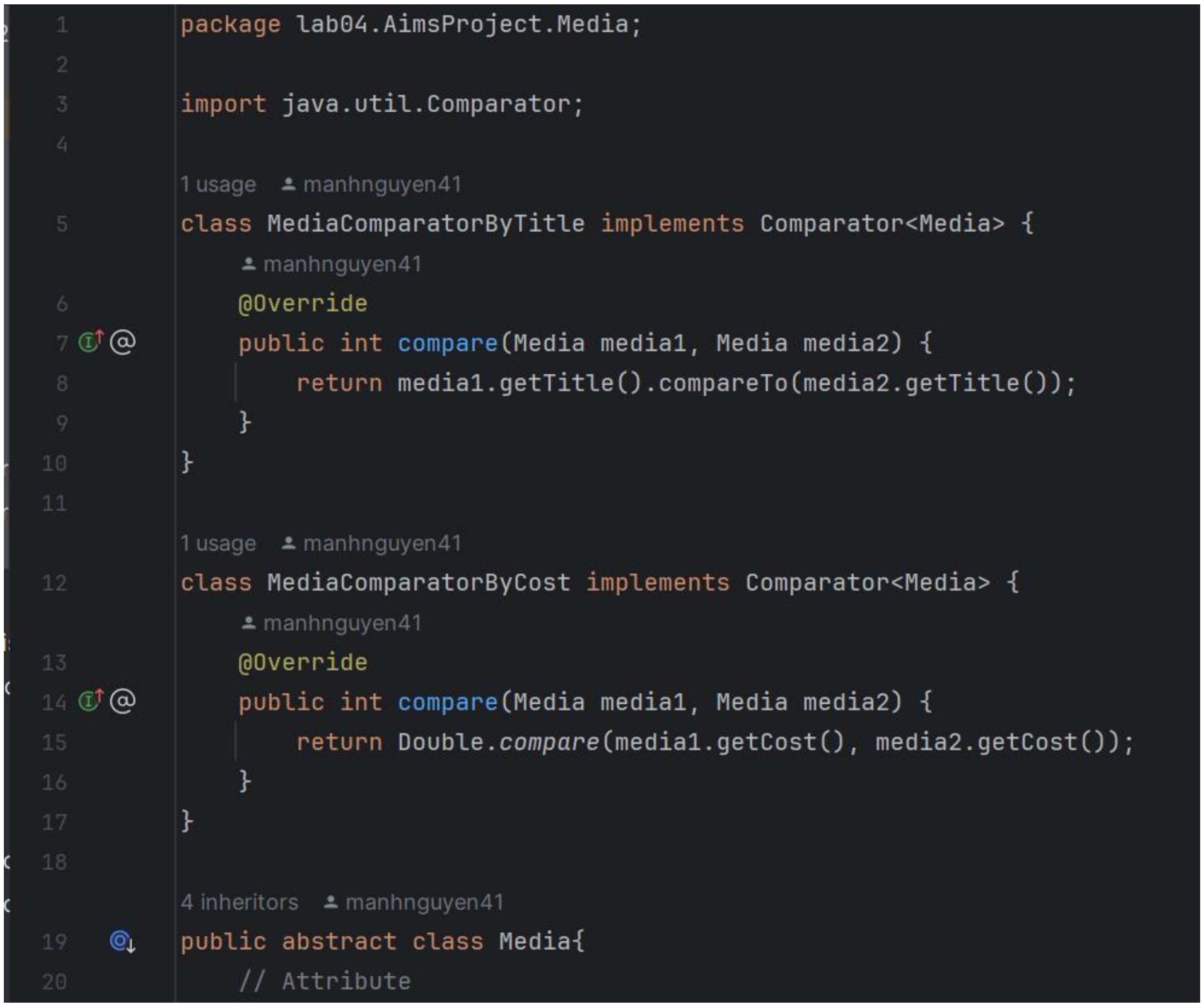
BÁO CÁO THỰC HÀNH LAP 4
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục nội dung

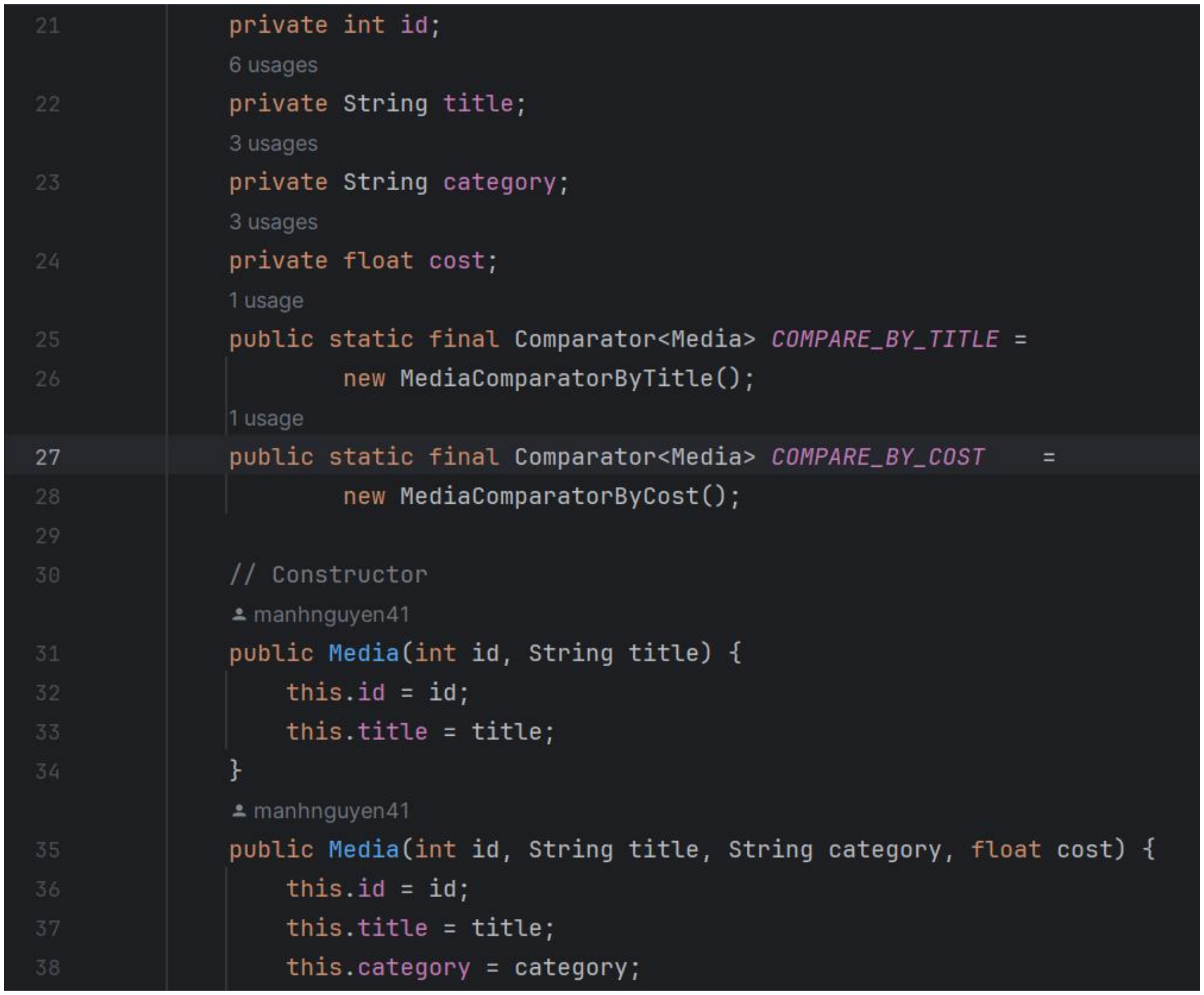
Mục lục hình ảnh

1. Creating the abstract Media class



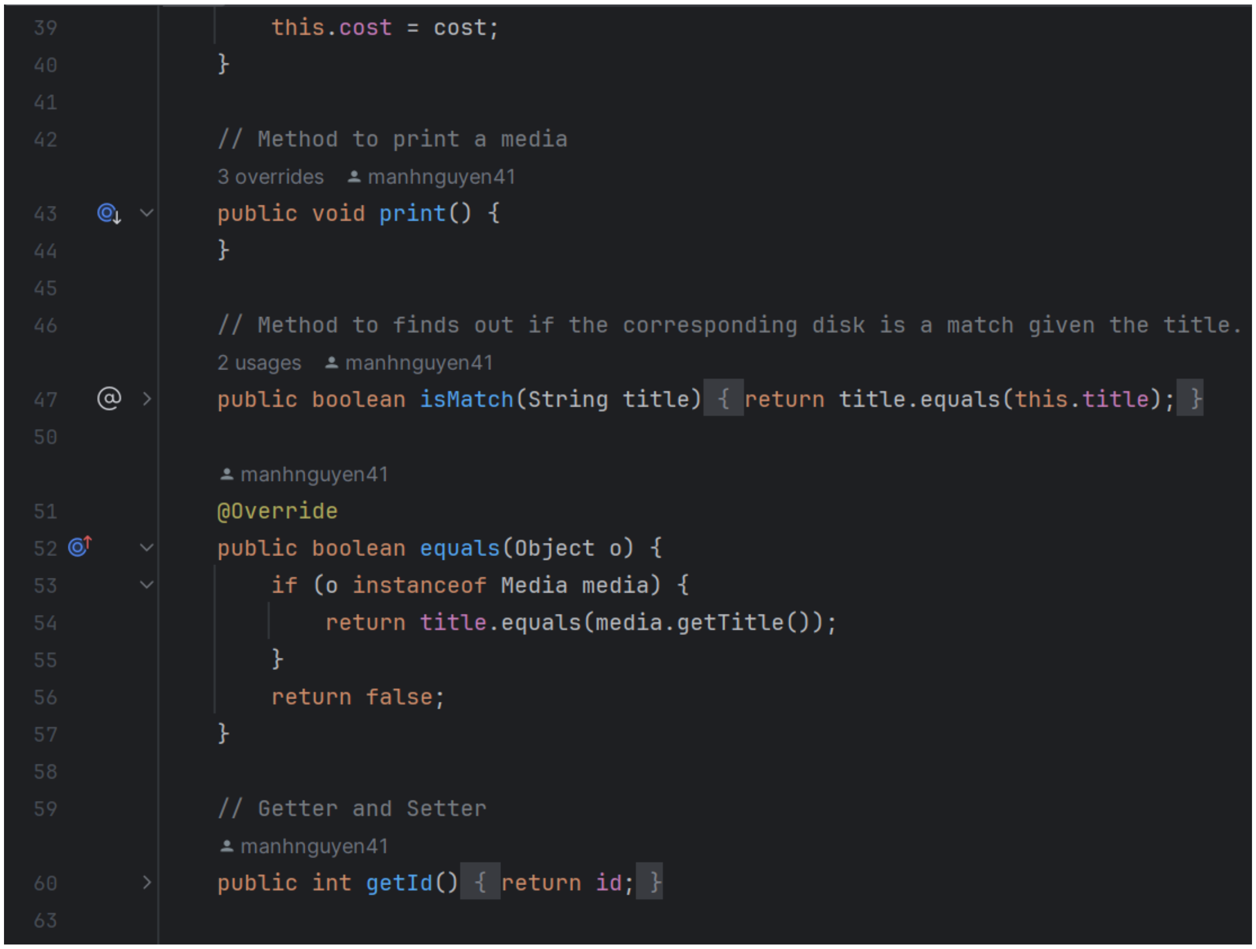
```
package lab04.AimsProject.Media;

import java.util.Comparator;

1 usage  manhnguyen41
class MediaComparatorByTitle implements Comparator<Media> {
     manhnguyen41
    @Override
    public int compare(Media media1, Media media2) {
        return media1.getTitle().compareTo(media2.getTitle());
    }
}

1 usage  manhnguyen41
class MediaComparatorByCost implements Comparator<Media> {
     manhnguyen41
    @Override
    public int compare(Media media1, Media media2) {
        return Double.compare(media1.getCost(), media2.getCost());
    }
}

4 inheritors  manhnguyen41
public abstract class Media{
    // Attribute
```

Figure 1 Media abstract class code



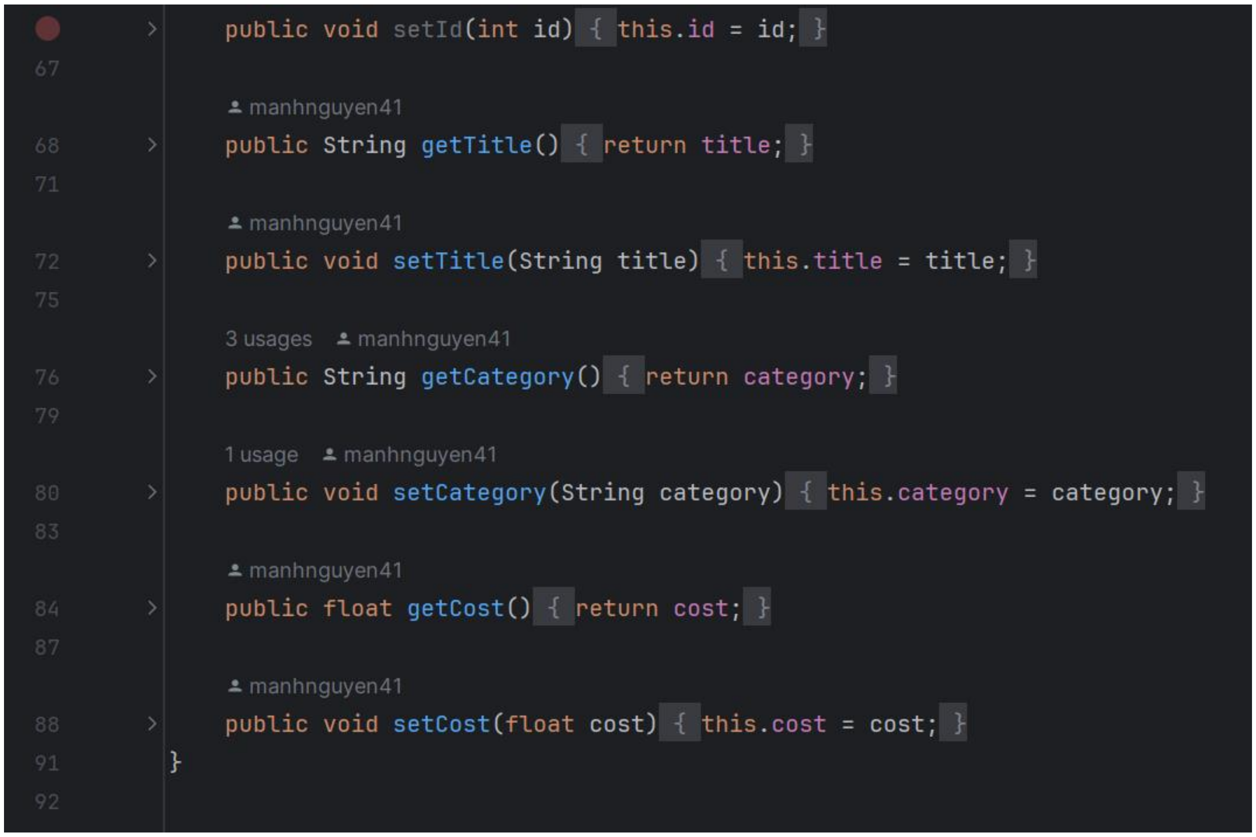
```java
    private int id;
    6 usages
    private String title;
    3 usages
    private String category;
    3 usages
    private float cost;
    1 usage
    public static final Comparator<Media> COMPARE_BY_TITLE =
            new MediaComparatorByTitle();
    1 usage
    public static final Comparator<Media> COMPARE_BY_COST    =
            new MediaComparatorByCost();

    // Constructor
    manhnguyen41
    public Media(int id, String title) {
        this.id = id;
        this.title = title;
    }
    manhnguyen41
    public Media(int id, String title, String category, float cost) {
        this.id = id;
        this.title = title;
        this.category = category;
```

Figure 2 Media abstract class code



```
        this.cost = cost;
    }

    // Method to print a media
    3 overrides  manhnguyen41
    public void print() {
    }

    // Method to finds out if the corresponding disk is a match given the title.
    2 usages  manhnguyen41
    public boolean isMatch(String title) { return title.equals(this.title); }

    manhnguyen41
    @Override
    public boolean equals(Object o) {
        if (o instanceof Media media) {
            return title.equals(media.getTitle());
        }
        return false;
    }

    // Getter and Setter
    manhnguyen41
    public int getId() { return id; }
```

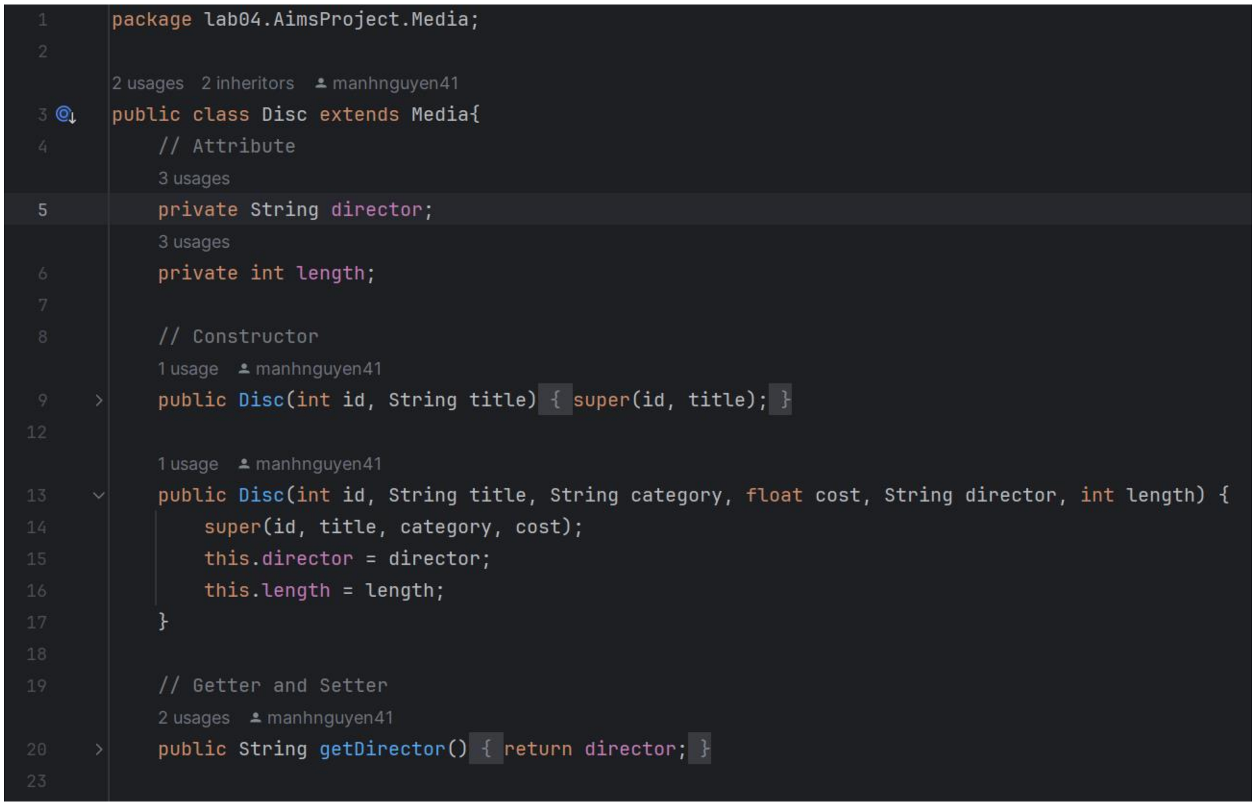
Figure 3 Media abstract class code



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

    manhnguyen41
    public String getTitle() { return title; }

    manhnguyen41
    public void setTitle(String title) { this.title = title; }

    3 usages   manhnguyen41
    public String getCategory() { return category; }

    1 usage   manhnguyen41
    public void setCategory(String category) { this.category = category; }

    manhnguyen41
    public float getCost() { return cost; }

    manhnguyen41
    public void setCost(float cost) { this.cost = cost; }
}
```

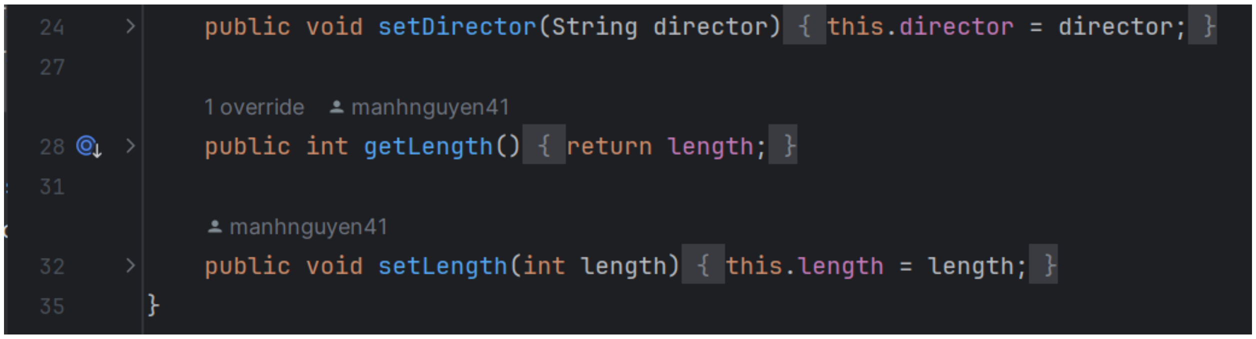
Figure 4 Media abstract class code

2. Create the Disc class extending the Media class



```
package lab04.AimsProject.Media;

2 usages   2 inheritors   manhnguyen41
public class Disc extends Media{
    // Attribute
    3 usages
    private String director;
    3 usages
    private int length;

    // Constructor
    1 usage   manhnguyen41
    public Disc(int id, String title) { super(id, title); }

    1 usage   manhnguyen41
    public Disc(int id, String title, String category, float cost, String director, int length) {
        super(id, title, category, cost);
        this.director = director;
        this.length = length;
    }

    // Getter and Setter
    2 usages   manhnguyen41
    public String getDirector() { return director; }
```

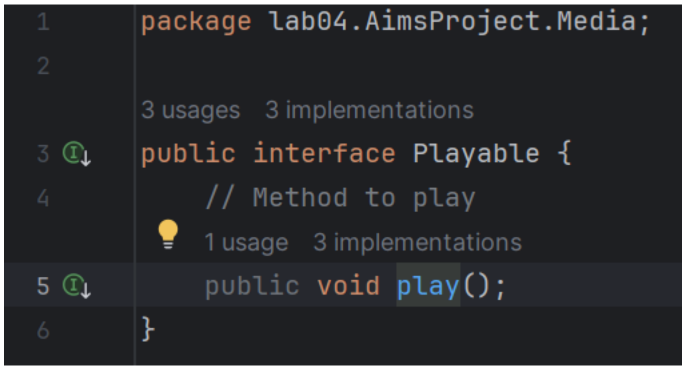
Figure 5 Disc class code



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

    1 override   manhnguyen41
    public int getLength() { return length; }

    manhnguyen41
    public void setLength(int length) { this.length = length; }
}
```

