**BÁO CÁO THỰC HÀNH LAB 4**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Contents**

[**1.** **Đề bài** 1](#_Toc152797331)

[**2.** **Yêu cầu bài toán** 1](#_Toc152797332)

[**3.** **Mã nguồn của chương trình** 2](#_Toc152797333)

[***3.1.*** ***hust.soict.hedspi.aims*** 2](#_Toc152797334)

[***3.2.*** ***hust.soict.hedspi.aims.media*** 17](#_Toc152797335)

[***3.3.*** ***hust.soict.hedspi.test:*** *ToStringTest.java* 26](#_Toc152797336)

[**4.** **Demo chương trình** 27](#_Toc152797337)

# **Đề bài**

Thiết kế hệ thống mới cho dự án AIMS

# **Yêu cầu bài toán**

* Đối với người mua hàng
* Duyệt danh sách các DVD có sẵn trong cửa hàng
* Tìm kiếm DVD theo: tiêu đề, danh mục, giả cả
* Xem thông tin chi tiết của 1 DVD
* Thêm DVD vào giỏ hàng
* Xem giỏ hàng
* Sắp xếp DVD trong giỏ hàng theo tiêu đề hoặc chi phí
* Cập nhật số lương DVD trong giỏ hàng
* Đặt hàng:
* Đối với người quản lý cửa hàng
* Đăng nhập, kiểm tra quyền
* Xem danh sách các đơn hàng đang chờ xử lý
* Xem chi tiết 1 đơn hàng và xử lý nó (đồng ý từ chối đặt hàng)
* Thêm (xóa) DVD mới vào (ra) danh sách sản phẩm

# **Mã nguồn của chương trình**

* 1. ***hust.soict.hedspi.aims***

1. Aims.java

//Nguyen Van Cuong - 20215006

package hust.soict.hedspi.aims;

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

import java.util.Scanner;

public class Aims {

    private static String mediaTitle;

    public static Scanner scanner = new Scanner(System.in);

    public static Store aStore;

    public static Cart aCart = new Cart();

    private static Media foundMedia;

    // Khai báo các biến static

    public static void main(String[] args)

   {   aStore = new Store();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",

                         "Animation", "Roger Allers", 87, 19.95f );

        aStore.addMedia(dvd1);

        DigitalVideoDisc dvd2 = new  DigitalVideoDisc("Star Wars",

                "Science Fiction", "George Lucas", 87, 24.95f );

        aStore.addMedia(dvd2);

        DigitalVideoDisc dvd3 = new  DigitalVideoDisc("Aladin",

                "Animation", 18.99f );

        aStore.addMedia(dvd3);

        CompactDisc cd1 = new CompactDisc("Disco Elysium", "High Classic",12.3f);

        CompactDisc cd2 = new CompactDisc("Divine dive", "Blue", 16f);

        aStore.addMedia(cd1);

        aStore.addMedia(cd2);

        Book book1 = new Book("Introduction to Demons summoning", "Occult",12f);

        Book book2 = new Book("High and low", "shitpost", 120f);

        aStore.addMedia(book1);

        aStore.addMedia(book2);

     // Tạo các media với các loại khác nhau và thêm vào store

        showMenu();

        // Chạy hàm showMenu tạo giao diện cho người dùng

   }

   public static void showMenu() {

       //Hàm cho thấy menu

       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");

       //In ra các lựa chọn

       int choice = scanner.nextInt();

       //Lấy lựa chọn

       switch (choice){

           case 1: storeMenu(); break;

           case 2: updateStore(); break;

           case 3: cartMenu(); break;

           //Với các lựa chọn hợp lệ chạy hàm tương ứng

           case 0: return;

           //Với 0 kết thúc giao diện

           default:

               System.out.println("Error, choose again!");

               //Không báo lỗi và quay lại giao diện

               showMenu();

       }

   }

   public static void storeMenu() {

       //Hàm cho thấy có gì trong store

       aStore.showStore();

       //In ra những gì đan cos trong store

       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");

       //In ra các lựa chọn

       int choice = scanner.nextInt();

       //Lấy lựa chọn

       switch (choice){

           case 0: showMenu(); break;

           case 1:

           System.out.println("Enter media's title: ");

           scanner.nextLine();

           mediaTitle = scanner.nextLine();

           foundMedia = aStore.searchByTitle(mediaTitle);

           if (foundMedia != null) {

               mediaDetailsMenu();

           }

           //Trong lựa chọn 1 , lấy tên media rồi tìm nó trong store, nếu tìm thấy vào mediaDetailsMenu

           //Xong quay trở lại giao diện

           storeMenu();

           break;

           case 2:

               System.out.println("Enter media's title");

                scanner.nextLine();

