1CD1);
    cd1.addTrack(track2CD1);
    cd1.addTrack(track3CD1);

    CompactDisc cd2 = new CompactDisc("The Gods We Can Touch", "Music","Aurora", 2000.22f);
    Track track1CD2 = new Track("Everything Matters", 180+34);
    Track track2CD2 = new Track("Blood in the Wine", 180+30);
    Track track3CD2 = new Track("Artemis", 60*2+39);
    cd2.addTrack(track1CD2);
    cd2.addTrack(track2CD2);
    cd2.addTrack(track3CD2);

    CompactDisc cd3 = new CompactDisc("Purpose", "Music","Justin Bieber", 1000.98f);
    Track track1CD3 = new Track("The Feeling", 4*60+5);
    Track track2CD3 = new Track("No Sense", 4*60+35);
    cd3.addTrack(track1CD3);
    cd3.addTrack(track2CD3);

    store.addMedia(cd1);
    store.addMedia(cd2);
    store.addMedia(cd3);
```

```java
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");
}
public static void storeMenu(Scanner scanner) {
    boolean back = false;
    while (!back) {
        store.print();
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. See a media's details");
        System.out.println("2. Add a media to cart");
        System.out.println("3. Play a media");
        System.out.println("4. See current cart");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3-4");
        int option = scanner.nextInt();
        scanner.nextLine();
        switch (option) {
            case 0:
                clearConsole();
                back = true;
                break;
            case 1:
                boolean foundDetails = false;
                while (!foundDetails) {
                    System.out.println("Enter the title of the media (type 0 to stop): ");
                    String title = scanner.nextLine();
                    if (title.equals("0")) {
                        clearConsole();
                        break;
                    }
                    Media media = store.search(title);
                    if (media != null) {
                        clearConsole();
                        System.out.println("Details: ");
                        System.out.println(media);
                        mediaDetailsMenu(scanner, media);
                        foundDetails = true;
                    } else {
                        System.out.println("***MEDIA NOT FOUND***");
                    }
                        System.out.println("***MEDIA NOT FOUND***");
                    }
                }
                break;
            case 2:
                boolean foundToAdd = false;
                while (!foundToAdd) {
                    System.out.println("Enter the title of the media (type 0 to stop): ");
                    String title = scanner.nextLine();
                    if (title.equals("0")) {
                        clearConsole();
                        break;
                    }
                    Media media = store.search(title);
                    if (media != null) {
                        cart.addMedia(media);
                        foundToAdd = true;
                    } else {
                        System.out.println("***MEDIA NOT FOUND***");
                    }
                }
                break;
            case 3:
                boolean foundToPlay = false;
                while (!foundToPlay) {
                    System.out.println("Enter the title of the media (type 0 to stop): ");
                    String title = scanner.nextLine();
                    if (title.equals("0")) {
                        clearConsole();
                        break;
                    }
                    Media media = store.search(title);
                    if (media != null) {
                        if (media instanceof Disc || media instanceof CompactDisc) {
                            media.play();
                        } else {
                            System.out.println("This type of media is not supported!");
                        }
                        foundToPlay = true;
                    } else {
                        System.out.println("***MEDIA NOT FOUND***");
                    }
                }
                break;
            case 4:
                clearConsole();
                cartMenu(scanner);
                break;
```

```
                default:
                    clearConsole();
                    System.out.println("Invalid option, please choose again.");
                    break;
            }
        }
    }
    public static void mediaDetailsMenu(Scanner scanner, Media media) {
        boolean back = false;
        while (!back) {
            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 option = scanner.nextInt();
            scanner.nextLine();
            switch (option) {
                case 0:
                    clearConsole();
                    back = true;
                    break;
                case 1:
                    cart.addMedia(media);
                    break;
                case 2:
                    if (media instanceof Disc || media instanceof CompactDisc) {
                        media.play();
                    } else {
                        System.out.println("This type of media is not supported!");
                    }
                    break;
                default:
                    clearConsole();
                    System.out.println("Invalid option, please choose again.");
                    break;
            }
        }
    }
    public static void cartMenu(Scanner scanner) {
        boolean back = false;
        while (!back) {
            cart.print();
            System.out.println("Options: ");
            System.out.println("---------------------------------");
            System.out.println("1. Filter medias in cart");
            System.out.println("2. Sort medias in cart");
            System.out.println("3. Play a media");
            System.out.println("4. Remove media from cart");
            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");
            int option = scanner.nextInt();
            scanner.nextLine();
            switch (option) {
                case 0:
                    clearConsole();
                    back = true;
                    break;
                case 1:
                    System.out.println("Filter medias in cart by (1) id or (2) title:");
                    int filterOption = scanner.nextInt();
                    scanner.nextLine();
                    boolean found = false;
                    while (!found) {
                        if (filterOption == 1) {
                            System.out.println("Enter the id to filter (type 0 to stop):");
                            int id = scanner.nextInt();
                            scanner.nextLine();
                            if (id == 0) {
                                clearConsole();
                                break;
                            }
                            cart.searchByID(id);
                            found = true;
                        } else if (filterOption == 2) {
                            System.out.println("Enter the title to filter (type 0 to stop):");
                            String title = scanner.nextLine();
                            if (title.equals("0")) {
                                clearConsole();
                                break;
                            }
                            cart.searchByTitle(title);
                            found = true;
                        } else if (filterOption == 0) {
                            clearConsole();
                            break;
                        } else {
                            System.out.println("Invalid option.");
                        }
                    }
                    break;
                case 2:
                    System.out.println("Sort medias in cart by (1) title or (2) cost:");
                    int sortOption = scanner.nextInt();
                    scanner.nextLine();
                    if (sortOption == 1) {
```