Figure 6 Disc class code

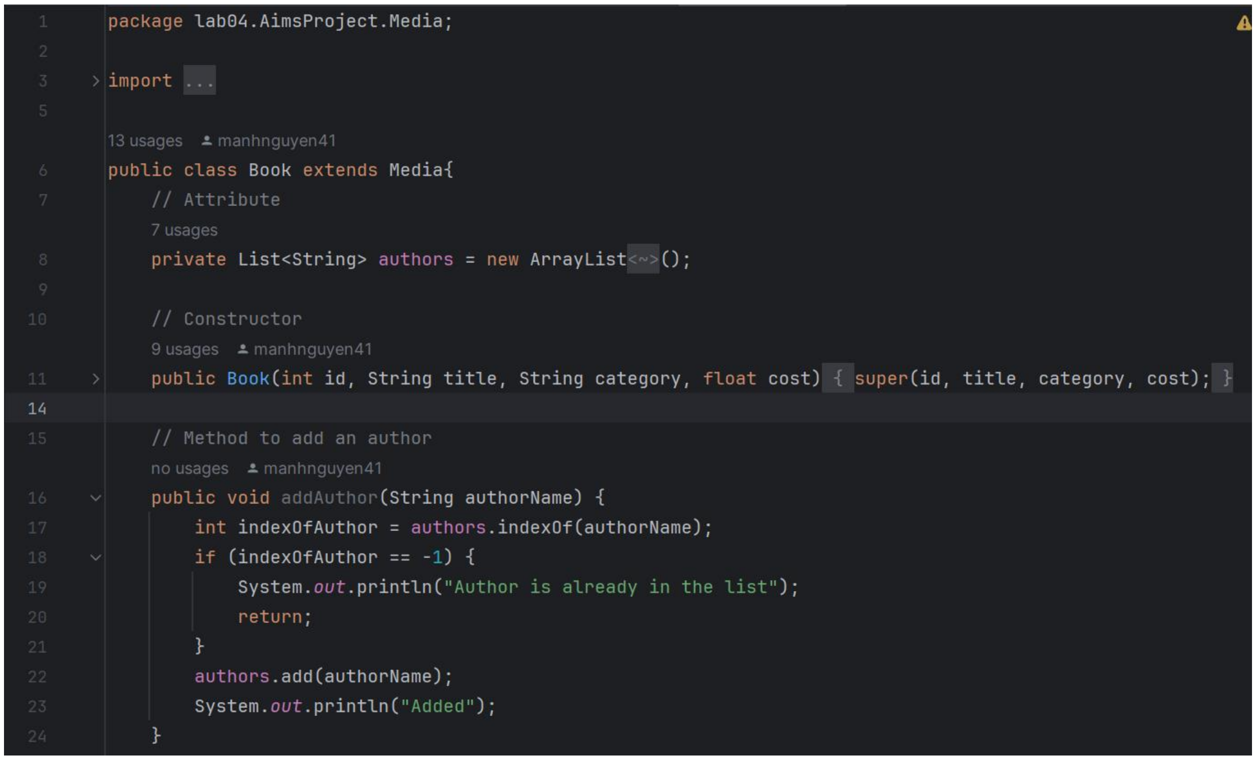
3. Creating the Playable interface



```java
package lab04.AimsProject.Media;

3 usages   3 implementations
public interface Playable {
    // Method to play
    1 usage   3 implementations
    public void play();
}
```

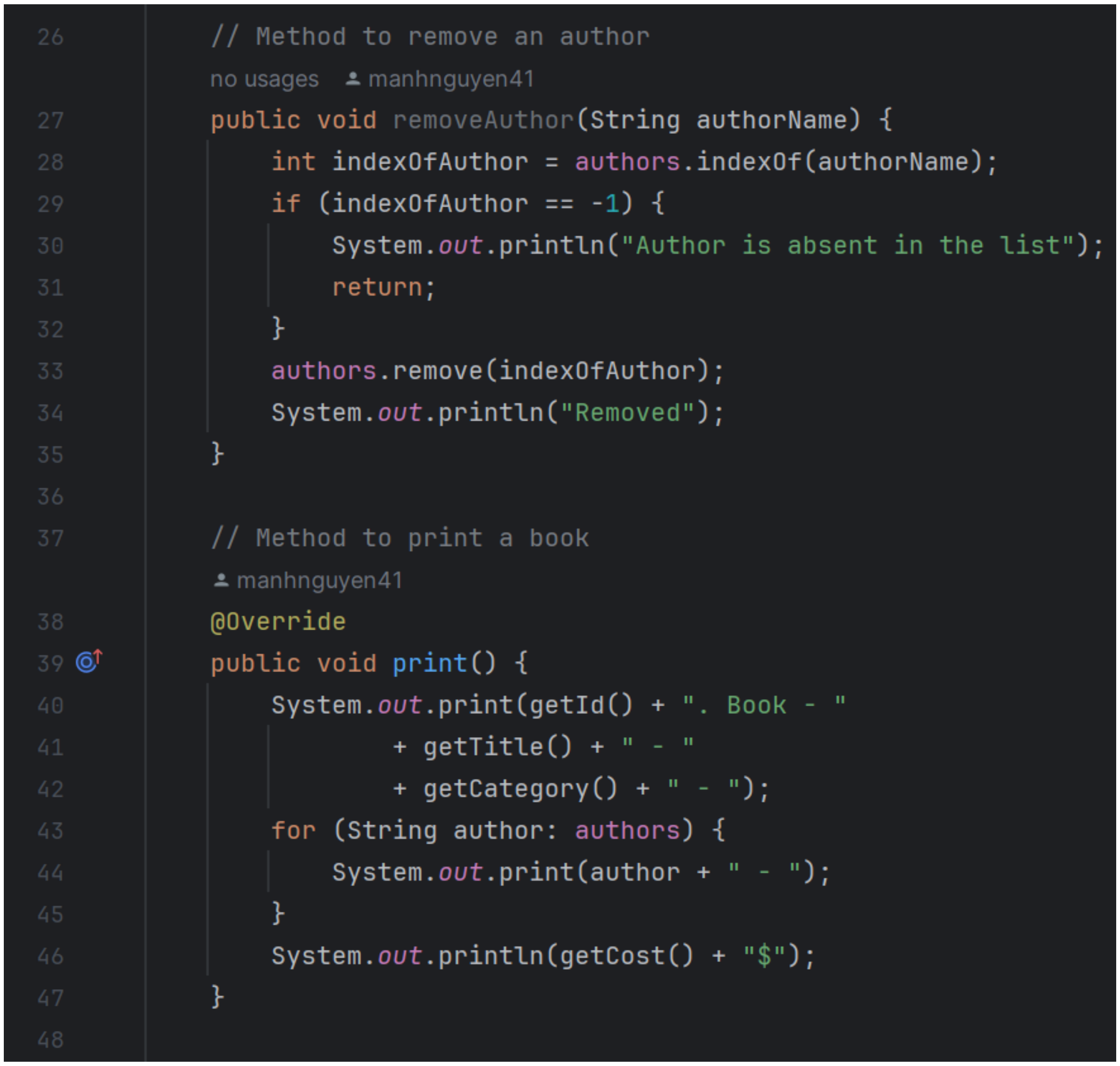
Figure 7 Playable interface code

4. Creating the Book class



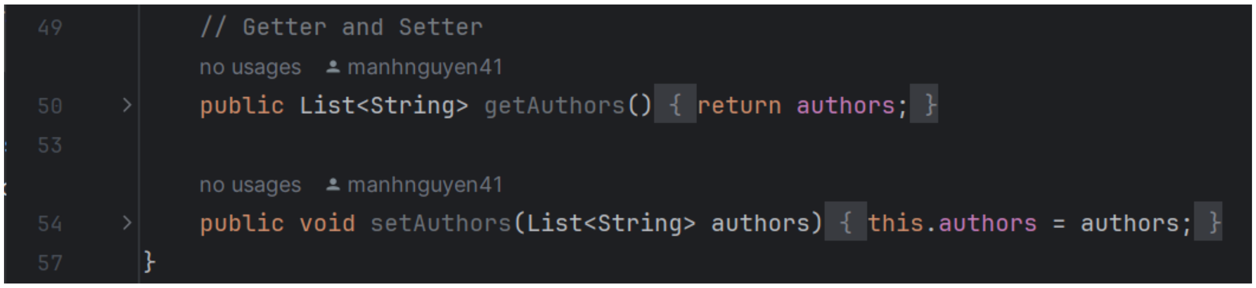
```java
package lab04.AimsProject.Media;

import ...

13 usages  manhnguyen41
public class Book extends Media{
    // Attribute
    7 usages
    private List<String> authors = new ArrayList<~>();

    // Constructor
    9 usages  manhnguyen41
    public Book(int id, String title, String category, float cost) { super(id, title, category, cost); }

    // Method to add an author
    no usages  manhnguyen41
    public void addAuthor(String authorName) {
        int indexOfAuthor = authors.indexOf(authorName);
        if (indexOfAuthor == -1) {
            System.out.println("Author is already in the list");
            return;
        }
        authors.add(authorName);
        System.out.println("Added");
    }
```

Figure 8 Book class code



```
    // Method to remove an author
    no usages  manhnguyen41
    public void removeAuthor(String authorName) {
        int indexOfAuthor = authors.indexOf(authorName);
        if (indexOfAuthor == -1) {
            System.out.println("Author is absent in the list");
            return;
        }
        authors.remove(indexOfAuthor);
        System.out.println("Removed");
    }

    // Method to print a book
    manhnguyen41
    @Override
    public void print() {
        System.out.print(getId() + ". Book - "
                + getTitle() + " - "
                + getCategory() + " - ");
        for (String author: authors) {
            System.out.print(author + " - ");
        }
        System.out.println(getCost() + "$");
    }

```

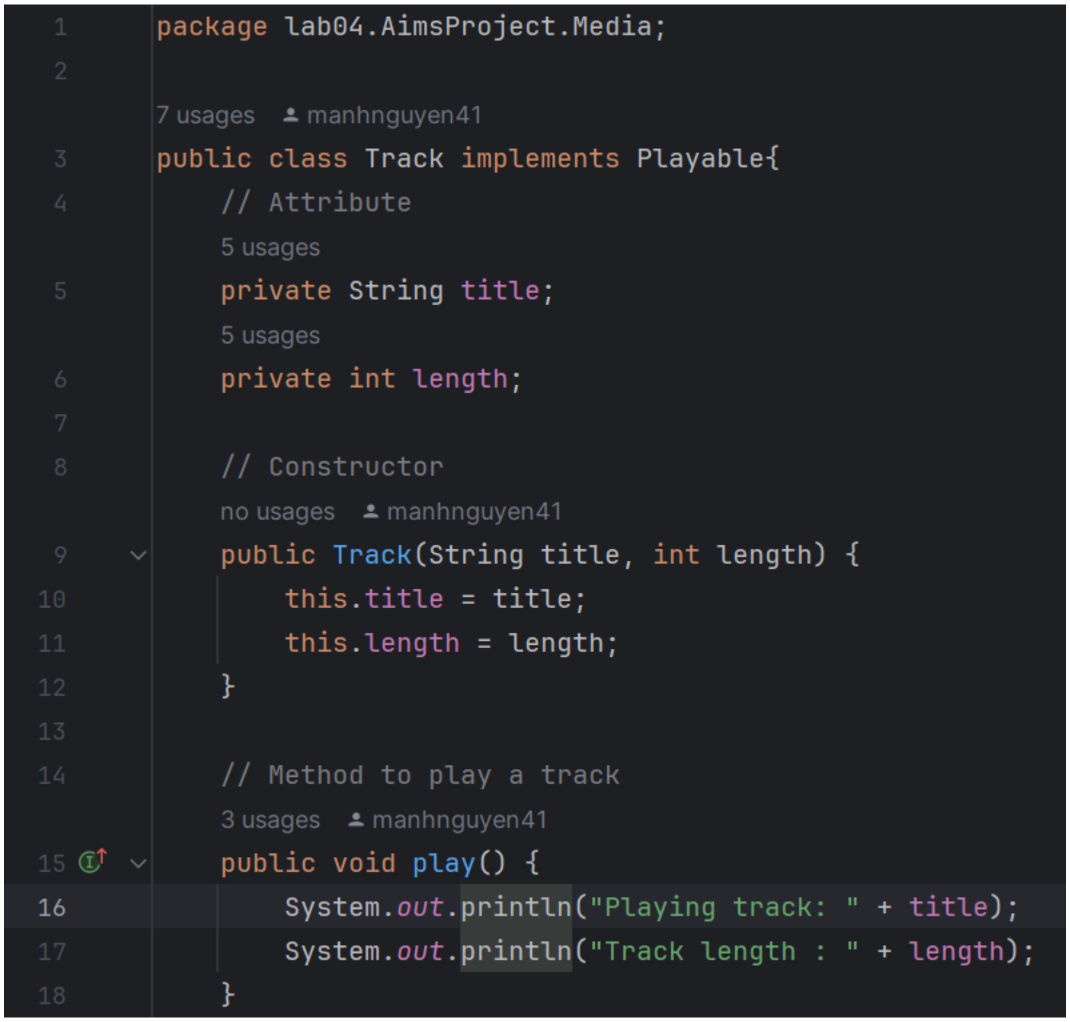
Figure 9 Book class code



```
    // Getter and Setter
    no usages  manhnguyen41
    public List<String> getAuthors() { return authors; }

    no usages  manhnguyen41
    public void setAuthors(List<String> authors) { this.authors = authors; }
}
```

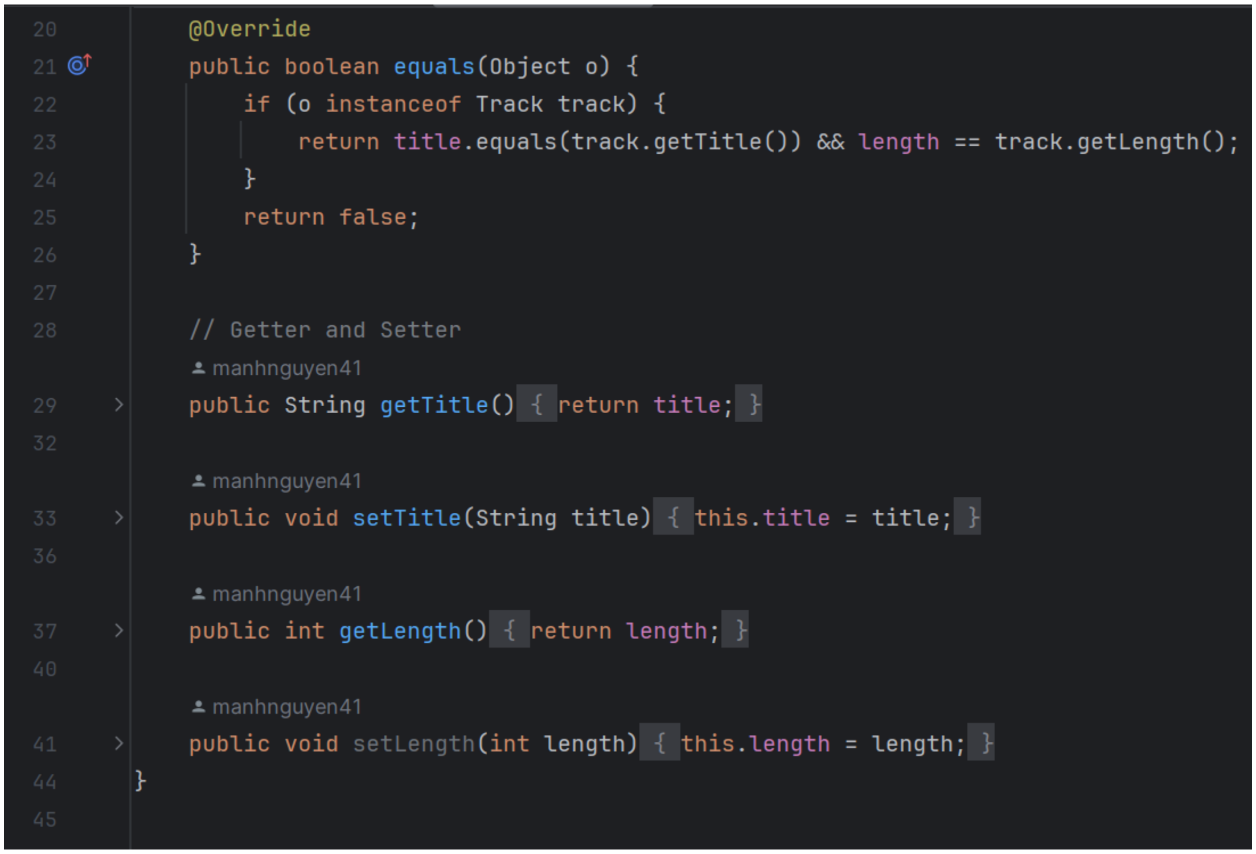
Figure 10 Book class code

5. Creating the Track class



```
package lab04.AimsProject.Media;

7 usages  manhnguyen41
public class Track implements Playable{
    // Attribute
    5 usages
    private String title;
    5 usages
    private int length;

    // Constructor
    no usages  manhnguyen41
    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }

    // Method to play a track
    3 usages  manhnguyen41
    public void play() {
        System.out.println("Playing track: " + title);
        System.out.println("Track length : " + length);
    }
```

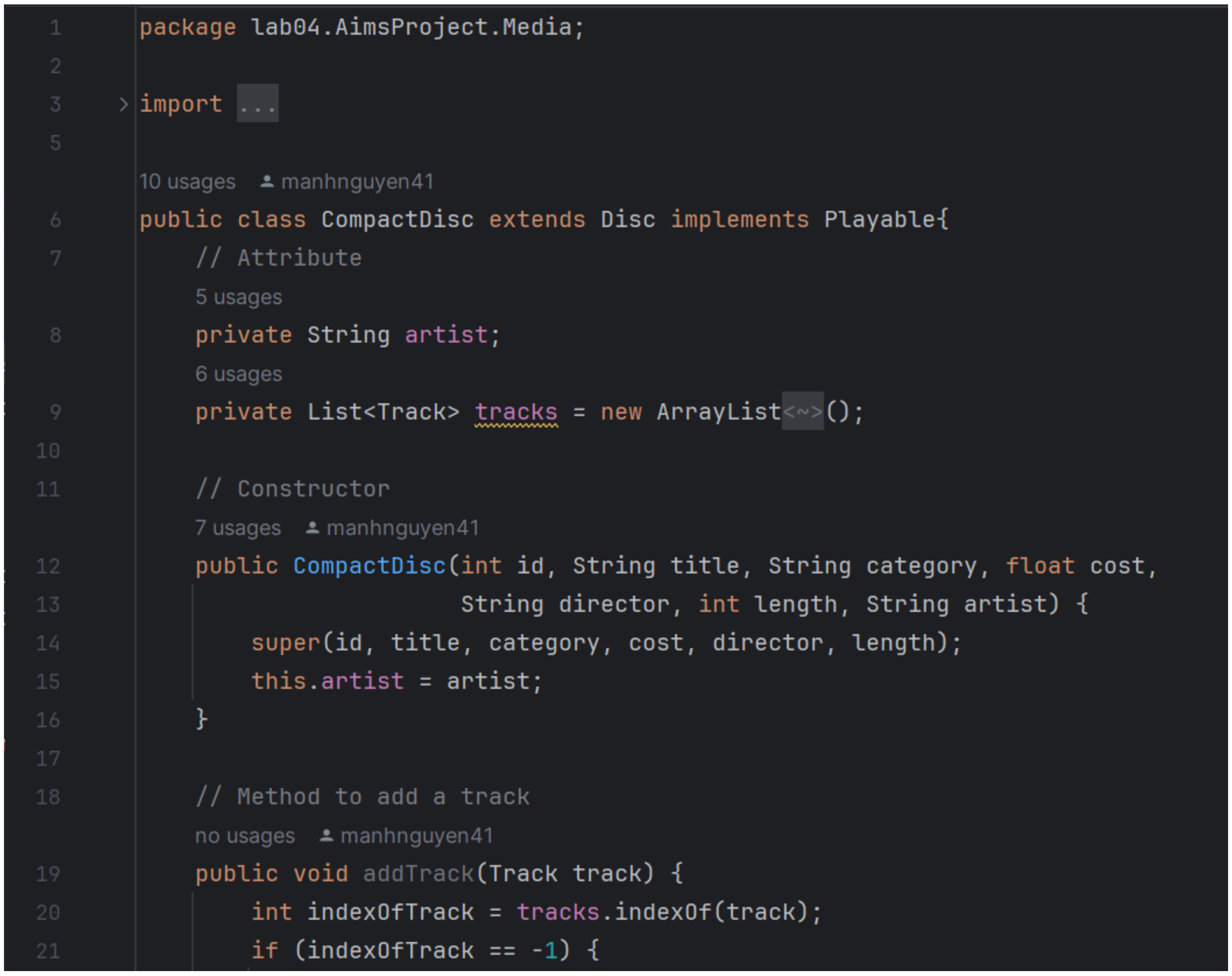
Figure 11 Track class code



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

    // Getter and Setter
    manhnguyen41
    public String getTitle() { return title; }