                String mediaTitleAdd = scanner.nextLine();

                foundMedia = aStore.searchByTitle(mediaTitleAdd);

                //Lấy tên media rồi tìm nó trong store và lưu kết quả vào biến foundMedia

                if (foundMedia != null) {

                    aCart.addMedia(foundMedia);

                    aStore.removeMedia(foundMedia);

                    System.out.println("Media added to cart successfully");

                    //Nếu có kết quả thì thêm vào giỏ hàng và bỏ khỏi store

                } else {

                    System.out.println("Media not found in the store");

                    //Không thì báo không tìm thấy

                }

                storeMenu();

               break;

           case 3:

               System.out.println("Enter media's title");

               scanner.nextLine();

               String mediaTitlePlay = scanner.nextLine();

               foundMedia = aStore.searchByTitle(mediaTitlePlay);

               //Lấy tên media rồi tìm nó trong store và lưu kết quả vào biến foundMedia

               if (foundMedia instanceof CompactDisc) {

                    CompactDisc cd = (CompactDisc) foundMedia;

                    cd.play();

                } else if (foundMedia instanceof DigitalVideoDisc) {

                    DigitalVideoDisc dvd = (DigitalVideoDisc) foundMedia;

                    dvd.play();

                }

               //Kiểm tra xem media có phải loại có thể play không rồi play

                else {

                System.out.println("Media can't be played");

                //Nếu không báo lỗi

            }

               storeMenu();

               break;

           case 4:

               cartMenu();

               break;

           default:

               System.out.println("Error, choose again!");

               storeMenu();

       }

       }

   public static void mediaDetailsMenu() {

       //method cho xem thông tin của media đã chọn

       int check = 0;

       System.out.println("Options: ");

       System.out.println("--------------------------------");

       System.out.println("1. Add to cart");

       if (foundMedia instanceof CompactDisc || foundMedia instanceof DigitalVideoDisc) {

            System.out.println("2. Play");

            check = 1;

            //Kiểm tra xem media xem có phải loại play được không để in ra lựa chọn

        }

       System.out.println("0. Back");

       System.out.println("--------------------------------");

       System.out.println("Please choose a number: 0-1-(2)");

       int detailMenuChoice = scanner.nextInt();

       switch (detailMenuChoice){

       case 1:

            aCart.addMedia(foundMedia);

            aStore.removeMedia(foundMedia);

            System.out.println("Media added to cart successfully");

            //Thểm media lưu trong biến foundMedia từ trước vào cart rồi bỏ khỏi store

          storeMenu();

           break;

       case 2:

           if (check == 0)

           {

               System.out.println("Error, choose again!");

               mediaDetailsMenu();

               //Nếu media không phải loại play được thì không có lựa chọn 2, coi nó như lựa chọn lõi

           }

           else

           {

               if (foundMedia instanceof CompactDisc) {

                CompactDisc cd = (CompactDisc) foundMedia;

                cd.play();

            } else if (foundMedia instanceof DigitalVideoDisc) {

                DigitalVideoDisc dvd = (DigitalVideoDisc) foundMedia;

                dvd.play();

            }

               //Nếu là loại play được thì ép kiểu rồi dùng hàm play tương ứng

           }

           break;

       case 0:

           storeMenu();

           break;

       default:

           System.out.println("Error, choose again!");

           mediaDetailsMenu();

   }

       }

   public static void updateStore(){

       //Method cho cập nhật cửa hàng

       Media item;

       // In các lựa chọn

       System.out.println("Options: ");

       System.out.println("--------------------------------");

       System.out.println("1. Add new media.");

       System.out.println("2. Remove media.");

       System.out.println("0. Back");

       System.out.println("---------------------------------");

       System.out.println("Please choose a number 0-1-2");

       int updateStoreChoice = scanner.nextInt();

       switch (updateStoreChoice){

           case 0: showMenu(); break;

           case 1:

               AddMedia();

               break;

           case 2:

               System.out.println("Removed media");

               scanner.nextLine();

               String removeFromStore = scanner.nextLine();

               //Đọc tên media

               item = aStore.searchByTitle(removeFromStore);

               //Tìm trong cửa hàng media đó

               if (item != null) {

                System.out.println("Media removed successfully");

                aStore.removeMedia(item);

                //nếu có thì loại bỏ và báo

            } else {

                System.out.println("Can't remove");

                //Nếu không báo lỗi

            }

               updateStore();

               break;

           default:

               System.out.println("Error, choose again!");

               updateStore();

       }

   }

   public static void AddMedia(){

       //Method cho việc thêm media trong store

       String Title;

       //In ra các lựa chọn

       System.out.println("Options: ");

       System.out.println("--------------------------------");

       System.out.println("1. Add a book.");

       System.out.println("2. Add a DVD.");

       System.out.println("3. Add a CD.");