```
                cart.sortMediaByTitle();
            } else if (sortOption == 2) {
                cart.sortMediaByCost();
            } else {
                System.out.println("Invalid option.");
            }
            break;
        case 3:
            boolean foundToRemove = false;
            while (!foundToRemove) {
                System.out.println("Enter the title of the media (type 0 to stop): ");
                String title = scanner.nextLine();
                if (title.equals("0")) {
                    clearConsole();
                    break;
                }
                Media media = cart.searchToRemove(title);
                if (media != null) {
                    clearConsole();
                    cart.removeMedia(media);
                    foundToRemove = true;
                } else {
                    System.out.println("***MEDIA NOT FOUND***");
                }
            }
            break;
        case 4:
            boolean foundToPlay = false;
            while (!foundToPlay) {
                System.out.println("Enter the title of the media (type 0 to stop): ");
                String title = scanner.nextLine();
                if (title.equals("0")) {
                    clearConsole();
                    break;
                }
                Media media = store.search(title);
                if (media != null) {
                    if (media instanceof Disc || media instanceof CompactDisc) {
                        media.play();
                    } else {
                        System.out.println("This type of media is not supported!");
                    }
                    foundToPlay = true;
                } else {
                    System.out.println("***MEDIA NOT FOUND***");
                }
            }
            break;
```

```
    public static void updateStoreMenu(Scanner scanner) {
        boolean back = false;
        while (!back) {
            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("Please choose a number: 0-1-2");
            int option = scanner.nextInt();
            scanner.nextLine();
            switch (option) {
                case 0:
                    clearConsole();
                    back = true;
                    break;
                case 1:
                    System.out.println("Enter the category of the media (1) Book, (2) CD, (3) DVD or (0) exit:");
                    int categoryChoice = scanner.nextInt();
                    scanner.nextLine();

                    if (categoryChoice == 1) {
                        System.out.println("Enter book title: ");
                        String bookTitle = scanner.nextLine();
                        System.out.println("Enter book category: ");
                        String bookCategory = scanner.nextLine();
                        System.out.println("Enter book cost: ");
                        Float bookCost = scanner.nextFloat();
                        scanner.nextLine();

                        Book newBook = new Book(bookTitle, bookCategory, bookCost);
                        store.addMedia(newBook);
                    } else if (categoryChoice == 2) {
                        System.out.println("Enter CD title: ");
                        String cdTitle = scanner.nextLine();
                        System.out.println("Enter CD category: ");
                        String cdCategory = scanner.nextLine();
                        System.out.println("Enter CD artist: ");
                        String cdArtist = scanner.nextLine();
                        System.out.println("Enter CD cost: ");
                        Float cdCost = scanner.nextFloat();
                        scanner.nextLine();

                        CompactDisc newCD = new CompactDisc(cdTitle, cdCategory, cdArtist, cdCost);


                        System.out.println("Do you want to add tracks to your CD? (1) Yes (0) No:");
                        int addTrack = scanner.nextInt();
```

```
                    scanner.nextLine();
                    for (int i = 0; i < numTrack; i++) {
                        System.out.println("Your " + (i+1) + " track: ");
                        System.out.println("Enter track title: ");
                        String trackTitle = scanner.nextLine();
                        System.out.println("Enter track length: ");
                        int trackLength = scanner.nextInt();
                        scanner.nextLine();

                        Track newTrack = new Track(trackTitle, trackLength);
                        newCD.addTrack(newTrack);
                    }
                    store.addMedia(newCD);
                } else if (addTrack == 0) {
                    store.addMedia(newCD);
                }
            } else if (categoryChoice == 3) {
                System.out.println("Enter DVD title: ");
                String dvdTitle = scanner.nextLine();
                System.out.println("Enter DVD category: ");
                String dvdCategory = scanner.nextLine();
                System.out.println("Enter book cost: ");
                Float dvdCost = scanner.nextFloat();
                scanner.nextLine();

                DigitalVideoDisc newDVD = new DigitalVideoDisc(dvdTitle, dvdCategory, dvdCost);
                store.addMedia(newDVD);
            } else if (categoryChoice == 0) {
                clearConsole();
                break;
            } else {
                System.out.println("Invalid option.");
            }
            break;
        case 2:
            boolean foundToRemove = false;
            while (!foundToRemove) {
                System.out.println("Enter the title of the media (type 0 to stop): ");
                String titleForRemove = scanner.nextLine();
                if (titleForRemove.equals("0")) {
                    clearConsole();
                    break;
                }
                Media media = store.search(titleForRemove);
                if (media != null) {
                    clearConsole();
```

```
                    clearConsole();
                    break;
                }
                Media media = store.search(titleForRemove);
                if (media != null) {
                    clearConsole();
                    store.removeMedia(media);
                    foundToRemove = true;
                } else {
                    System.out.println("***MEDIA NOT FOUND***");
                }
            }
            break;
        default:
            clearConsole();
            System.out.println("Invalid option, please choose again.");
            break;
        }
    }
  }
}
```