    manhnguyen41
    public void setTitle(String title) { this.title = title; }

    manhnguyen41
    public int getLength() { return length; }

    manhnguyen41
    public void setLength(int length) { this.length = length; }
}
```

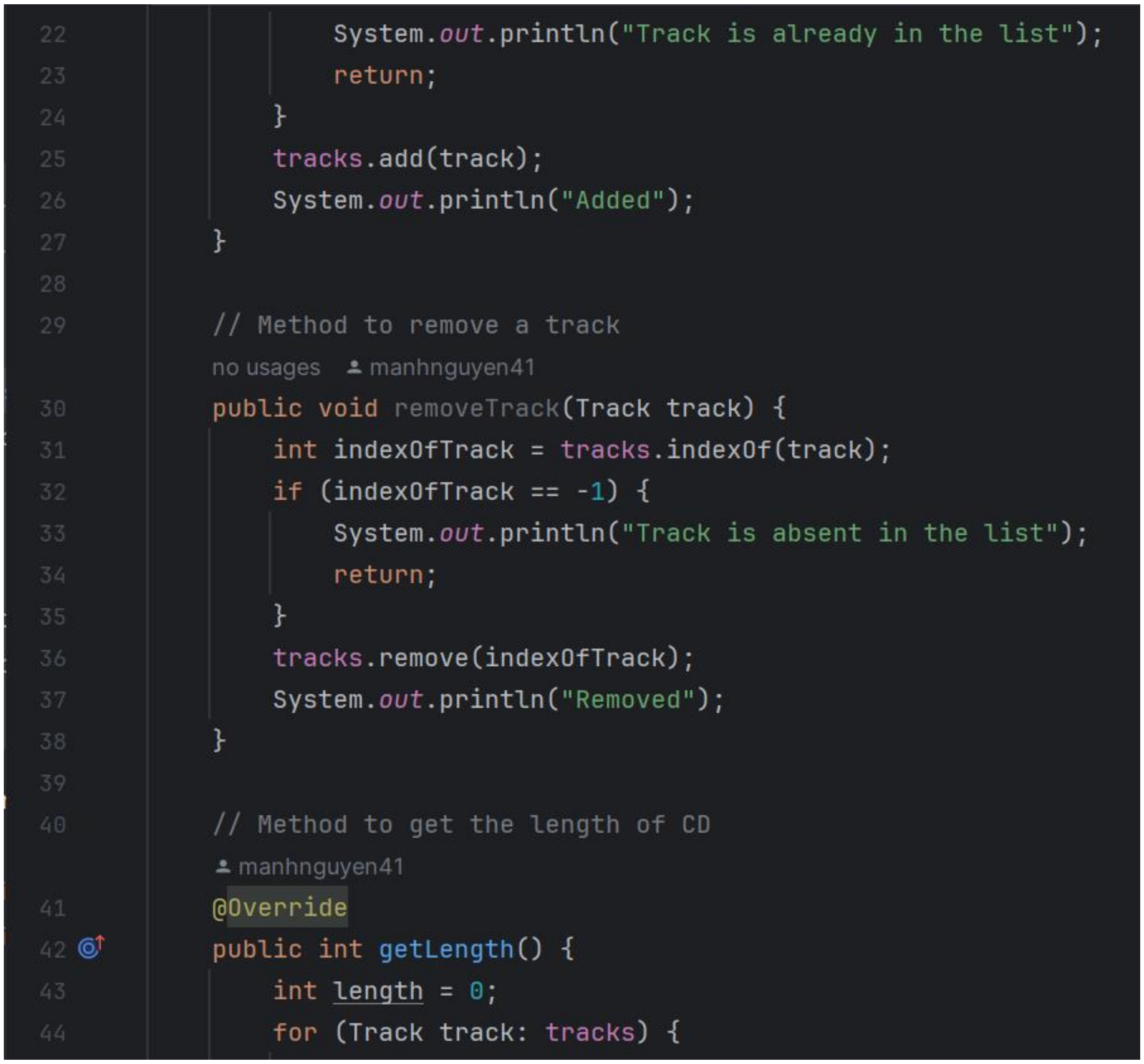
Figure 12 Track class code

6. Creating the CompactDisc class



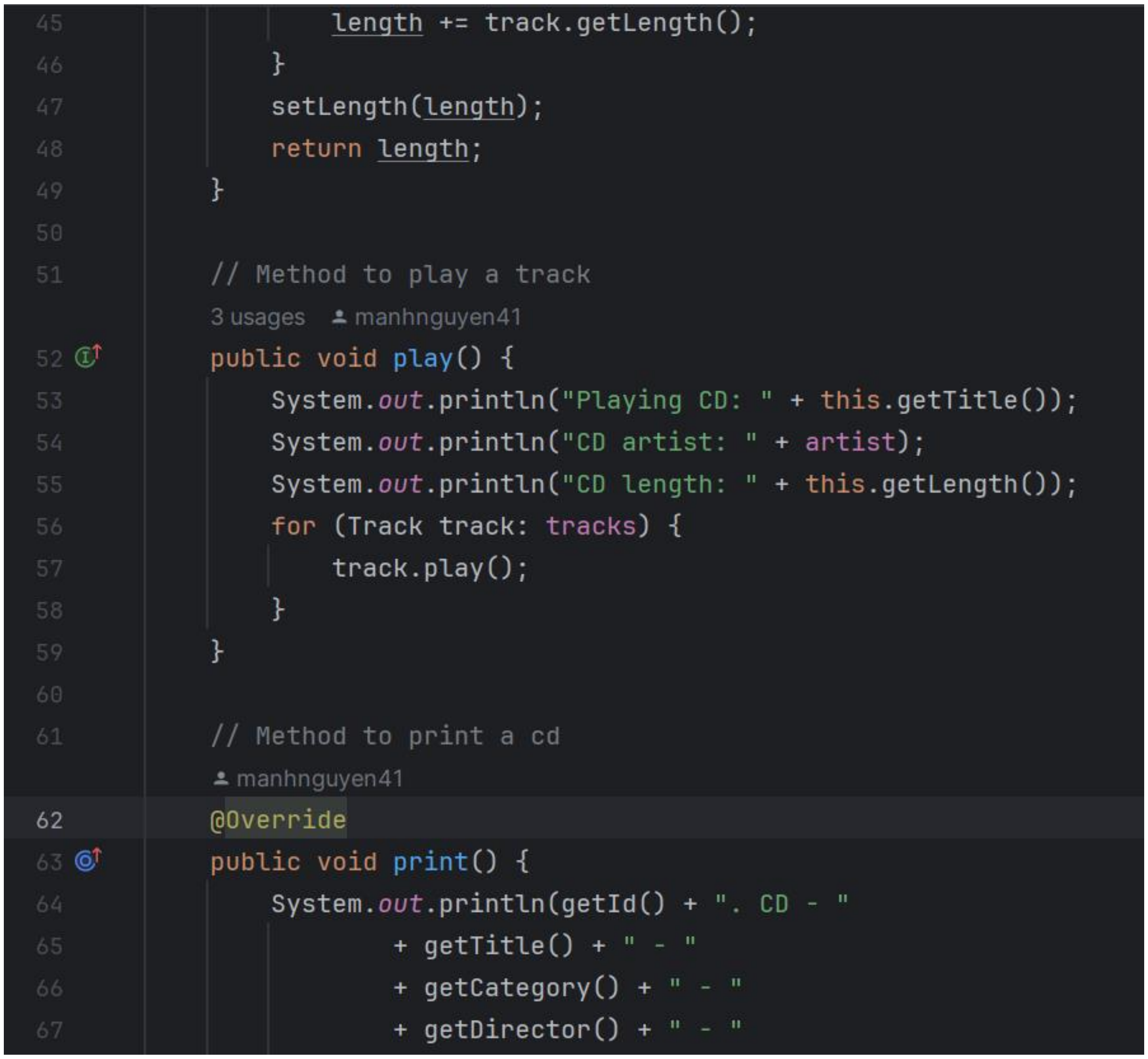
```
package lab04.AimsProject.Media;

import ...

10 usages  manhnguyen41
public class CompactDisc extends Disc implements Playable{
    // Attribute
    5 usages
    private String artist;
    6 usages
    private List<Track> tracks = new ArrayList<~>();

    // Constructor
    7 usages  manhnguyen41
    public CompactDisc(int id, String title, String category, float cost,
                       String director, int length, String artist) {
        super(id, title, category, cost, director, length);
        this.artist = artist;
    }

    // Method to add a track
    no usages  manhnguyen41
    public void addTrack(Track track) {
        int indexOfTrack = tracks.indexOf(track);
        if (indexOfTrack == -1) {
```

Figure 13 CompactDisc class code



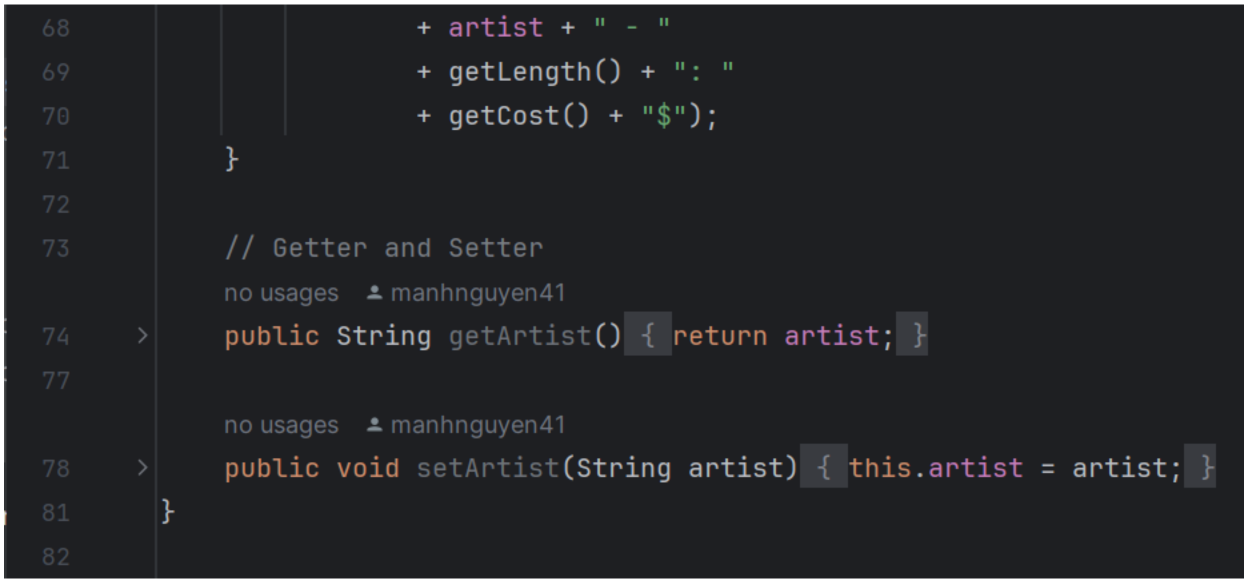
```
            System.out.println("Track is already in the list");
            return;
        }
        tracks.add(track);
        System.out.println("Added");
    }

    // Method to remove a track
    no usages   manhnguyen41
    public void removeTrack(Track track) {
        int indexOfTrack = tracks.indexOf(track);
        if (indexOfTrack == -1) {
            System.out.println("Track is absent in the list");
            return;
        }
        tracks.remove(indexOfTrack);
        System.out.println("Removed");
    }

    // Method to get the length of CD
    manhnguyen41
    @Override
    public int getLength() {
        int length = 0;
        for (Track track: tracks) {
```

Figure 14 CompactDisc class code



```java
                length += track.getLength();
            }
            setLength(length);
            return length;
        }

        // Method to play a track
        3 usages  manhnguyen41
        public void play() {
            System.out.println("Playing CD: " + this.getTitle());
            System.out.println("CD artist: " + artist);
            System.out.println("CD length: " + this.getLength());
            for (Track track: tracks) {
                track.play();
            }
        }

        // Method to print a cd
        manhnguyen41
        @Override
        public void print() {
            System.out.println(getId() + ". CD - "
                    + getTitle() + " - "
                    + getCategory() + " - "
                    + getDirector() + " - "
```

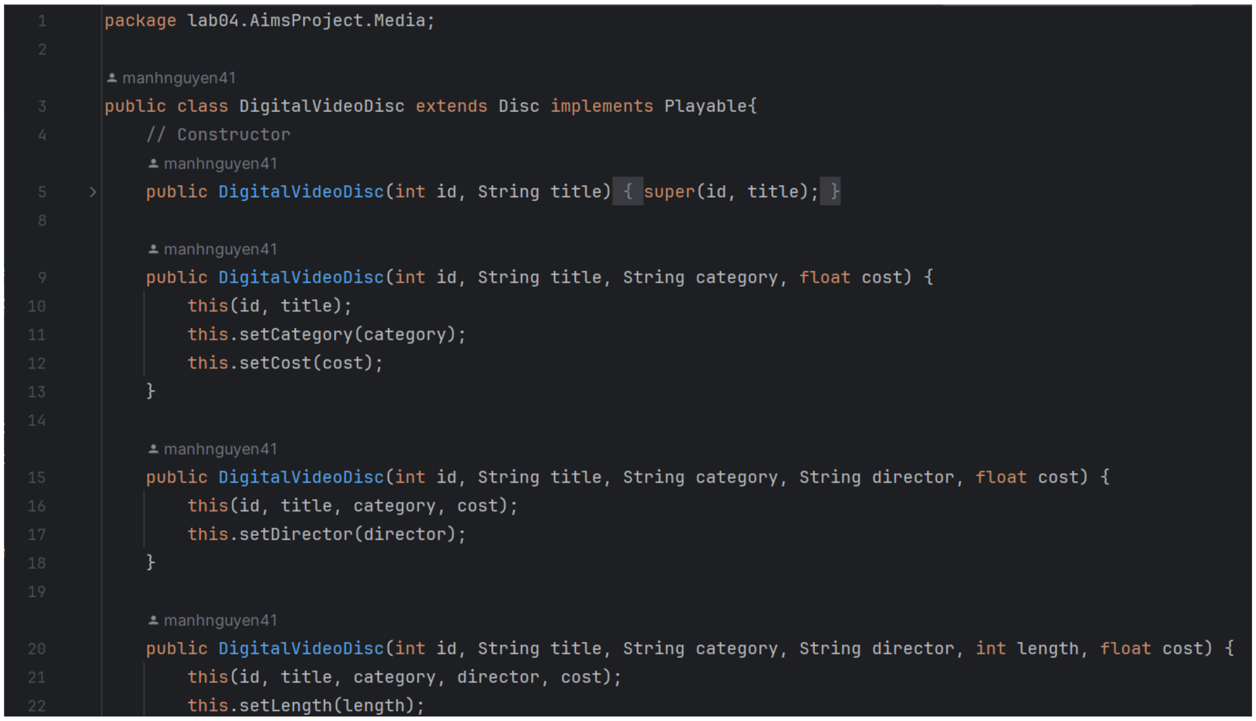
Figure 15 CompactDisc class code



```
                    + artist + " - "
                    + getLength() + ": "
                    + getCost() + "$");
    }

    // Getter and Setter
    no usages  manhnguyen41
    public String getArtist() { return artist; }

    no usages  manhnguyen41
    public void setArtist(String artist) { this.artist = artist; }
}
```

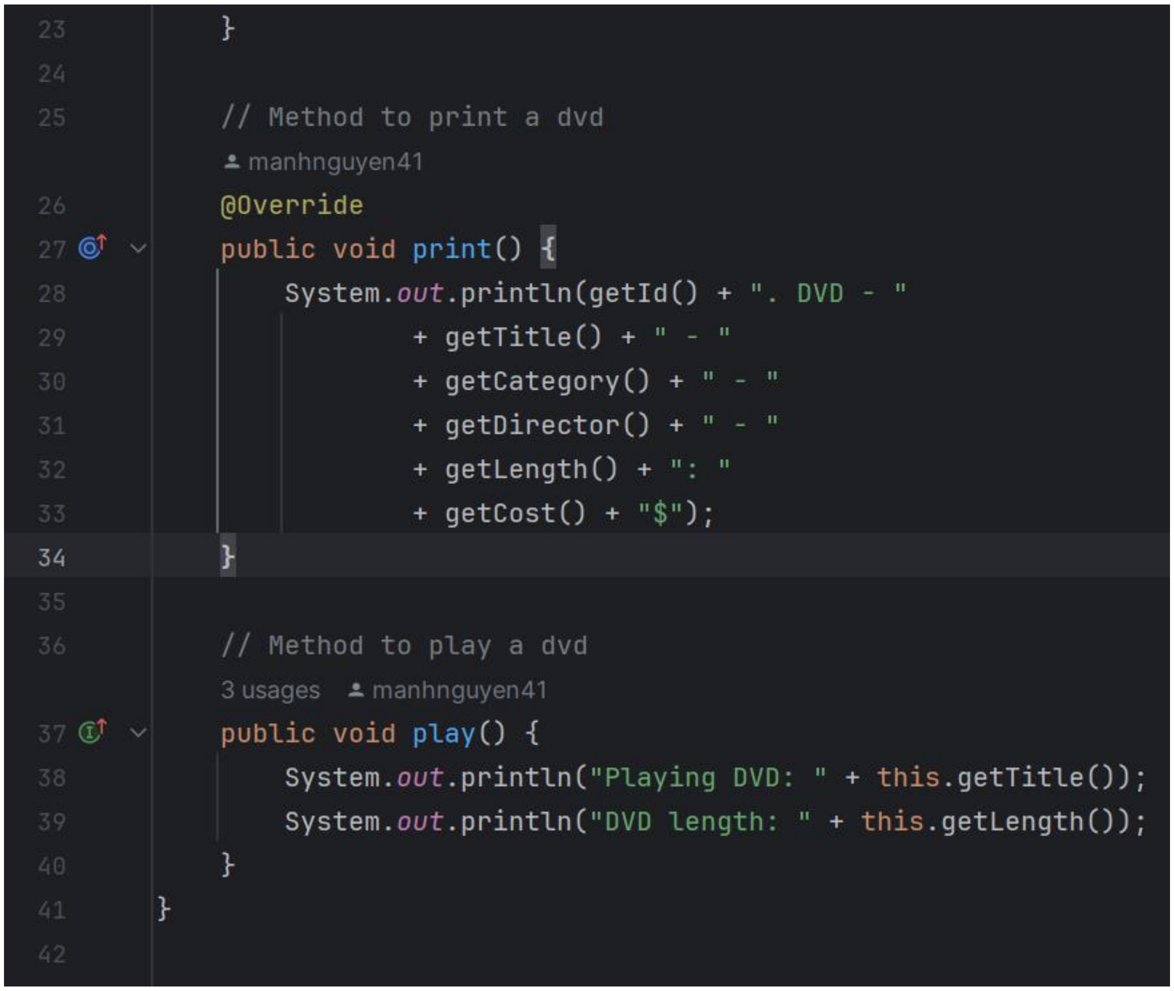
Figure 16 CompactDisc class code

7. Update the DigitalVideoDisc class



```
package lab04.AimsProject.Media;

manhnguyen41
public class DigitalVideoDisc extends Disc implements Playable{
    // Constructor
    manhnguyen41
    public DigitalVideoDisc(int id, String title) { super(id, title); }

    manhnguyen41
    public DigitalVideoDisc(int id, String title, String category, float cost) {
        this(id, title);
        this.setCategory(category);
        this.setCost(cost);
    }

    manhnguyen41
    public DigitalVideoDisc(int id, String title, String category, String director, float cost) {
        this(id, title, category, cost);
        this.setDirector(director);
    }

    manhnguyen41
    public DigitalVideoDisc(int id, String title, String category, String director, int length, float cost) {
        this(id, title, category, director, cost);
        this.setLength(length);
```

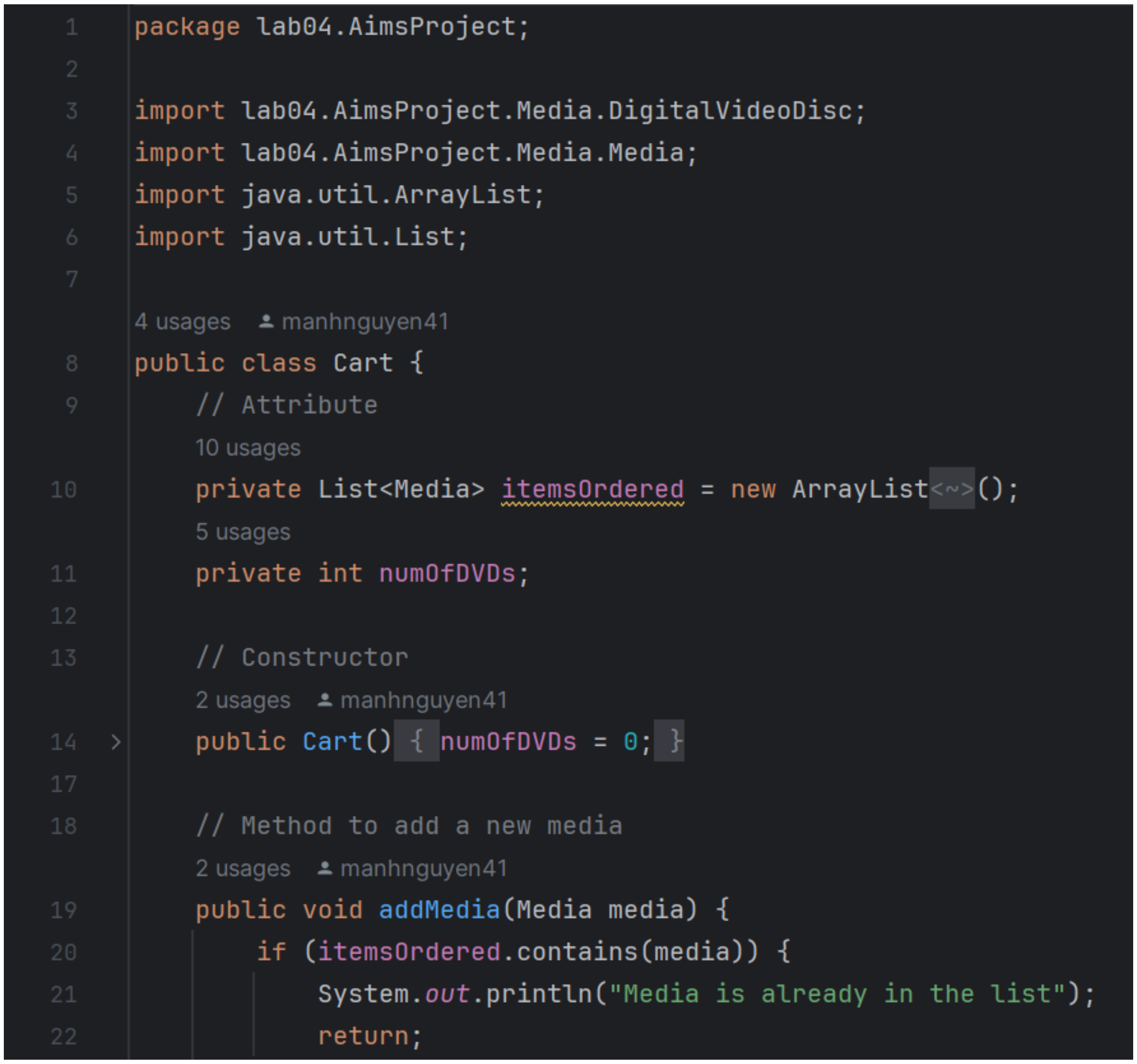
Figure 17 DVD class code



```
    }

    // Method to print a dvd
    manhnguyen41
    @Override
    public void print() {
        System.out.println(getId() + ". DVD - "
                + getTitle() + " - "
                + getCategory() + " - "
                + getDirector() + " - "
                + getLength() + ": "
                + getCost() + "$");
    }

    // Method to play a dvd
    3 usages  manhnguyen41
    public void play() {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }
}
```

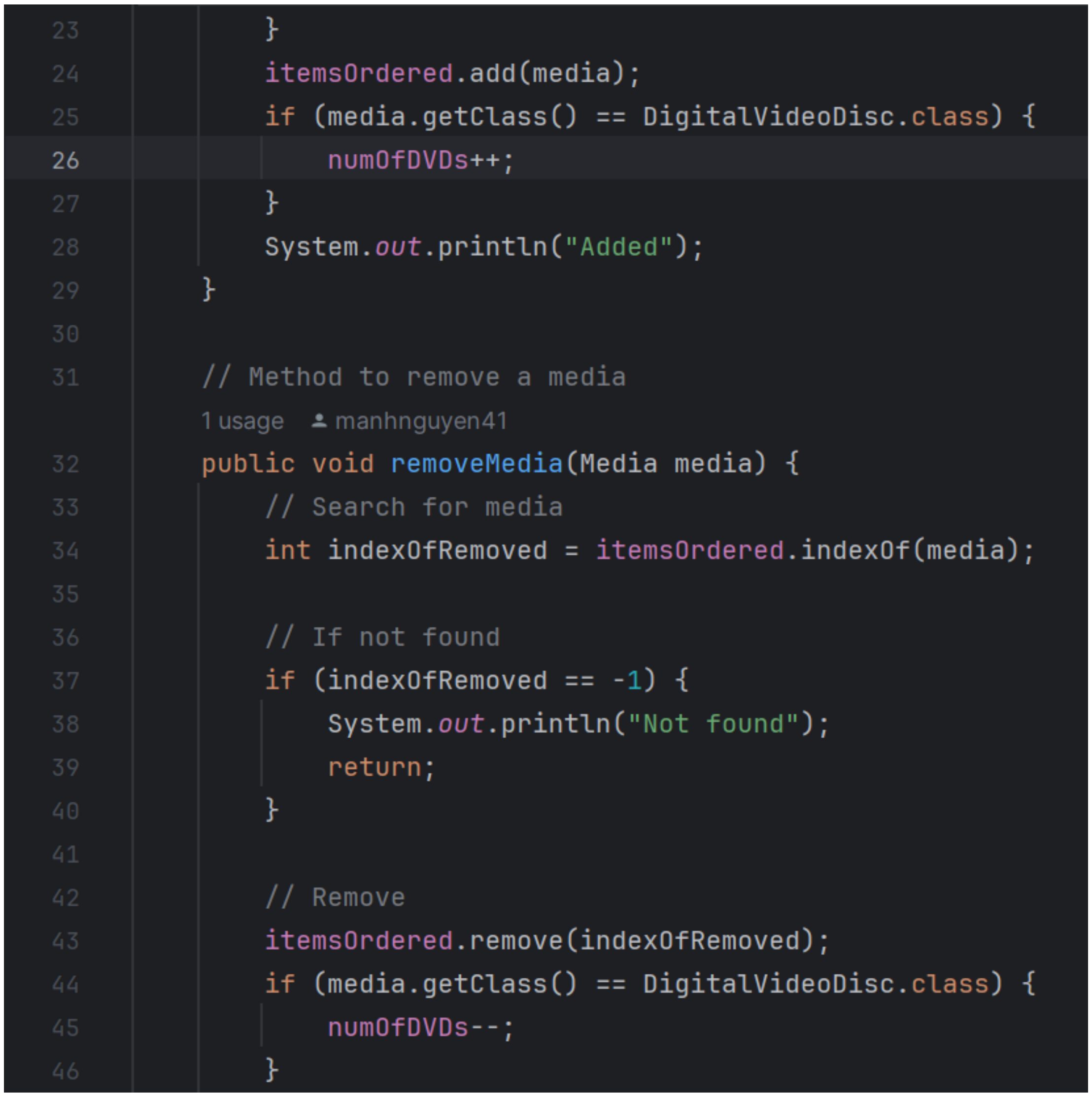
Figure 18 DVD class code

8. Update the Cart class



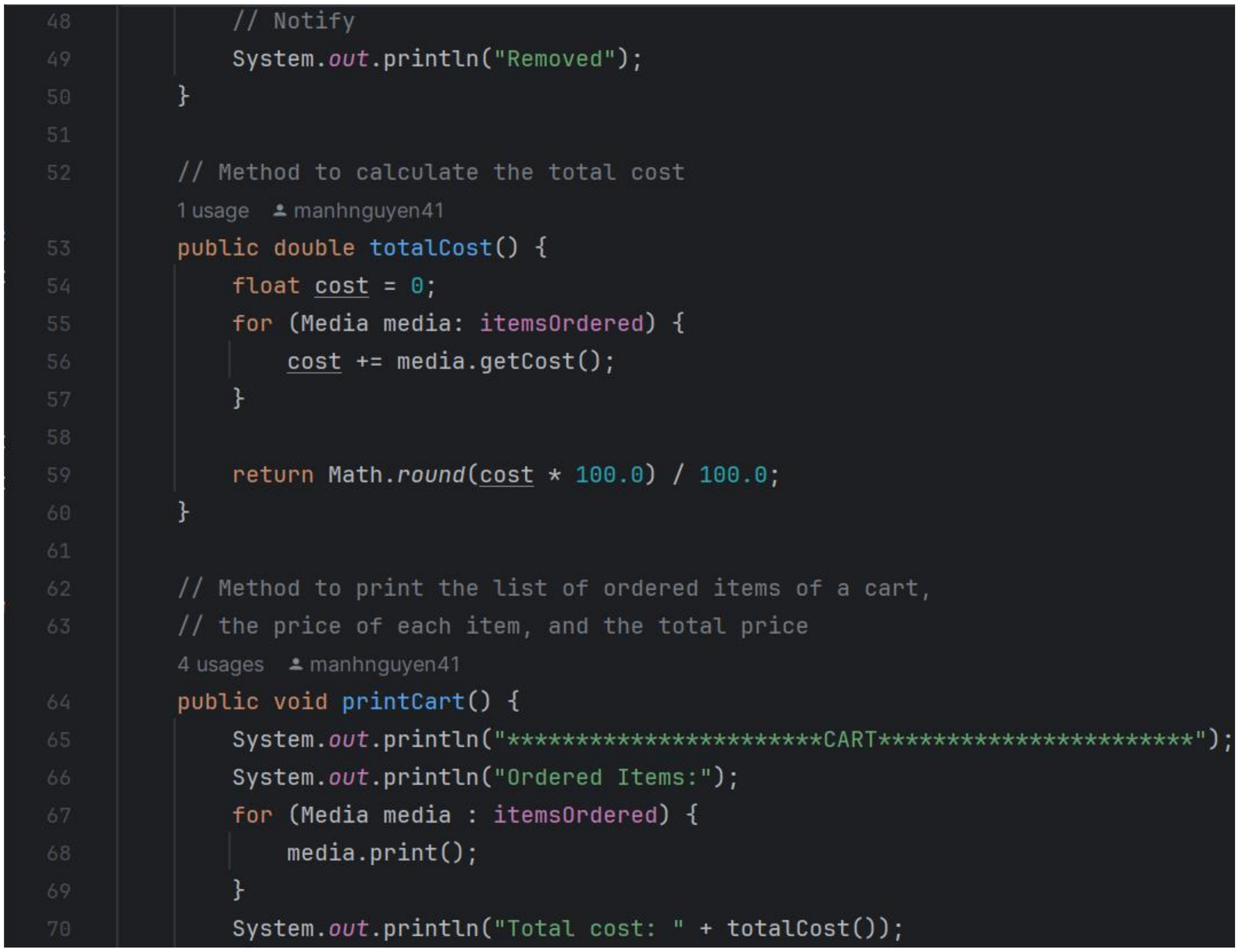
```java
package lab04.AimsProject;

import lab04.AimsProject.Media.DigitalVideoDisc;
import lab04.AimsProject.Media.Media;
import java.util.ArrayList;
import java.util.List;

4 usages   manhnguyen41
public class Cart {
    // Attribute
    10 usages
    private List<Media> itemsOrdered = new ArrayList<~>();
    5 usages
    private int numOfDVDs;

    // Constructor
    2 usages   manhnguyen41
    public Cart() { numOfDVDs = 0; }

    // Method to add a new media
    2 usages   manhnguyen41
    public void addMedia(Media media) {
        if (itemsOrdered.contains(media)) {
            System.out.println("Media is already in the list");
            return;
```

Figure 19 Cart class code



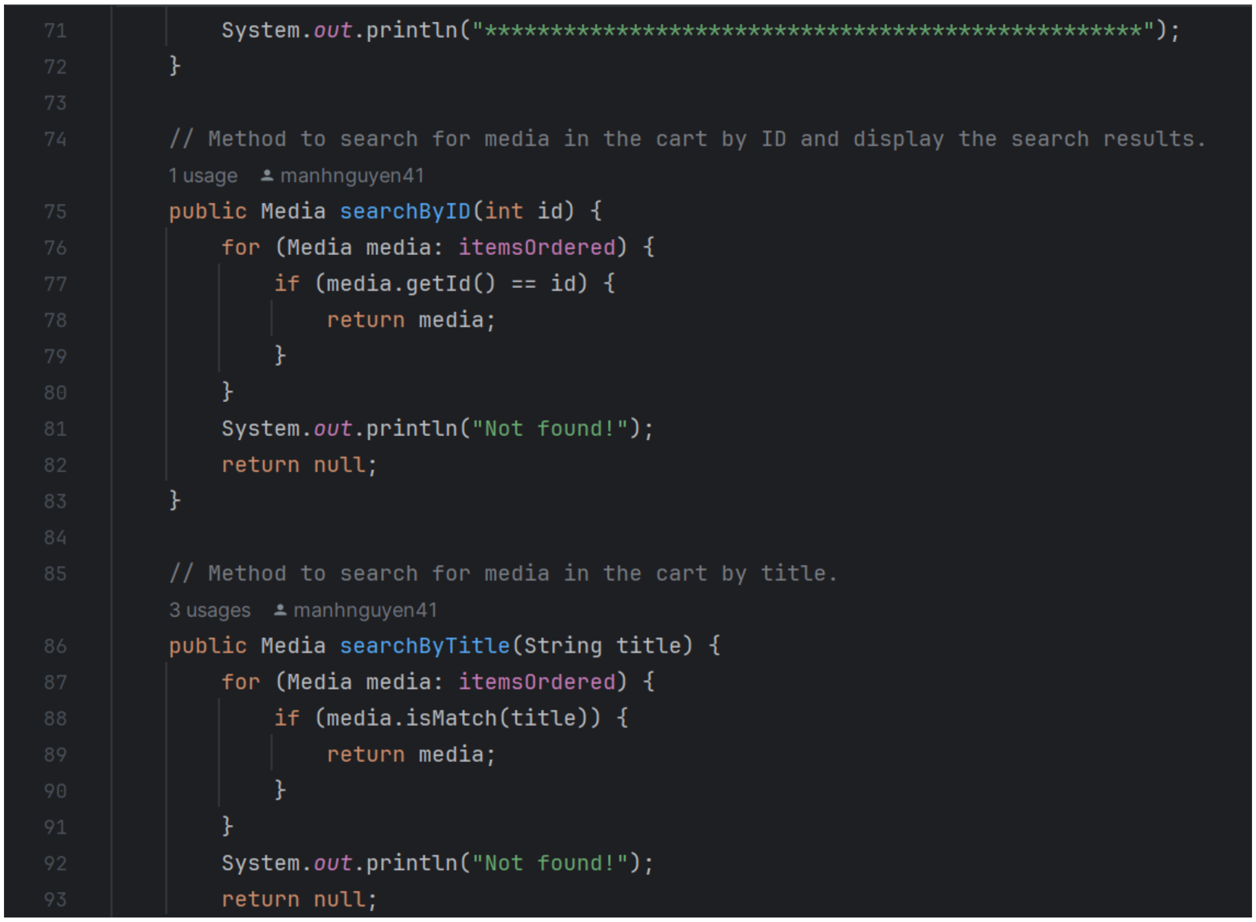
```
        }
        itemsOrdered.add(media);
        if (media.getClass() == DigitalVideoDisc.class) {
            numOfDVDs++;
        }
        System.out.println("Added");
    }

    // Method to remove a media
    1 usage   manhnguyen41
    public void removeMedia(Media media) {
        // Search for media
        int indexOfRemoved = itemsOrdered.indexOf(media);

        // If not found
        if (indexOfRemoved == -1) {
            System.out.println("Not found");
            return;
        }

        // Remove
        itemsOrdered.remove(indexOfRemoved);
        if (media.getClass() == DigitalVideoDisc.class) {
            numOfDVDs--;
        }
```

Figure 20 Cart class code



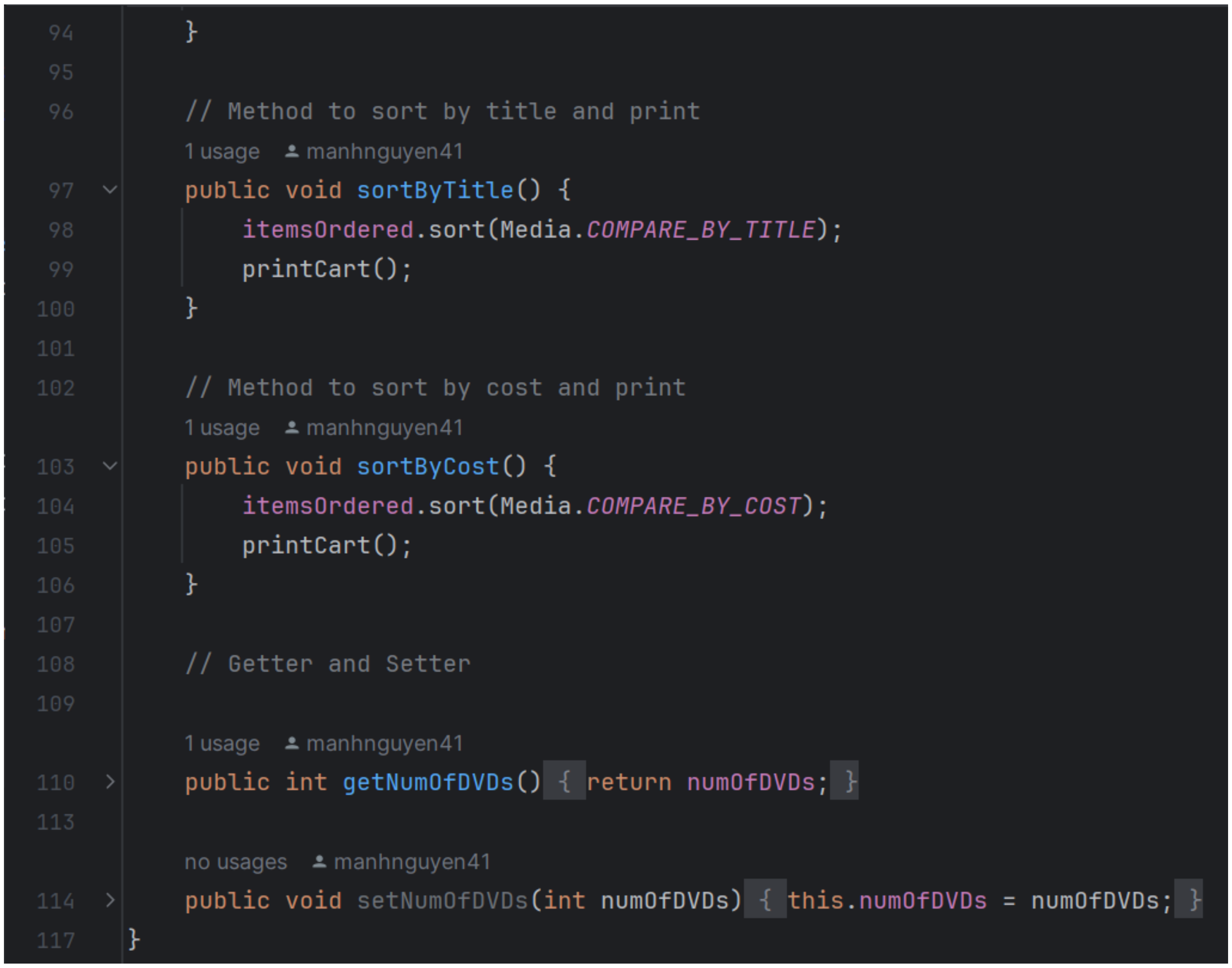
```
        // Notify
        System.out.println("Removed");
    }

    // Method to calculate the total cost
    1 usage  manhnguyen41
    public double totalCost() {
        float cost = 0;
        for (Media media: itemsOrdered) {
            cost += media.getCost();
        }

        return Math.round(cost * 100.0) / 100.0;
    }

    // Method to print the list of ordered items of a cart,
    // the price of each item, and the total price
    4 usages  manhnguyen41
    public void printCart() {
        System.out.println("***********************CART***********************");
        System.out.println("Ordered Items:");
        for (Media media : itemsOrdered) {
            media.print();
        }
        System.out.println("Total cost: " + totalCost());
```

Figure 21 Cart class code



```java
        System.out.println("***************************************************");
    }

    // Method to search for media in the cart by ID and display the search results.
    1 usage  manhnguyen41
    public Media searchByID(int id) {
        for (Media media: itemsOrdered) {
            if (media.getId() == id) {
                return media;
            }
        }
        System.out.println("Not found!");
        return null;
    }

    // Method to search for media in the cart by title.
    3 usages  manhnguyen41
    public Media searchByTitle(String title) {
        for (Media media: itemsOrdered) {
            if (media.isMatch(title)) {
                return media;
            }
        }
        System.out.println("Not found!");
        return null;
```

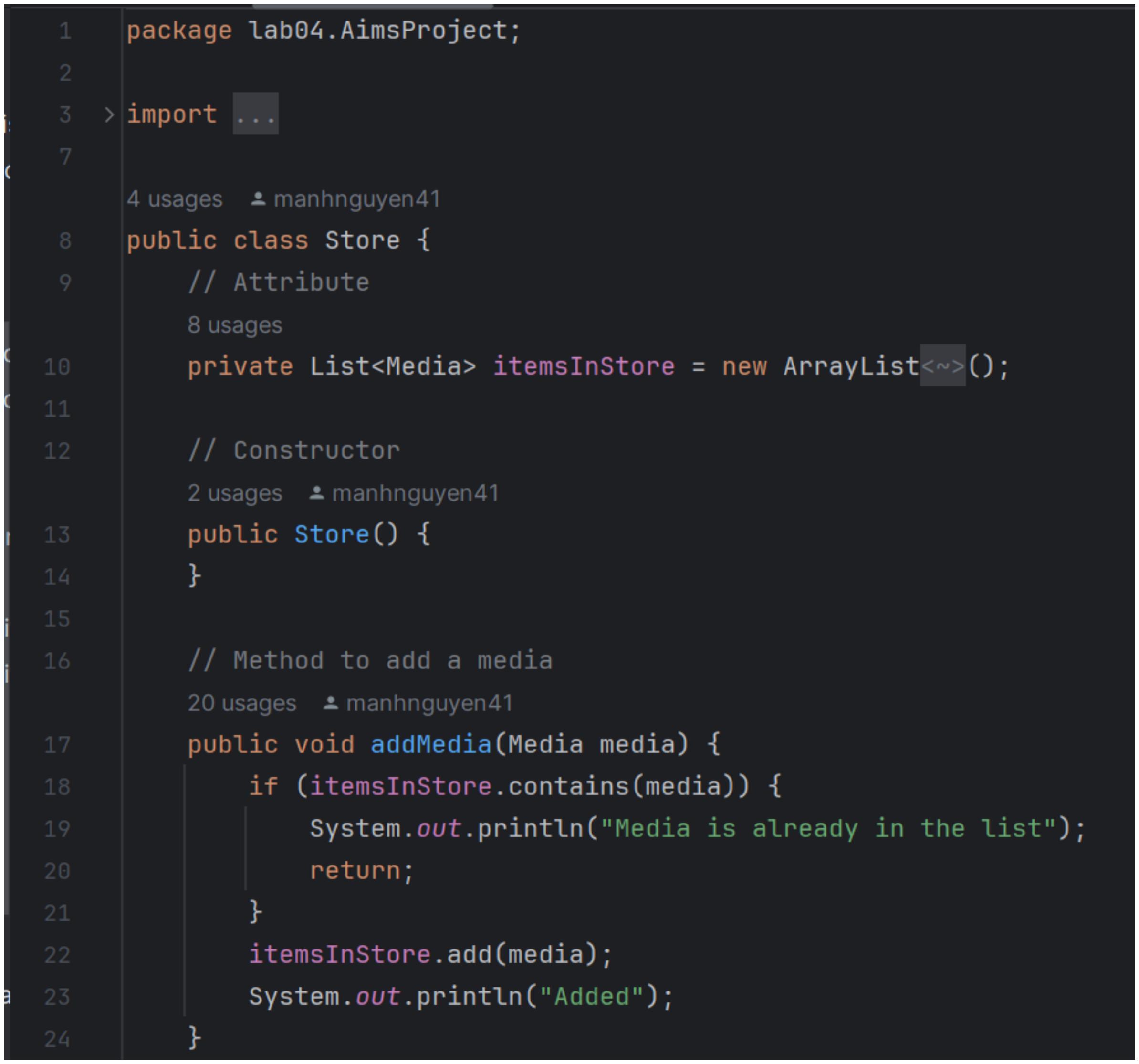
Figure 22 Cart class code



```
    }

    // Method to sort by title and print
    1 usage  manhnguyen41
    public void sortByTitle() {
        itemsOrdered.sort(Media.COMPARE_BY_TITLE);
        printCart();
    }

    // Method to sort by cost and print
    1 usage  manhnguyen41
    public void sortByCost() {
        itemsOrdered.sort(Media.COMPARE_BY_COST);
        printCart();
    }

    // Getter and Setter

    1 usage  manhnguyen41
    public int getNumOfDVDs() { return numOfDVDs; }

    no usages  manhnguyen41
    public void setNumOfDVDs(int numOfDVDs) { this.numOfDVDs = numOfDVDs; }
}
```

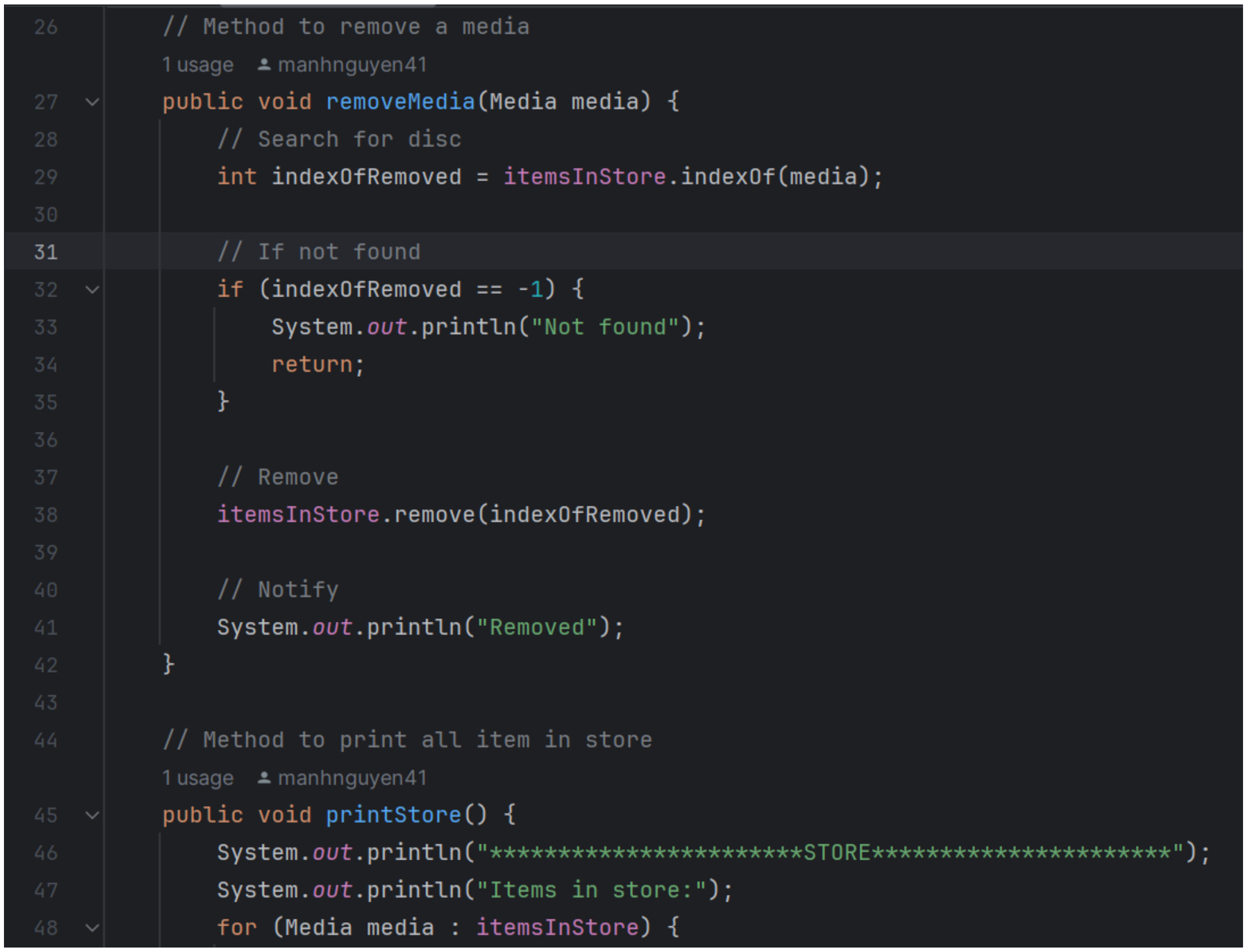
Figure 23 Cart class code

9. Update the Store class



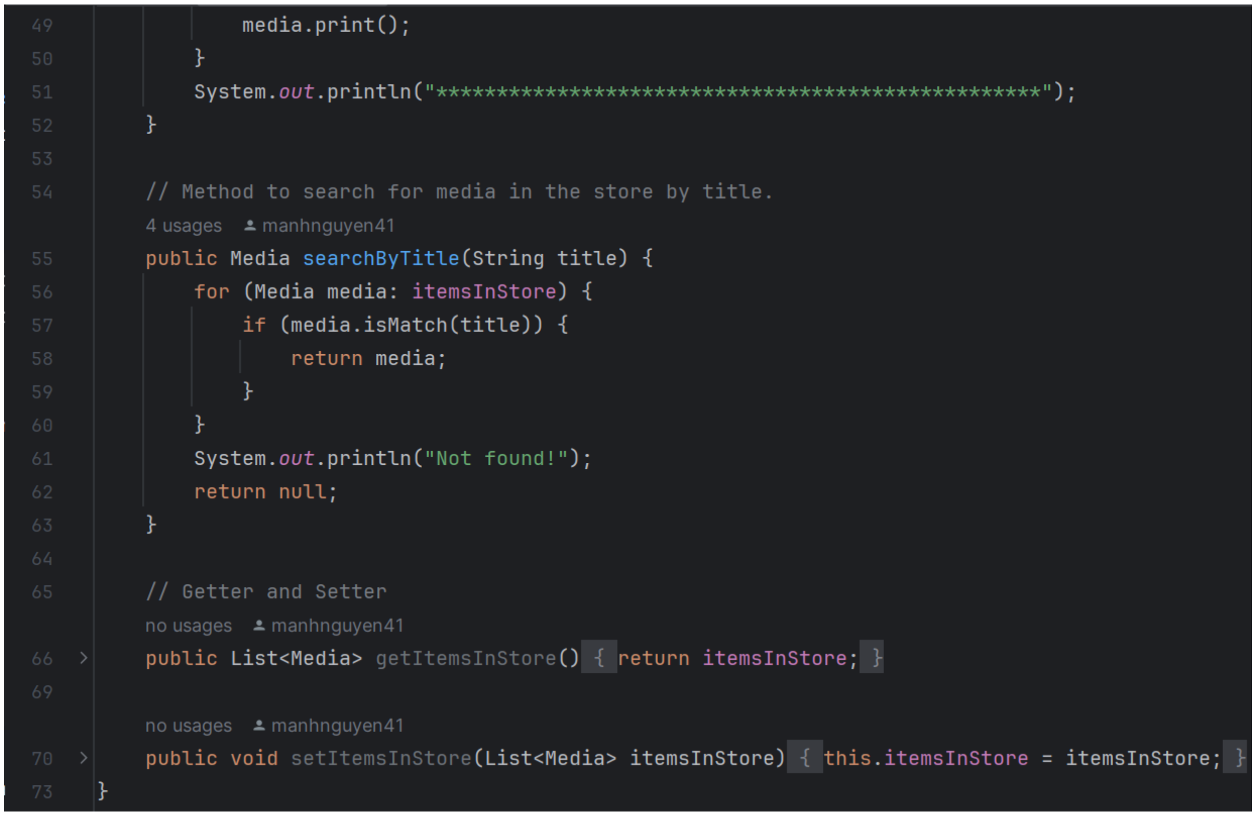
```
package lab04.AimsProject;

import ...

4 usages  manhnguyen41
public class Store {
    // Attribute
    8 usages
    private List<Media> itemsInStore = new ArrayList<~>();

    // Constructor
    2 usages  manhnguyen41
    public Store() {
    }

    // Method to add a media
    20 usages  manhnguyen41
    public void addMedia(Media media) {
        if (itemsInStore.contains(media)) {
            System.out.println("Media is already in the list");
            return;
        }
        itemsInStore.add(media);
        System.out.println("Added");
    }
```

Figure 24 Store class code



```java
    // Method to remove a media
    1 usage  manhnguyen41
    public void removeMedia(Media media) {
        // Search for disc
        int indexOfRemoved = itemsInStore.indexOf(media);

        // If not found
        if (indexOfRemoved == -1) {
            System.out.println("Not found");
            return;
        }

        // Remove
        itemsInStore.remove(indexOfRemoved);

        // Notify
        System.out.println("Removed");
    }

    // Method to print all item in store
    1 usage  manhnguyen41
    public void printStore() {
        System.out.println("**********************STORE**********************");
        System.out.println("Items in store:");
        for (Media media : itemsInStore) {
```

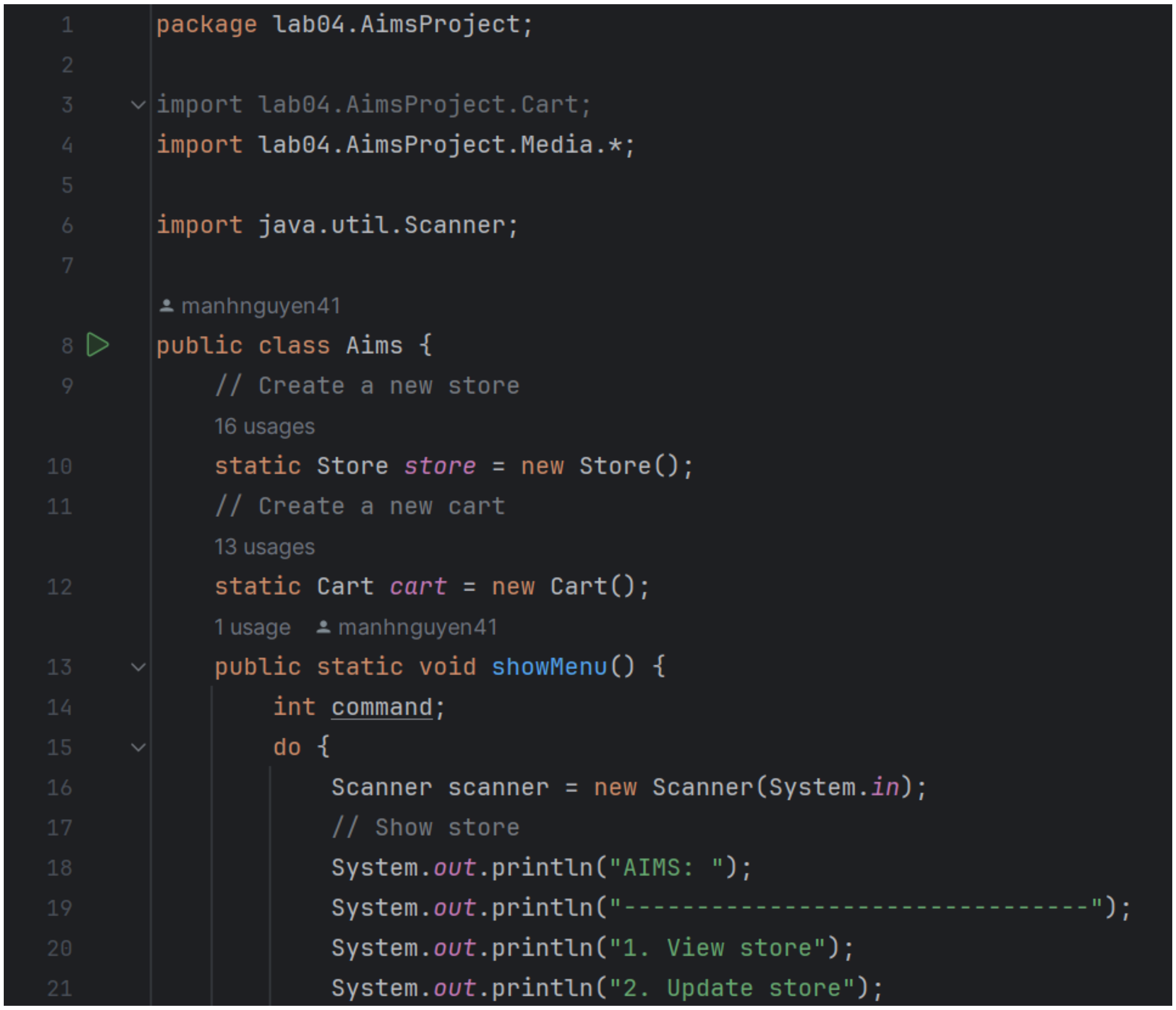
Figure 25 Store class code



```
                media.print();
            }
            System.out.println("*****************************************************");
        }

        // Method to search for media in the store by title.
        4 usages  manhnguyen41
        public Media searchByTitle(String title) {
            for (Media media: itemsInStore) {
                if (media.isMatch(title)) {
                    return media;
                }
            }
            System.out.println("Not found!");
            return null;
        }

        // Getter and Setter
        no usages  manhnguyen41
        public List<Media> getItemsInStore() { return itemsInStore; }

        no usages  manhnguyen41
        public void setItemsInStore(List<Media> itemsInStore) { this.itemsInStore = itemsInStore; }
}
```

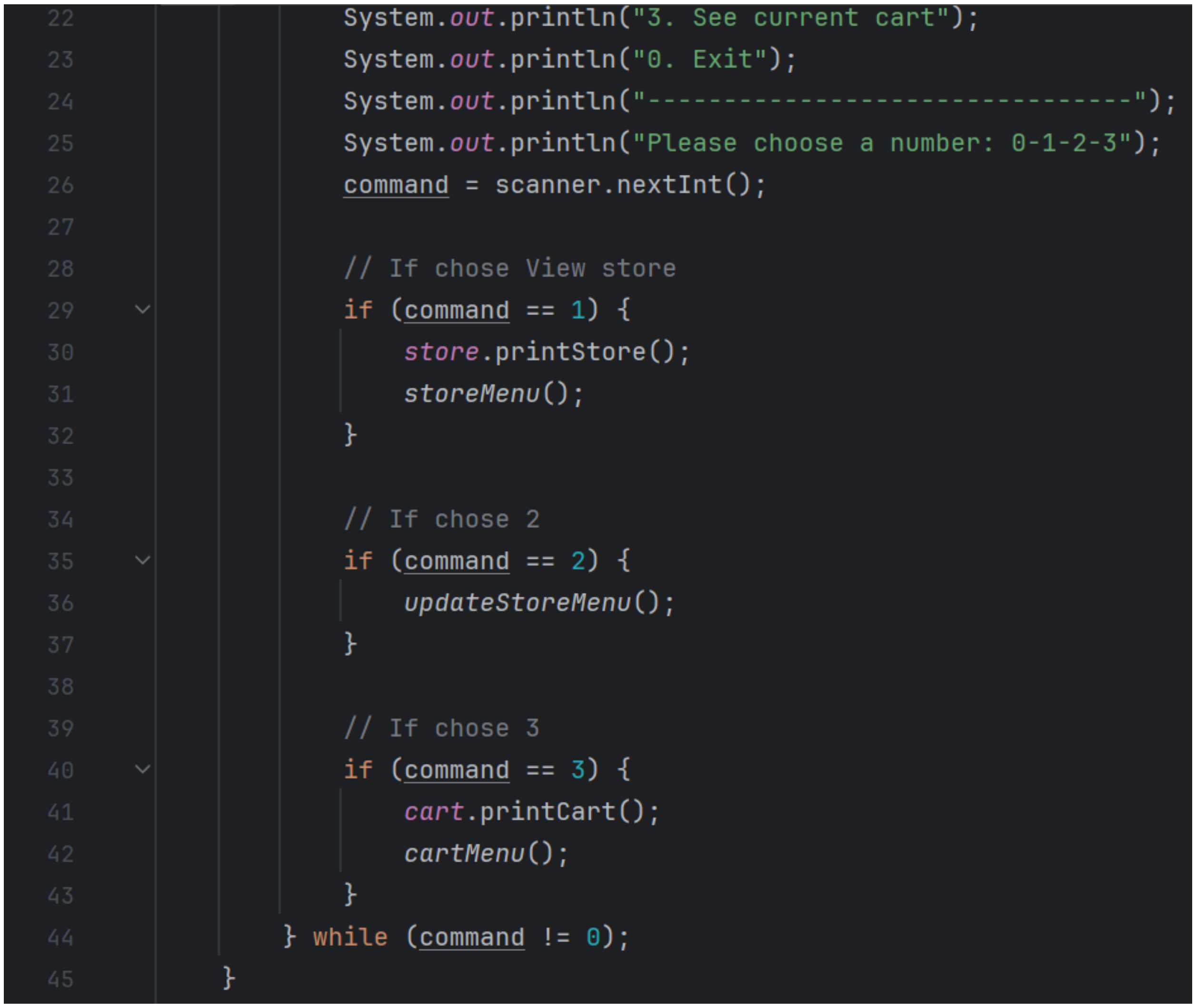
Figure 26 Store class code

10. Update the Aims class



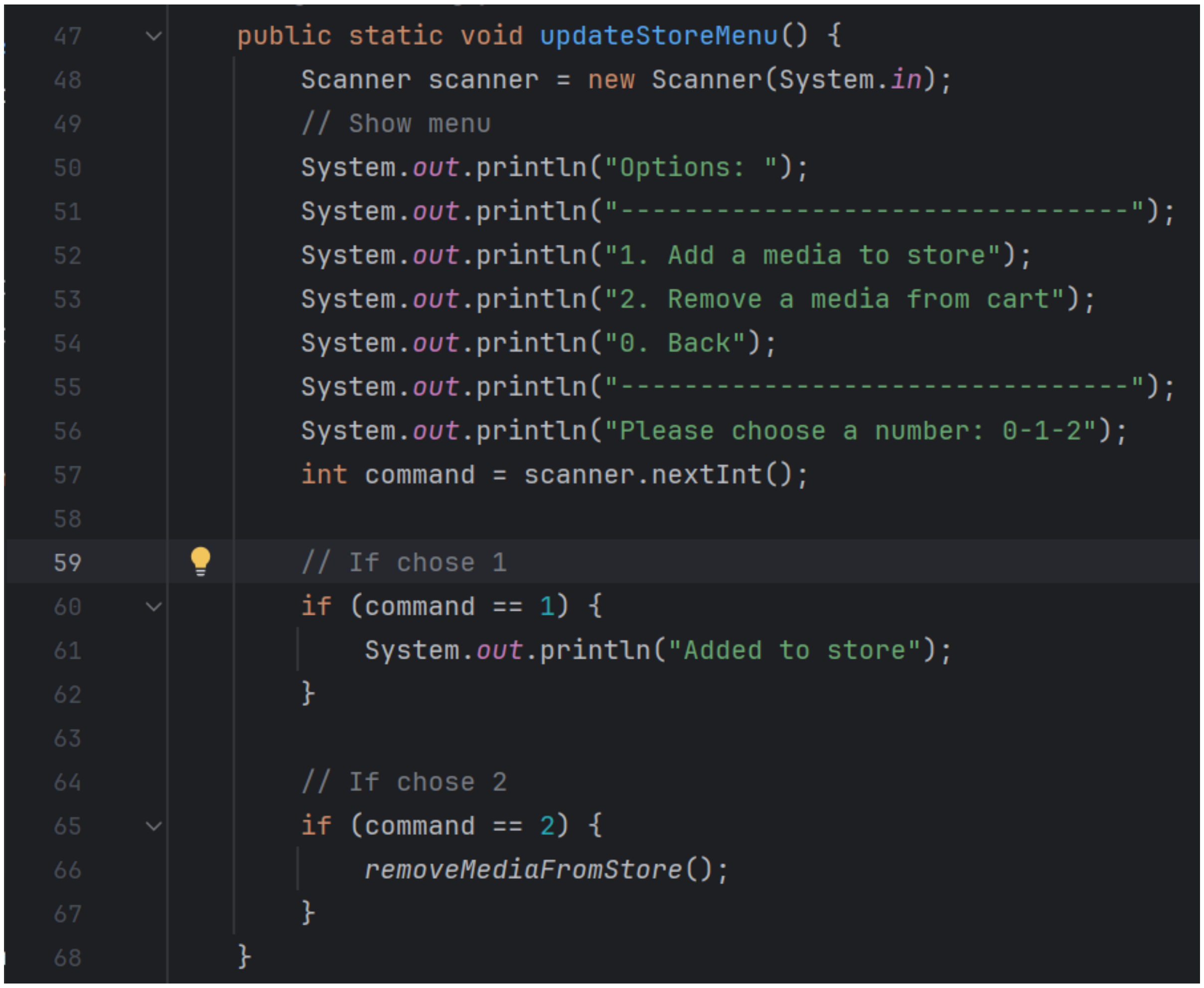
```java
package lab04.AimsProject;

import lab04.AimsProject.Cart;
import lab04.AimsProject.Media.*;

import java.util.Scanner;

manhnguyen41
public class Aims {
    // Create a new store
    16 usages
    static Store store = new Store();
    // Create a new cart
    13 usages
    static Cart cart = new Cart();
    1 usage  manhnguyen41
    public static void showMenu() {
        int command;
        do {
            Scanner scanner = new Scanner(System.in);
            // Show store
            System.out.println("AIMS: ");
            System.out.println("--------------------------------");
            System.out.println("1. View store");
            System.out.println("2. Update store");
```

Figure 27 Aims class code



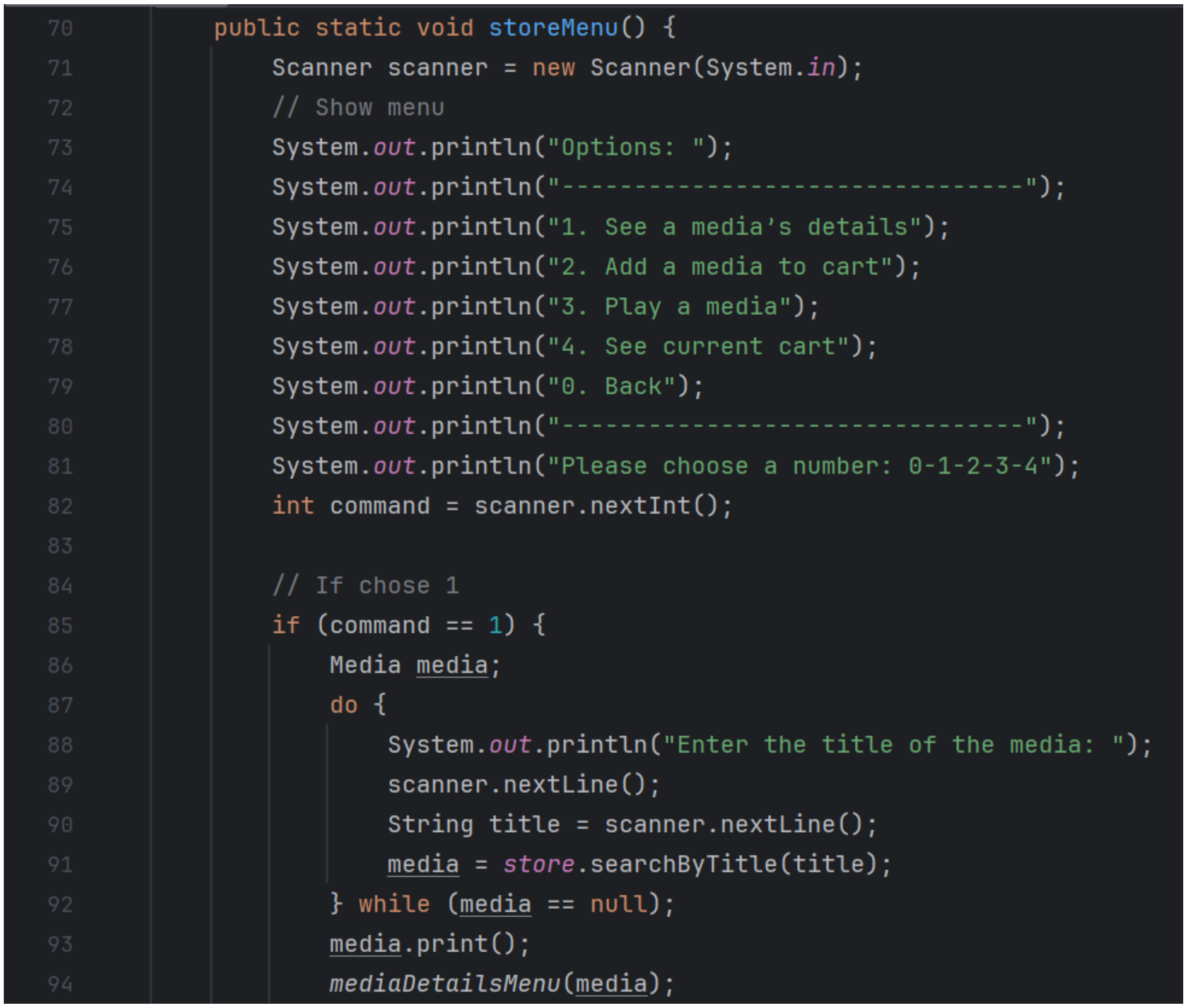
```
                System.out.println("3. See current cart");
                System.out.println("0. Exit");
                System.out.println("--------------------------------");
                System.out.println("Please choose a number: 0-1-2-3");
                command = scanner.nextInt();

                // If chose View store
                if (command == 1) {
                    store.printStore();
                    storeMenu();
                }

                // If chose 2
                if (command == 2) {
                    updateStoreMenu();
                }

                // If chose 3
                if (command == 3) {
                    cart.printCart();
                    cartMenu();
                }
            } while (command != 0);
        }
```

Figure 28 Aims class code



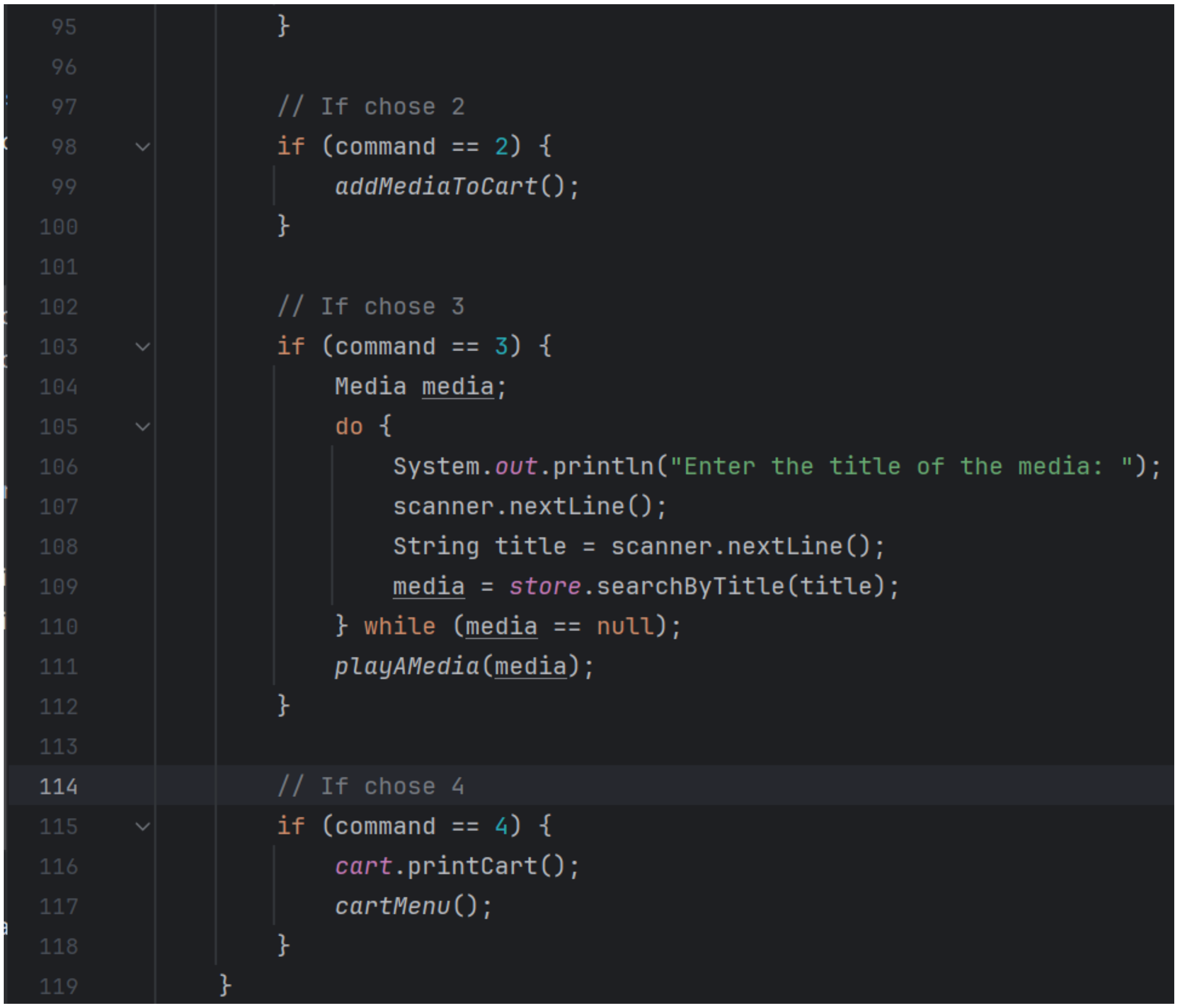
```java
public static void updateStoreMenu() {
    Scanner scanner = new Scanner(System.in);
    // Show menu
    System.out.println("Options: ");
    System.out.println("--------------------------------");
    System.out.println("1. Add a media to store");
    System.out.println("2. Remove a media from cart");
    System.out.println("0. Back");
    System.out.println("--------------------------------");
    System.out.println("Please choose a number: 0-1-2");
    int command = scanner.nextInt();

    // If chose 1
    if (command == 1) {
        System.out.println("Added to store");
    }

    // If chose 2
    if (command == 2) {
        removeMediaFromStore();
    }
}
```

Figure 29 Aims class code



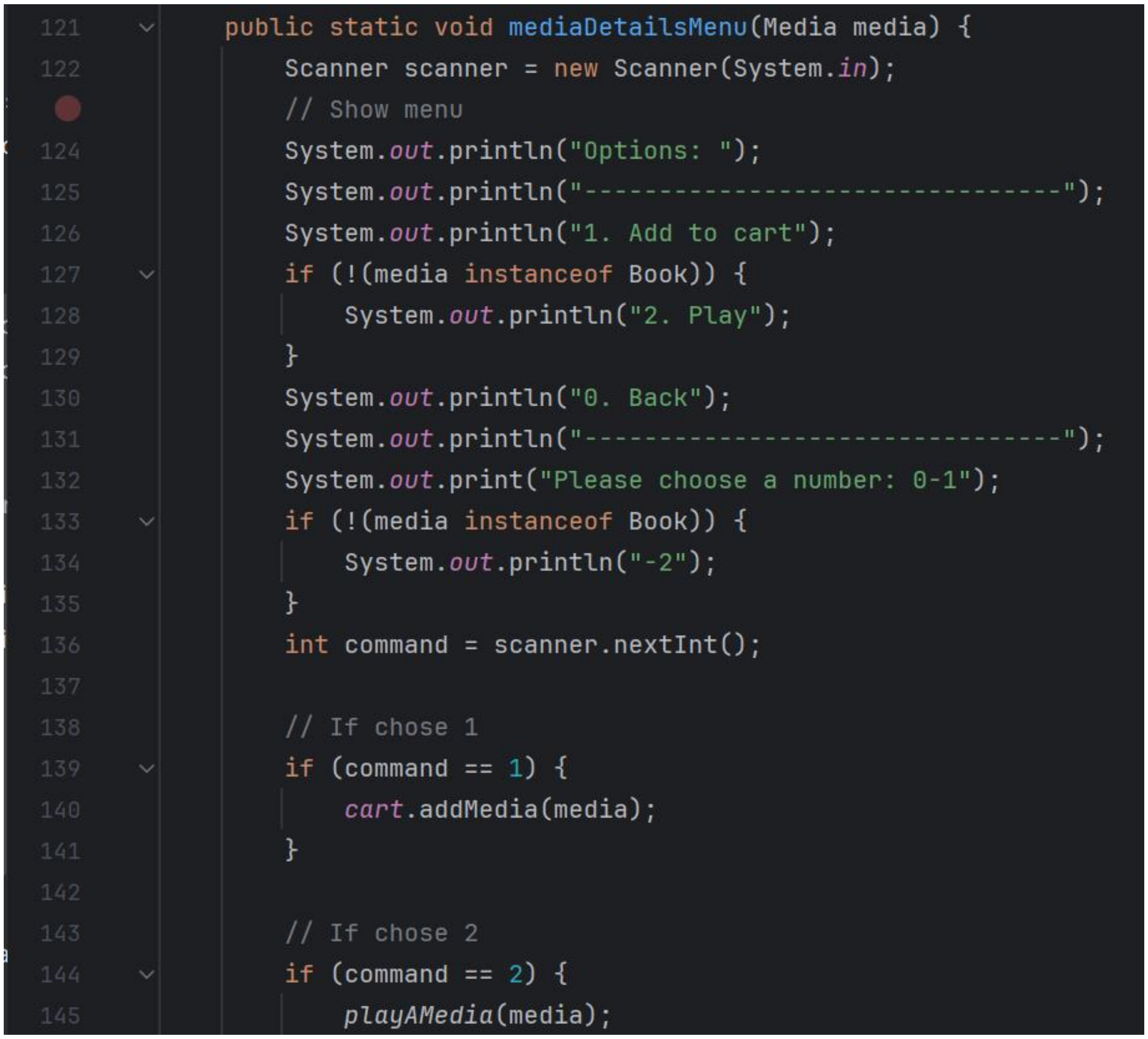
```java
    public static void storeMenu() {
        Scanner scanner = new Scanner(System.in);
        // Show menu
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. See a media's details");
        System.out.println("2. 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 command = scanner.nextInt();

        // If chose 1
        if (command == 1) {
            Media media;
            do {
                System.out.println("Enter the title of the media: ");
                scanner.nextLine();
                String title = scanner.nextLine();
                media = store.searchByTitle(title);
            } while (media == null);
            media.print();
            mediaDetailsMenu(media);
```

Figure 30 Aims class code



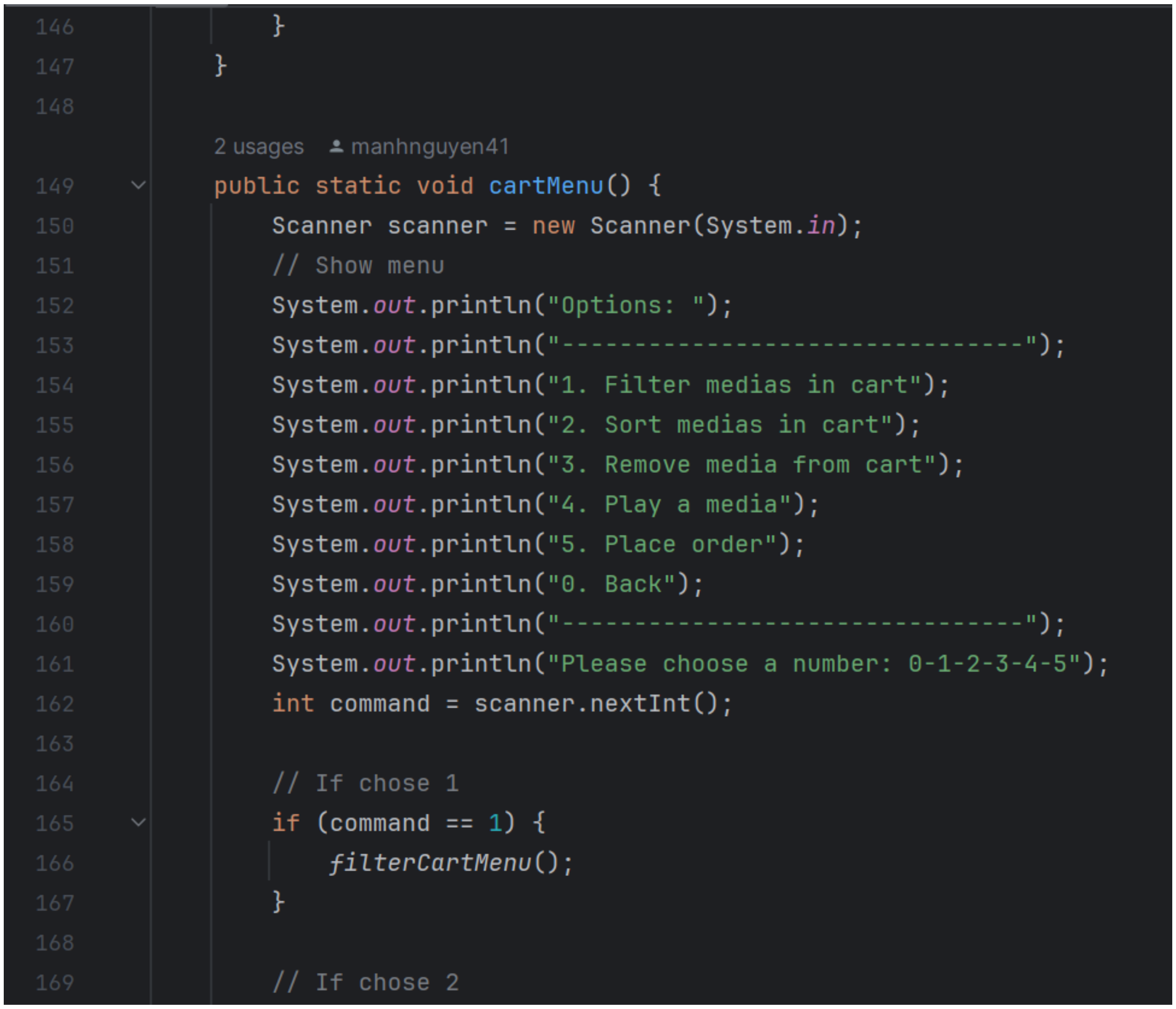
```
        }

        // If chose 2
        if (command == 2) {
            addMediaToCart();
        }

        // If chose 3
        if (command == 3) {
            Media media;
            do {
                System.out.println("Enter the title of the media: ");
                scanner.nextLine();
                String title = scanner.nextLine();
                media = store.searchByTitle(title);
            } while (media == null);
            playAMedia(media);
        }

        // If chose 4
        if (command == 4) {
            cart.printCart();
            cartMenu();
        }
    }
```

Figure 31 Aims class code



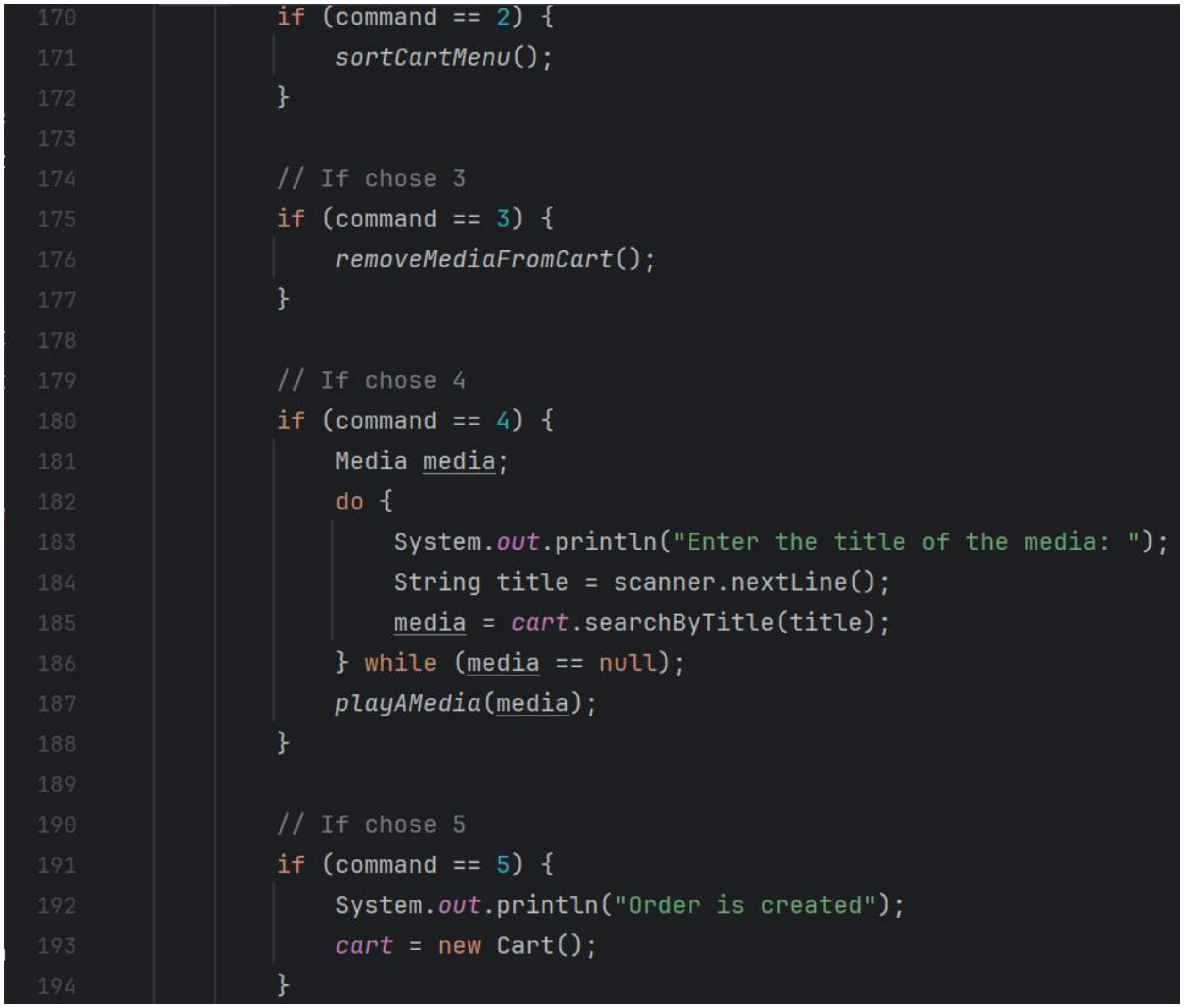
```java
    public static void mediaDetailsMenu(Media media) {
        Scanner scanner = new Scanner(System.in);
        // Show menu
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. Add to cart");
        if (!(media instanceof Book)) {
            System.out.println("2. Play");
        }
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.print("Please choose a number: 0-1");
        if (!(media instanceof Book)) {
            System.out.println("-2");
        }
        int command = scanner.nextInt();

        // If chose 1
        if (command == 1) {
            cart.addMedia(media);
        }

        // If chose 2
        if (command == 2) {
            playAMedia(media);
```

Figure 32 Aims class code



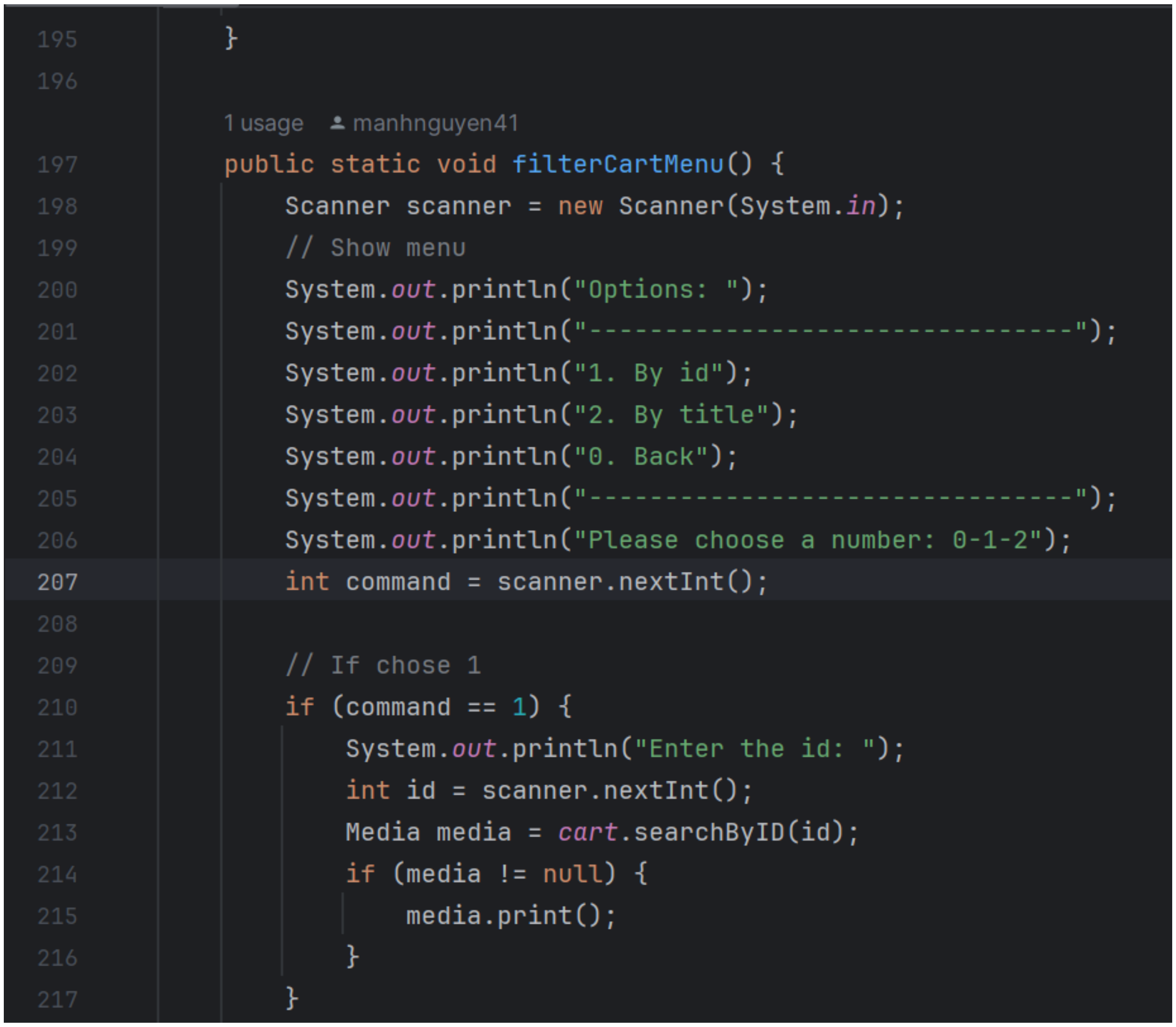
```
        }
    }

    2 usages  manhnguyen41
    public static void cartMenu() {
        Scanner scanner = new Scanner(System.in);
        // Show menu
        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");
        int command = scanner.nextInt();

        // If chose 1
        if (command == 1) {
            filterCartMenu();
        }

        // If chose 2
```

Figure 33 Aims class code



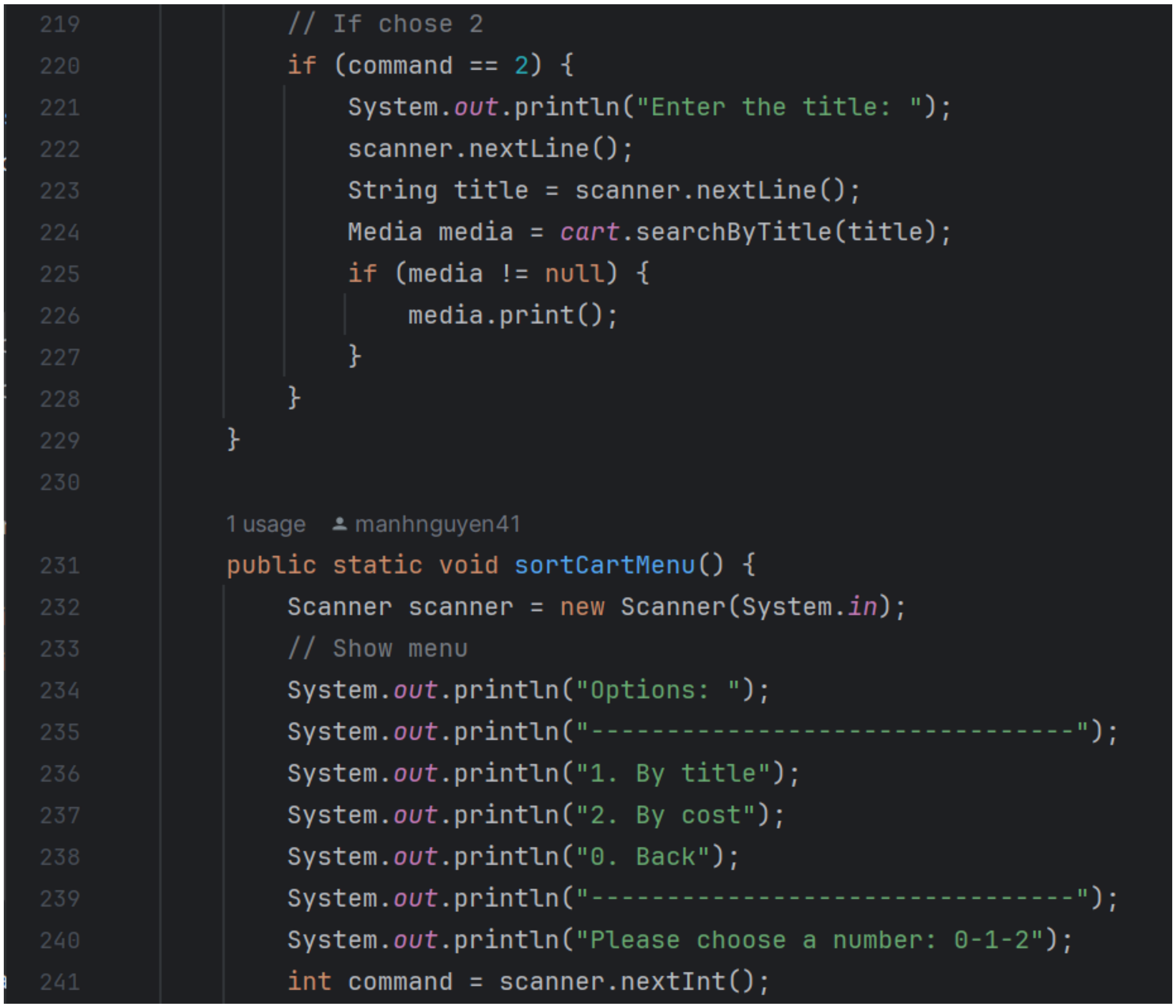
```
        if (command == 2) {
            sortCartMenu();
        }

        // If chose 3
        if (command == 3) {
            removeMediaFromCart();
        }

        // If chose 4
        if (command == 4) {
            Media media;
            do {
                System.out.println("Enter the title of the media: ");
                String title = scanner.nextLine();
                media = cart.searchByTitle(title);
            } while (media == null);
            playAMedia(media);
        }

        // If chose 5
        if (command == 5) {
            System.out.println("Order is created");
            cart = new Cart();
        }
```

Figure 34 Aims class code



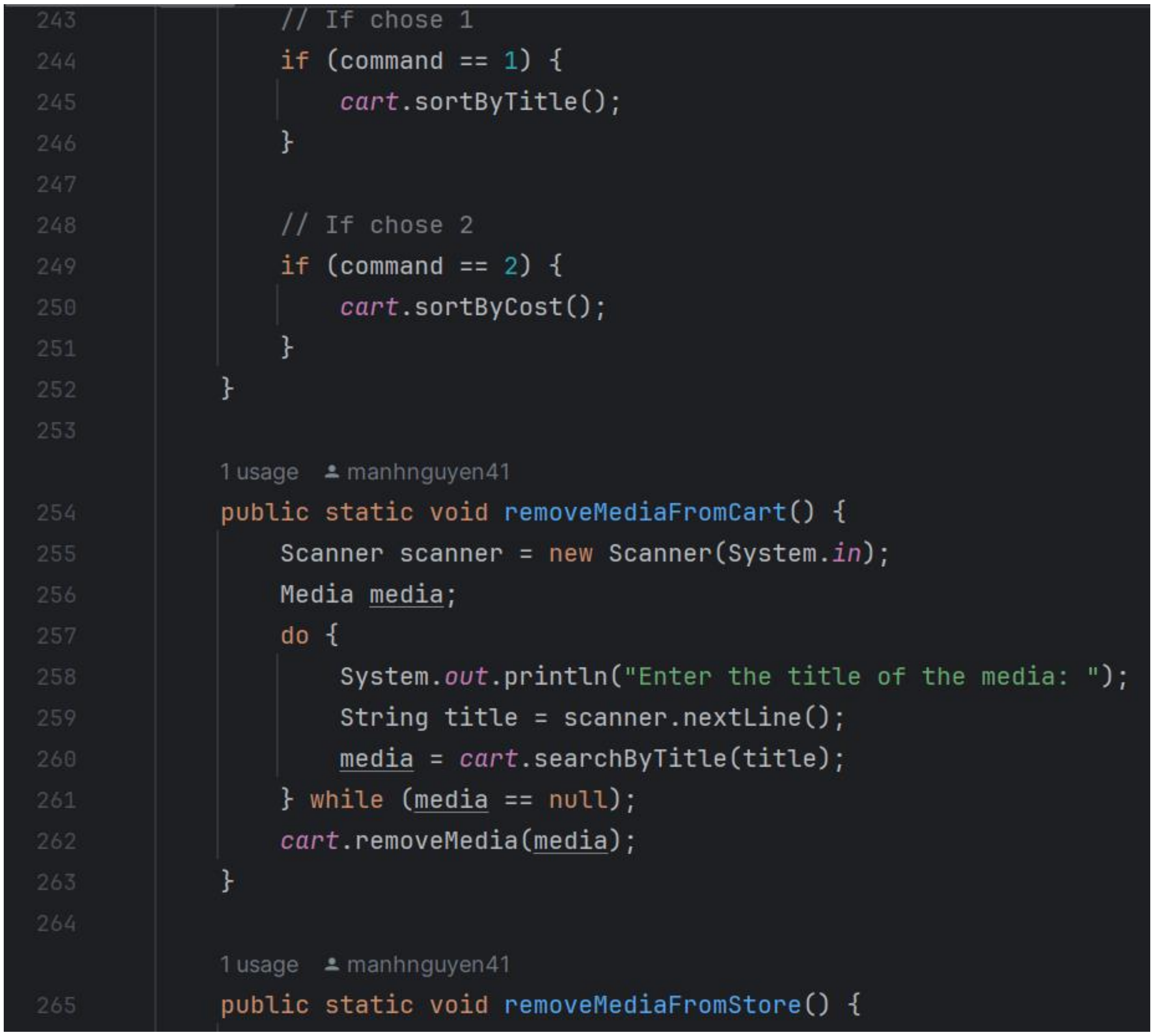
```
    }

    1 usage  manhnguyen41
    public static void filterCartMenu() {
        Scanner scanner = new Scanner(System.in);
        // Show menu
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. By id");
        System.out.println("2. By title");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2");
        int command = scanner.nextInt();

        // If chose 1
        if (command == 1) {
            System.out.println("Enter the id: ");
            int id = scanner.nextInt();
            Media media = cart.searchByID(id);
            if (media != null) {
                media.print();
            }
        }
```

Figure 35 Aims class code



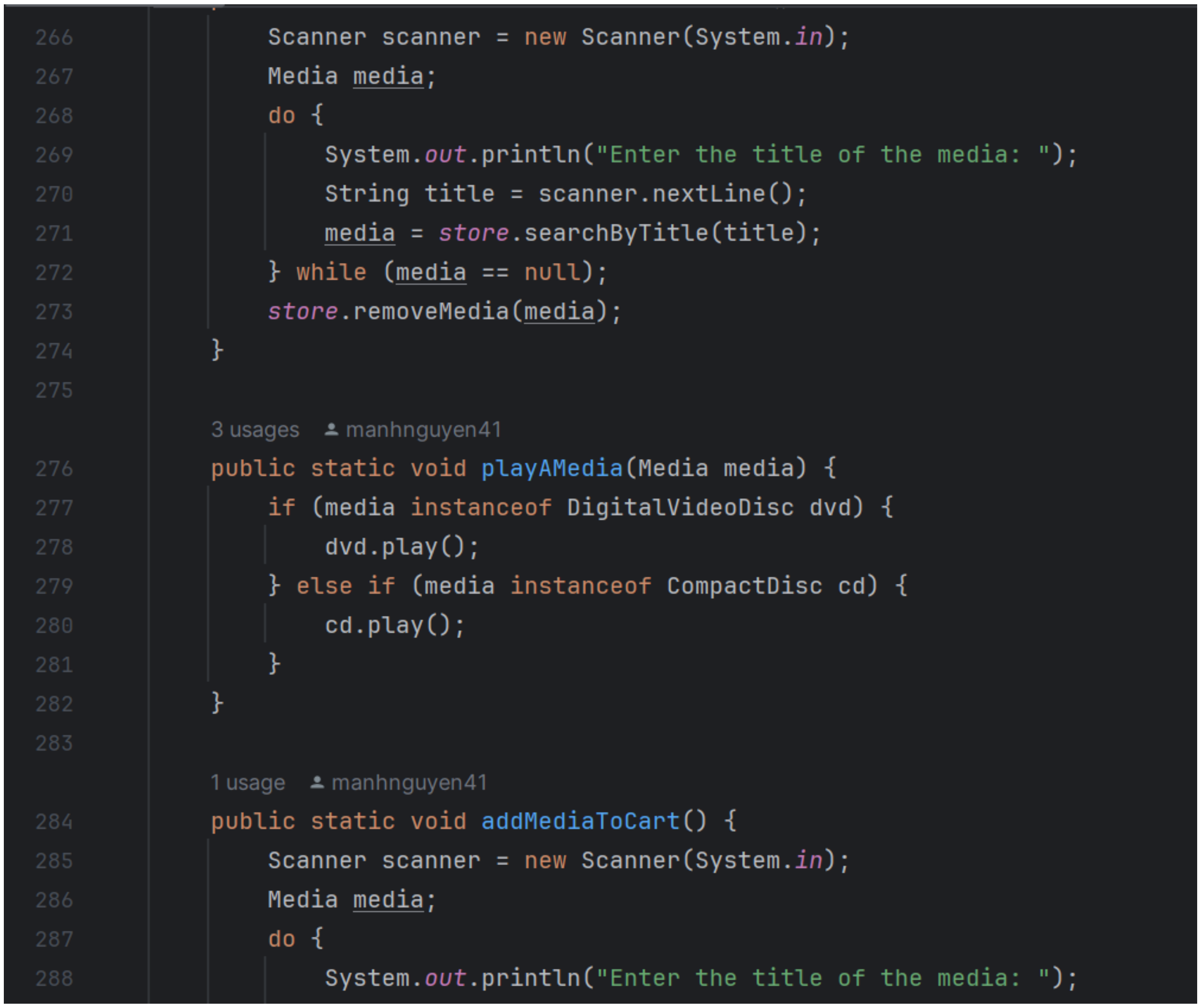
```
        // If chose 2
        if (command == 2) {
            System.out.println("Enter the title: ");
            scanner.nextLine();
            String title = scanner.nextLine();
            Media media = cart.searchByTitle(title);
            if (media != null) {
                media.print();
            }
        }
    }

    1 usage   manhnguyen41
    public static void sortCartMenu() {
        Scanner scanner = new Scanner(System.in);
        // Show menu
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. By title");
        System.out.println("2. By cost");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2");
        int command = scanner.nextInt();
```

Figure 36 Aims class code



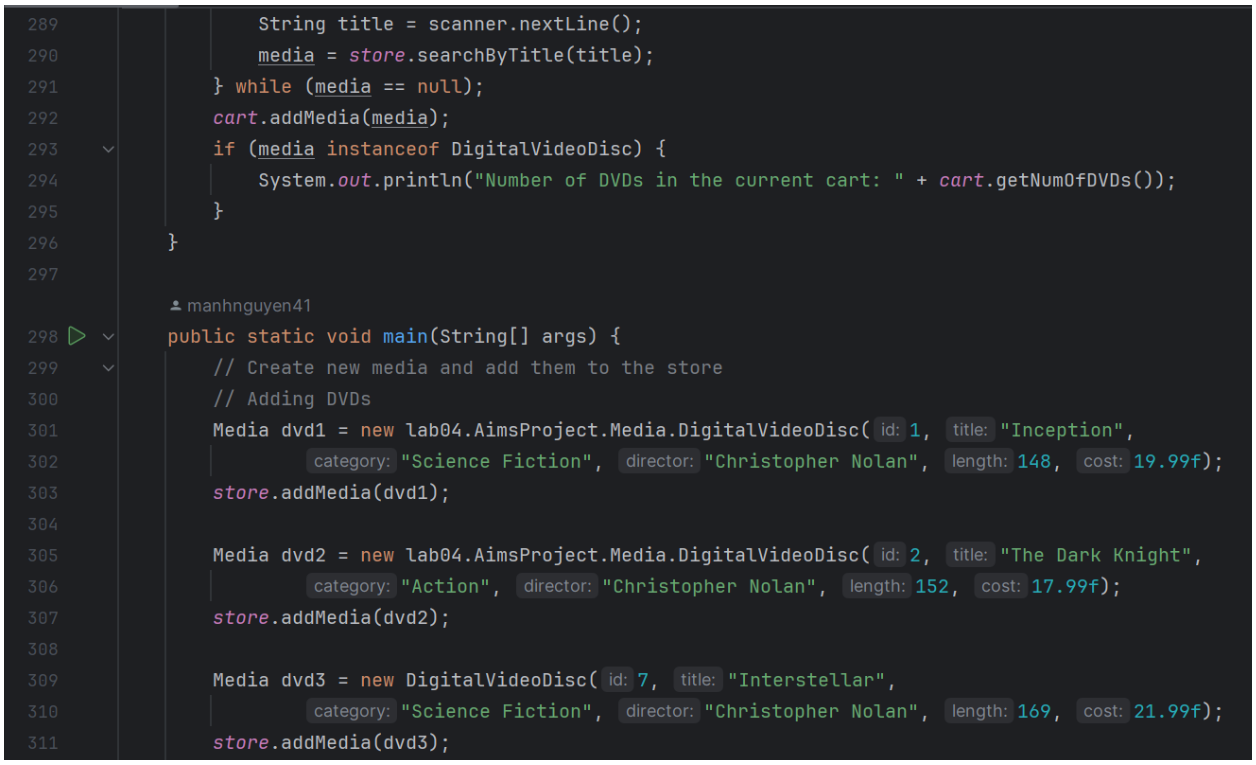
```java
            // If chose 1
            if (command == 1) {
                cart.sortByTitle();
            }

            // If chose 2
            if (command == 2) {
                cart.sortByCost();
            }
        }

        1 usage   manhnguyen41
        public static void removeMediaFromCart() {
            Scanner scanner = new Scanner(System.in);
            Media media;
            do {
                System.out.println("Enter the title of the media: ");
                String title = scanner.nextLine();
                media = cart.searchByTitle(title);
            } while (media == null);
            cart.removeMedia(media);
        }

        1 usage   manhnguyen41
        public static void removeMediaFromStore() {
```

Figure 37 Aims class code



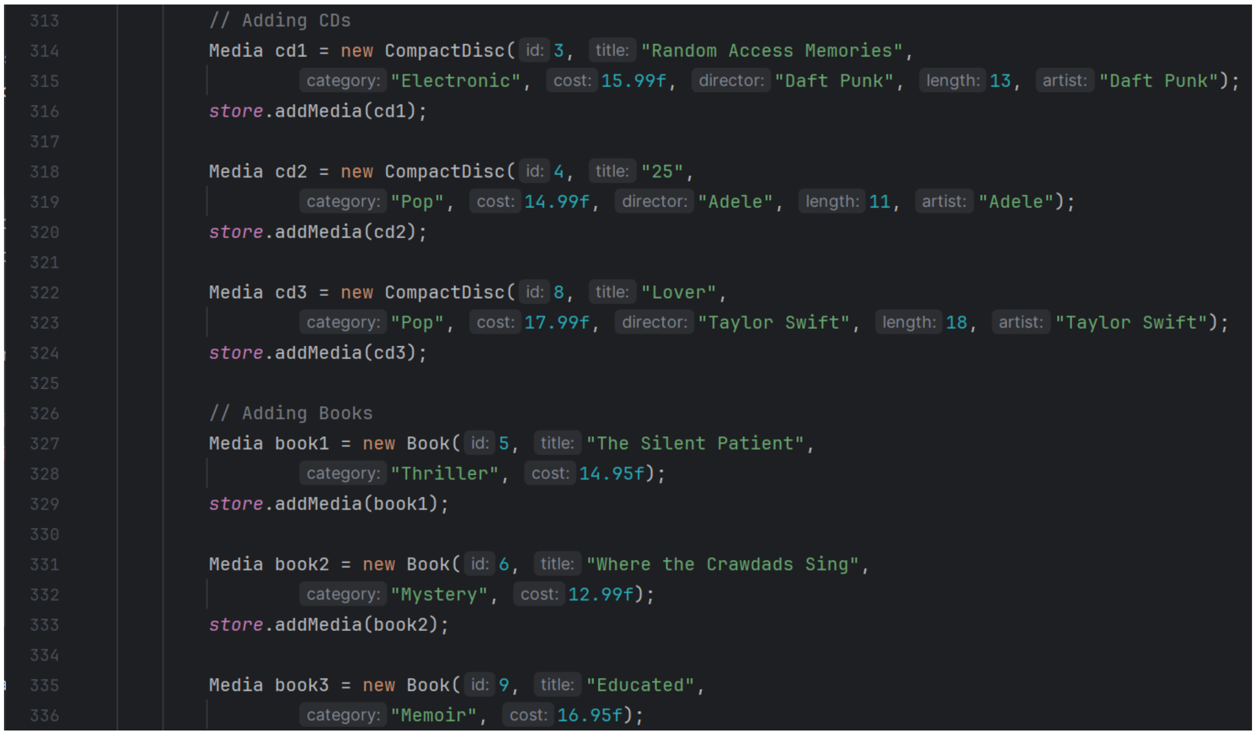
```
        Scanner scanner = new Scanner(System.in);
        Media media;
        do {
            System.out.println("Enter the title of the media: ");
            String title = scanner.nextLine();
            media = store.searchByTitle(title);
        } while (media == null);
        store.removeMedia(media);
    }

    3 usages  manhnguyen41
    public static void playAMedia(Media media) {
        if (media instanceof DigitalVideoDisc dvd) {
            dvd.play();
        } else if (media instanceof CompactDisc cd) {
            cd.play();
        }
    }

    1 usage  manhnguyen41
    public static void addMediaToCart() {
        Scanner scanner = new Scanner(System.in);
        Media media;
        do {
            System.out.println("Enter the title of the media: ");
```

Figure 38 Aims class code



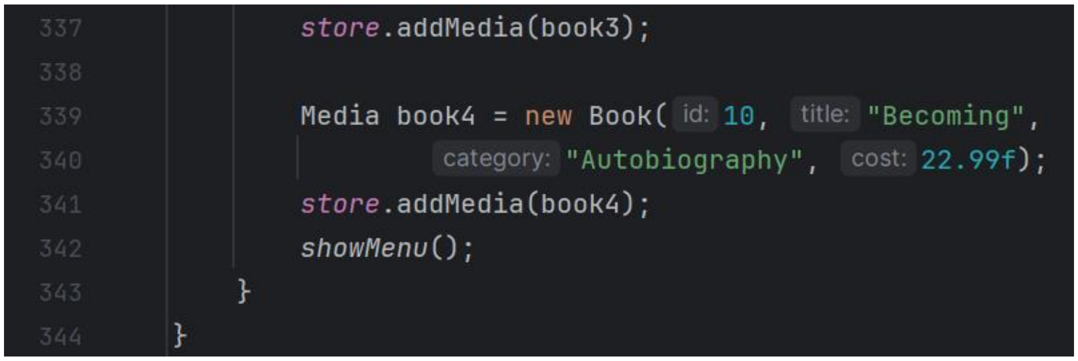
```java
            String title = scanner.nextLine();
            media = store.searchByTitle(title);
        } while (media == null);
        cart.addMedia(media);
        if (media instanceof DigitalVideoDisc) {
            System.out.println("Number of DVDs in the current cart: " + cart.getNumOfDVDs());
        }
    }

    manhnguyen41
    public static void main(String[] args) {
        // Create new media and add them to the store
        // Adding DVDs
        Media dvd1 = new lab04.AimsProject.Media.DigitalVideoDisc( id: 1, title: "Inception",
                category: "Science Fiction", director: "Christopher Nolan", length: 148, cost: 19.99f);
        store.addMedia(dvd1);

        Media dvd2 = new lab04.AimsProject.Media.DigitalVideoDisc( id: 2, title: "The Dark Knight",
                category: "Action", director: "Christopher Nolan", length: 152, cost: 17.99f);
        store.addMedia(dvd2);

        Media dvd3 = new DigitalVideoDisc( id: 7, title: "Interstellar",
                category: "Science Fiction", director: "Christopher Nolan", length: 169, cost: 21.99f);
        store.addMedia(dvd3);
```

Figure 39 Aims class code



```java
        // Adding CDs
        Media cd1 = new CompactDisc( id: 3, title: "Random Access Memories",
                category: "Electronic", cost: 15.99f, director: "Daft Punk", length: 13, artist: "Daft Punk");
        store.addMedia(cd1);

        Media cd2 = new CompactDisc( id: 4, title: "25",
                category: "Pop", cost: 14.99f, director: "Adele", length: 11, artist: "Adele");
        store.addMedia(cd2);

        Media cd3 = new CompactDisc( id: 8, title: "Lover",
                category: "Pop", cost: 17.99f, director: "Taylor Swift", length: 18, artist: "Taylor Swift");
        store.addMedia(cd3);

        // Adding Books
        Media book1 = new Book( id: 5, title: "The Silent Patient",
                category: "Thriller", cost: 14.95f);
        store.addMedia(book1);

        Media book2 = new Book( id: 6, title: "Where the Crawdads Sing",
                category: "Mystery", cost: 12.99f);
        store.addMedia(book2);

        Media book3 = new Book( id: 9, title: "Educated",
                category: "Memoir", cost: 16.95f);
```

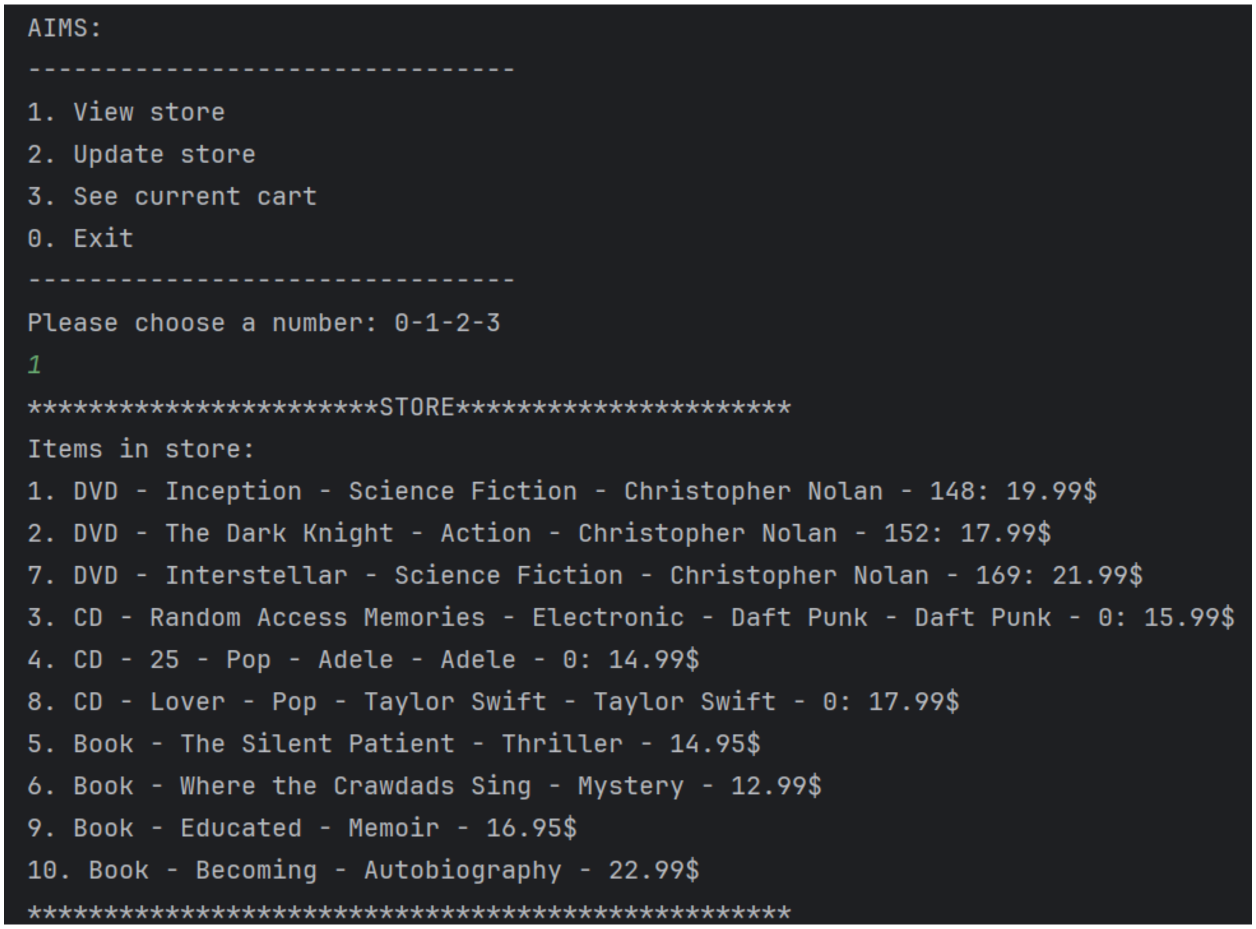
Figure 40 Aims class code



```
            store.addMedia(book3);

            Media book4 = new Book( id: 10,  title: "Becoming",
                    category: "Autobiography",  cost: 22.99f);
            store.addMedia(book4);
            showMenu();
        }
}
```

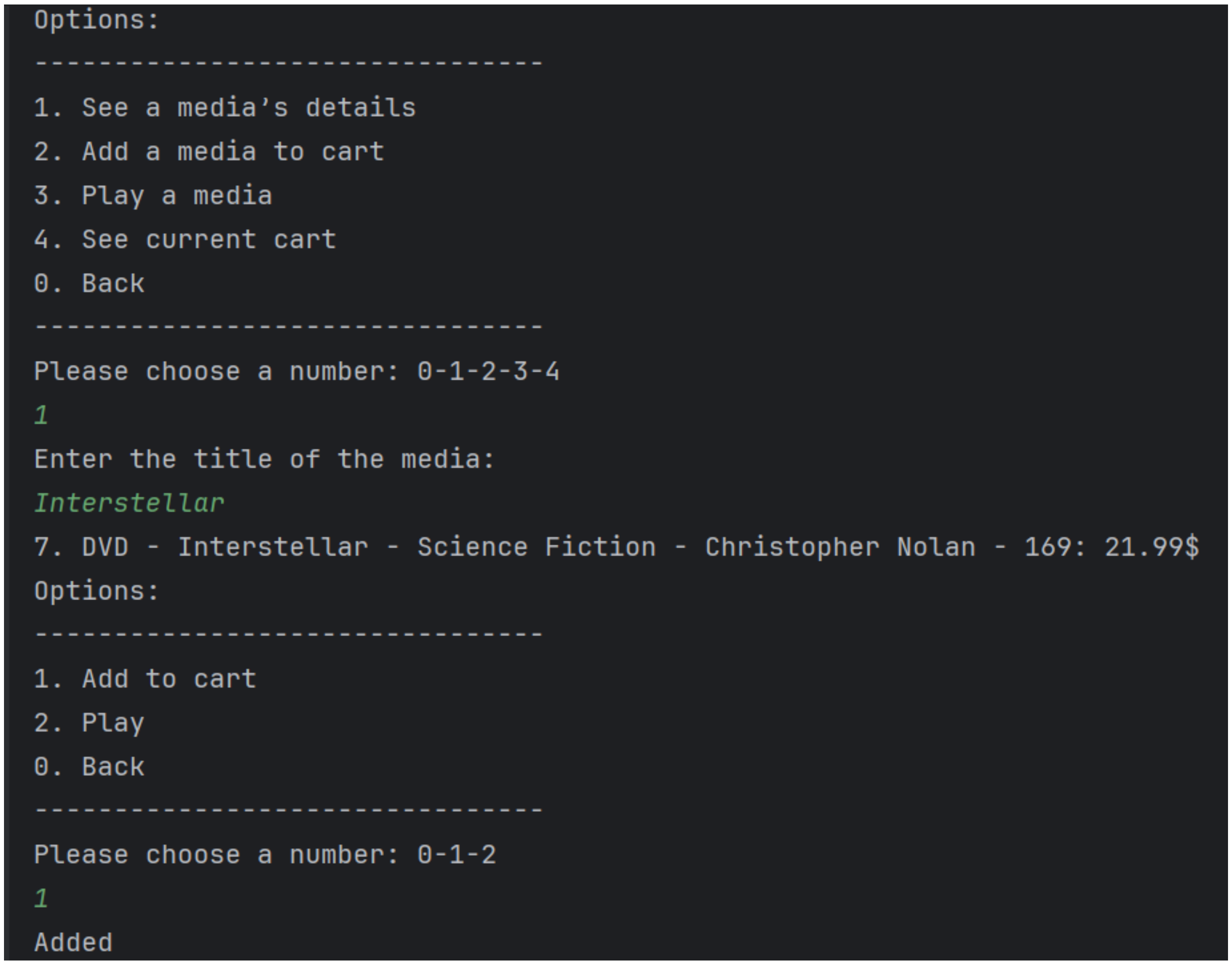
Figure 41 Aims class code

11. Demo Aims



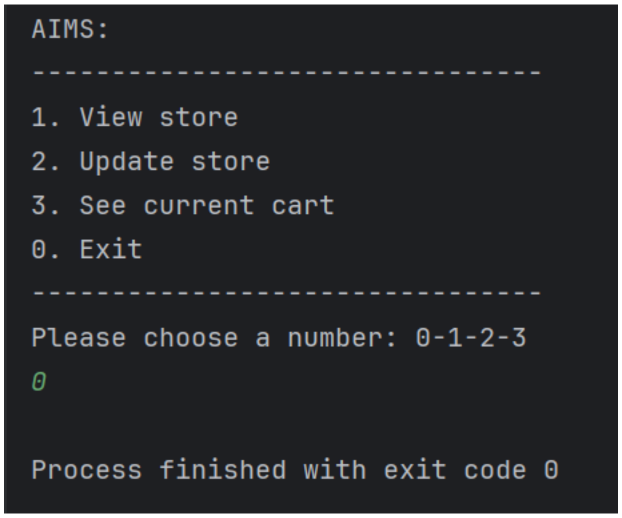
```
AIMS:
--------------------------------
1. View store
2. Update store
3. See current cart
0. Exit
--------------------------------
Please choose a number: 0-1-2-3
1
***********************STORE***********************
Items in store:
1. DVD - Inception - Science Fiction - Christopher Nolan - 148: 19.99$
2. DVD - The Dark Knight - Action - Christopher Nolan - 152: 17.99$
7. DVD - Interstellar - Science Fiction - Christopher Nolan - 169: 21.99$
3. CD - Random Access Memories - Electronic - Daft Punk - Daft Punk - 0: 15.99$
4. CD - 25 - Pop - Adele - Adele - 0: 14.99$
8. CD - Lover - Pop - Taylor Swift - Taylor Swift - 0: 17.99$
5. Book - The Silent Patient - Thriller - 14.95$
6. Book - Where the Crawdads Sing - Mystery - 12.99$
9. Book - Educated - Memoir - 16.95$
10. Book - Becoming - Autobiography - 22.99$
***************************************************
```

Figure 42 Aims demo



```
Options:
--------------------------------
1. See a media's details
2. Add a media to cart
3. Play a media
4. See current cart
0. Back
--------------------------------
Please choose a number: 0-1-2-3-4
1
Enter the title of the media:
Interstellar
7. DVD - Interstellar - Science Fiction - Christopher Nolan - 169: 21.99$
Options:
--------------------------------
1. Add to cart
2. Play
0. Back
--------------------------------
Please choose a number: 0-1-2
1
Added
```

Figure 43 Aims demo



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

Process finished with exit code 0
```

Figure 44 Aims demo

12. Update the class diagram

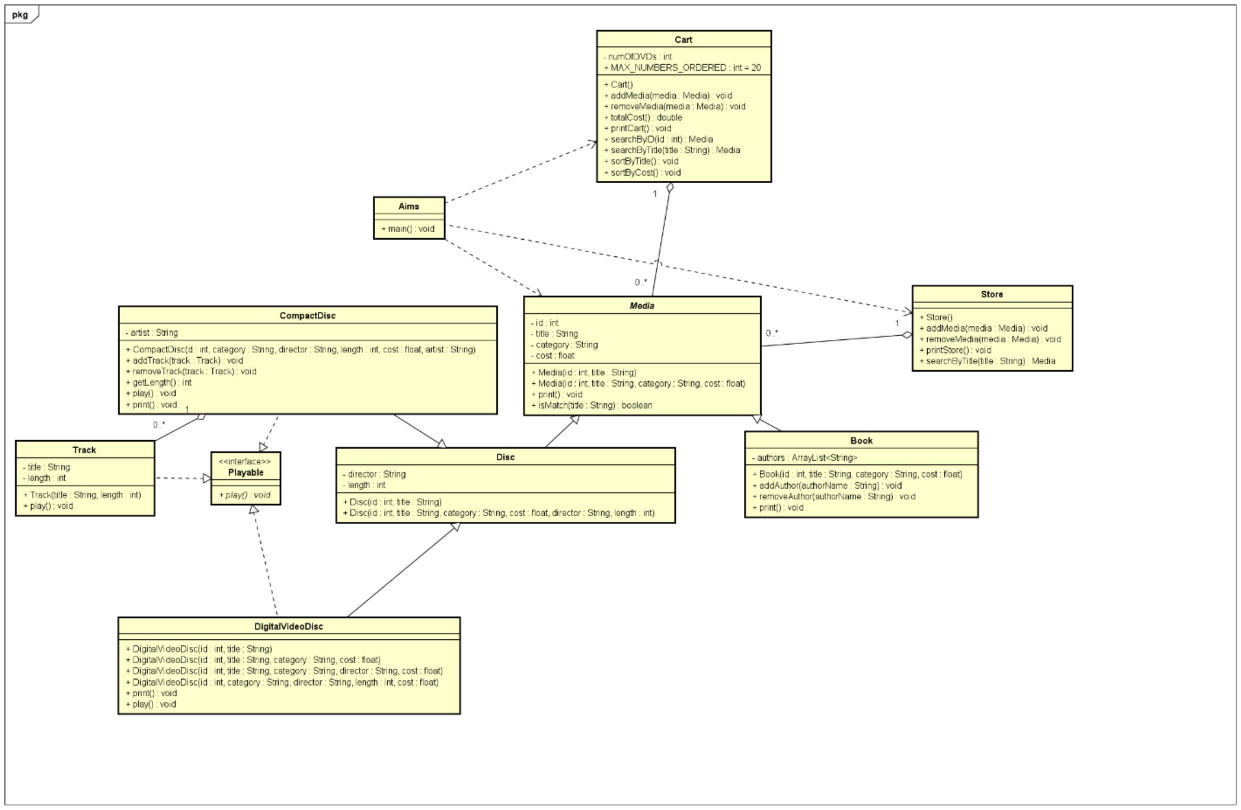

pkg
Cart
- numOfDVDs : int
+ MAX_NUMBERS_ORDERED : int = 20
+ Cart()
+ addMedia(media : Media) : void
+ removeMedia(media : Media) : void
+ totalCost() : double
+ printCart() : void
+ searchByID(id : int) : Media
+ searchByTitle(title : String) : Media
+ sortByTitle() : void
+ sortByCost() : void
Aims
+ main() : void
1
0..*
Store
+ Store()
+ addMedia(media : Media) : void
+ removeMedia(media : Media) : void
+ printStore() : void
+ searchByTitle(title : String) : Media
0..*
1
CompactDisc
- artist : String
+ CompactDisc(d : int, category : String, director : String, length : int, cost : float, artist : String)
+ addTrack(track : Track) : void
+ removeTrack(track : Track) : void
+ getLength() : int
+ play() : void
+ print() : void
Media
- id : int
- title : String
- category : String
- cost : float
+ Media(id : int, title : String)
+ Media(id : int, title : String, category : String, cost : float)
+ print() : void
+ isMatch(title : String) : boolean
1
0..*
Track
- title : String
- length : int
+ Track(title : String, length : int)
+ play() : void
<<interface>>
Playable
+ play() : void
Disc
- director : String
- length : int
+ Disc(id : int, title : String)
+ Disc(id : int, title : String, category : String, cost : float, director : String, length : int)
Book
- authors : ArrayList<String>
+ Book(id : int, title : String, category : String, cost : float)
+ addAuthor(authorName : String) : void
+ removeAuthor(authorName : String) : void
+ print() : void
DigitalVideoDisc
+ DigitalVideoDisc(id : int, title : String)
+ DigitalVideoDisc(id : int, title : String, category : String, cost : float)
+ DigitalVideoDisc(id : int, title : String, category : String, director : String, cost : float)
+ DigitalVideoDisc(id : int, category : String, director : String, length : int, cost : float)
+ print() : void
+ play() : void

Figure 45 Class diagram

13. Update the usecase diagram

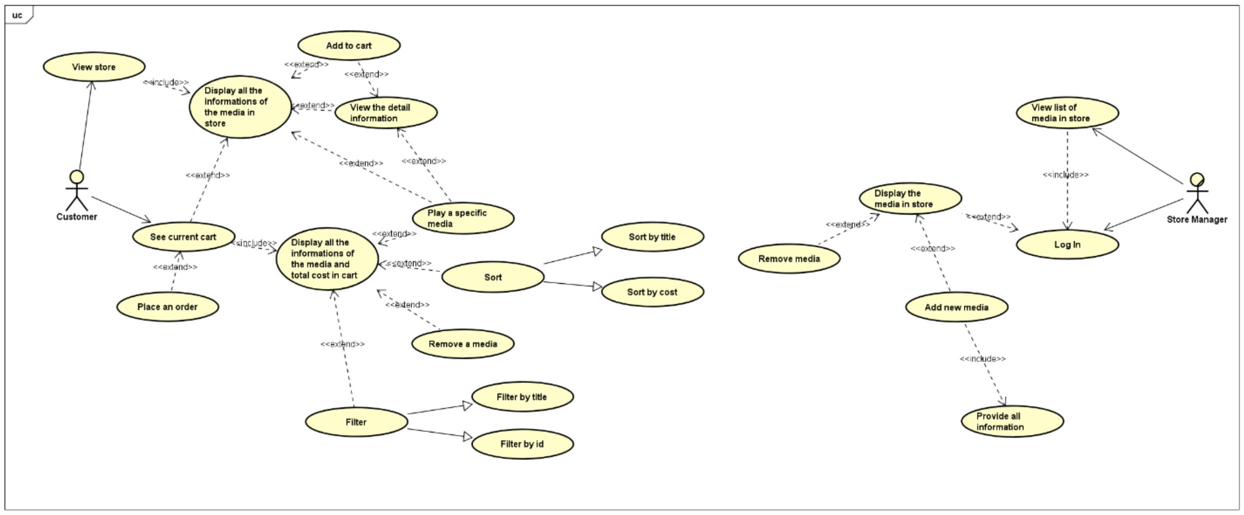
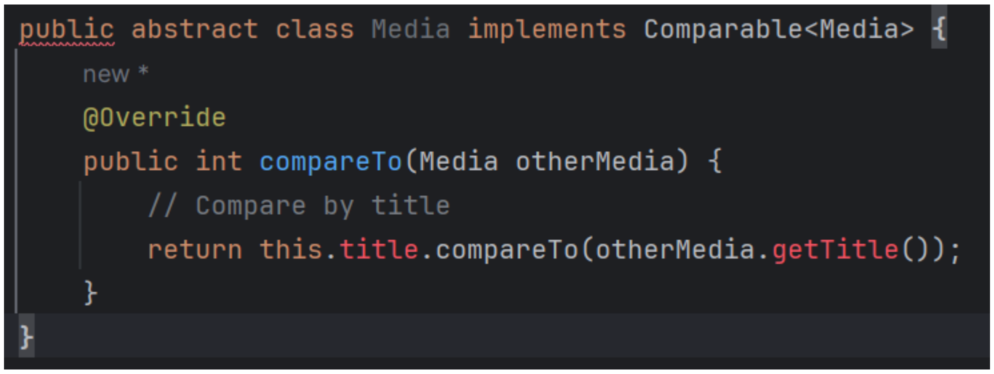


Figure 46 Usecase diagram

14. Answer the questions

-Trong trường hợp cần so sánh các Media với nhau bằng cách implement Comparable thay vì Comparator, thì thay vì tạo class riêng cho từng Comparator, ta cần cho class Media implement interface Comparable.

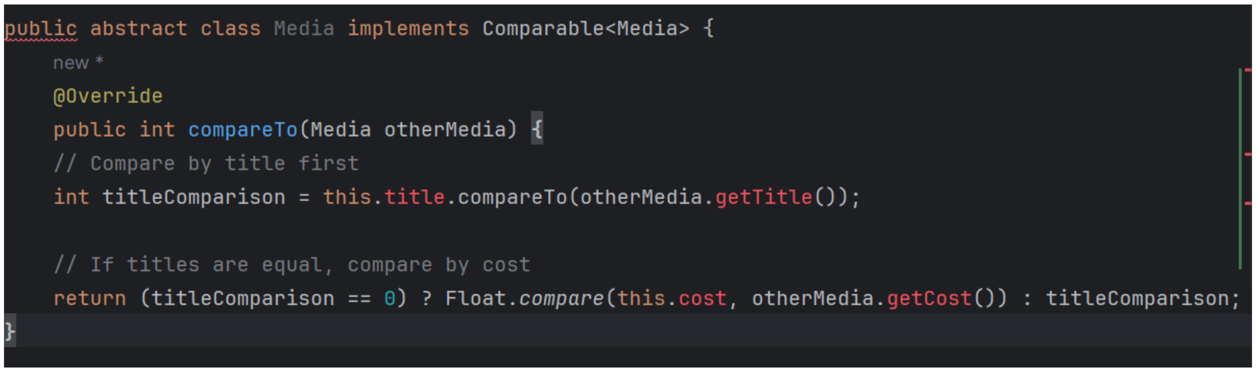
-Ví dụ như sau:



```java
public abstract class Media implements Comparable<Media> {
    new *
    @Override
    public int compareTo(Media otherMedia) {
        // Compare by title
        return this.title.compareTo(otherMedia.getTitle());
    }
}
```

Figure 47 Question code

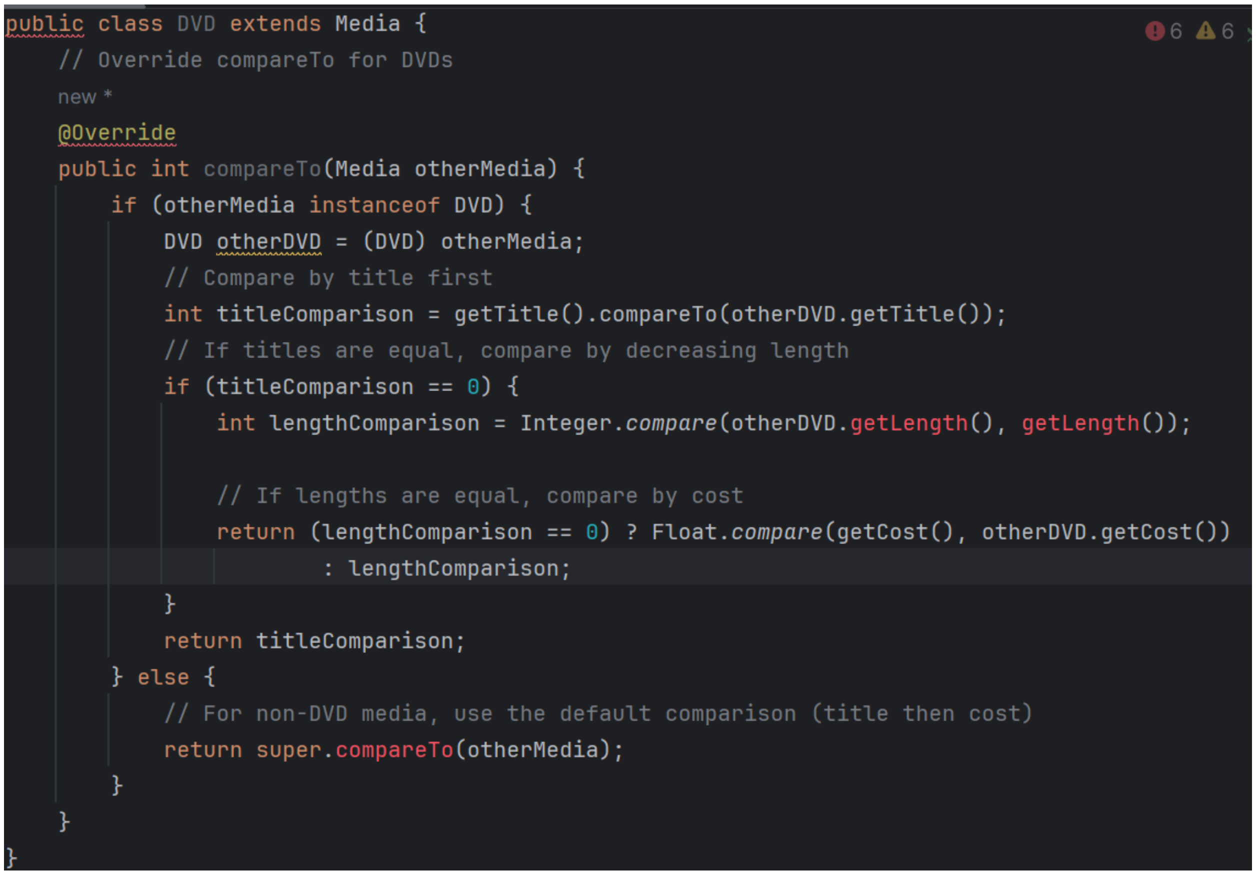
-Có thể, cài đặt như sau:



```java
public abstract class Media implements Comparable<Media> {
    new *
    @Override
    public int compareTo(Media otherMedia) {
    // Compare by title first
    int titleComparison = this.title.compareTo(otherMedia.getTitle());

    // If titles are equal, compare by cost
    return (titleComparison == 0) ? Float.compare(this.cost, otherMedia.getCost()) : titleComparison;
}
```

Figure 48 Question code

-Cài đặt như sau:



```java
public class DVD extends Media {
    // Override compareTo for DVDs
    new *
    @Override
    public int compareTo(Media otherMedia) {
        if (otherMedia instanceof DVD) {
            DVD otherDVD = (DVD) otherMedia;
            // Compare by title first
            int titleComparison = getTitle().compareTo(otherDVD.getTitle());
            // If titles are equal, compare by decreasing length
            if (titleComparison == 0) {
                int lengthComparison = Integer.compare(otherDVD.getLength(), getLength());

                // If lengths are equal, compare by cost
                return (lengthComparison == 0) ? Float.compare(getCost(), otherDVD.getCost())
                        : lengthComparison;
            }
            return titleComparison;
        } else {
            // For non-DVD media, use the default comparison (title then cost)
            return super.compareTo(otherMedia);
        }
    }
}
```

Figure 49 Question code