       System.out.println("0. Back");

       System.out.println("---------------------------------");

       System.out.println("Please choose a number 0-1-2-3");

       int updateStoreChoice = scanner.nextInt();

       scanner.nextLine();

       switch (updateStoreChoice){

           case 0: updateStore(); break;

           case 1:

               System.out.println("Title");

               Title = scanner.nextLine();

               //Lấy tên của book

               Book book = new Book(Title);

               //Tạo một instance mới rồi thêm vào store

               aStore.addMedia(book);

               updateStore();

               break;

           case 2:

               System.out.println("Title");

               Title = scanner.nextLine();

               //Lấy tên

               DigitalVideoDisc dvd = new DigitalVideoDisc(Title);

               aStore.addMedia(dvd);

             //Tạo một instance mới rồi thêm vào store

               updateStore();

               break;

           case 3:

               System.out.println("Title");

               Title = scanner.nextLine();

               //Lấy tên

               CompactDisc cd = new CompactDisc(Title);

               aStore.addMedia(cd);

               updateStore();

               break;

           default:

               System.out.println("Error, choose again!");

               updateStore();

       }

   }

   public static void cartMenu() {

       //Method cho xem trong cart có gì

       Media item;

       aCart.showCart();

       //In ra những gì có trong Cart

       System.out.println("Options: ");

       System.out.println("--------------------------------");

       System.out.println("1. Filter media in cart");

       System.out.println("2. Sort media 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 cartMenuChoice = scanner.nextInt();

       switch (cartMenuChoice){

           case 1:

               filterMenu();

               break;

           case 2:

               sortMenu();

               break;

           case 3:

               System.out.println("Enter media's title: ");

               scanner.nextLine();

               String removeMediaTitle = scanner.nextLine();

               item = aCart.searchByTitle(removeMediaTitle);

               //Lấy tên Media muốn xóa rồi tìm nó

               if (item != null) {

                    System.out.println("Media removed successfully");

                    aCart.removeMedia(item);

                } else {

                    System.out.println("Can't removed");

                }

               //Tìm được thì xóa không thì báo lỗi

               cartMenu();

               break;

           case 4:

               System.out.println("Enter media's title");

               scanner.nextLine();

               String mediaTitlePlay = scanner.nextLine();

               item =  aCart.searchByTitle(mediaTitlePlay);

               //Lấy tên rồi tìm media để play

               if ( item instanceof CompactDisc) {

                CompactDisc cd = (CompactDisc) item;

                cd.play();

            } else if (item instanceof DigitalVideoDisc) {

                DigitalVideoDisc dvd = (DigitalVideoDisc) item;

                dvd.play();

            }

                else {

                System.out.println("Media not found in the store");

            }

               //Nếu play được thì play không thì báo lỗi

               cartMenu();

               break;

           case 5:

               System.out.println("An order is created");

               cartMenu();

               break;

           case 0:

               storeMenu();

               break;

           default:

               System.out.println("Error, choose again");

               cartMenu();

       }

   }

   public static void filterMenu(){

       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 filterChoice = scanner.nextInt();

       switch (filterChoice){

           case 1:

               System.out.println("Enter media's ID: ");

               int filterID = scanner.nextInt();

               //Lấy Id rồi tìm các id khớp

               aCart.searchById(filterID);

               filterMenu();

               break;

           case 2:

               System.out.println("Enter media's title: ");

               scanner.nextLine();

               String filterTitle = scanner.nextLine();

               //Lấy title rồi tìm các title khớp

               aCart.searchByTitle(filterTitle);

               filterMenu();

               break;

           case 0:

               cartMenu();

               break;

           default:

               System.out.println("Error, choose again");

               filterMenu();

       }

   }

   public static void sortMenu(){

       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 sortChoice = scanner.nextInt();

       switch (sortChoice){

           case 1:

               aCart.sortByTitle();

               aCart.showCart();

               //Sort bằng title rồi show lại cart

               sortMenu();

               break;

           case 2:

               aCart.sortByCost();

               aCart.showCart();

               //Sortbycost rồi show lại cart

               sortMenu();

               break;

           case 0:

               cartMenu();

               break;

           default:

               System.out.println("Error, choose again");

               sortMenu();

       }

   }

}

1. Store.java

//Nguyen Van Cuong – 20215006

package hust.soict.hedspi.aims;

import java.util.ArrayList;

import java.util.List;

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

public class Store {

    private List<Media> itemsInStore =

            new ArrayList<>();

    public void addMedia (Media item)

    {   // Method cho việc thêm item

            //Thêm item vào store và thông báo thêm thành công

            itemsInStore.add(item);

            System.out.println("The item has been added");

    }

    public void removeMedia (Media item)

    {       // Method cho việc loại bỏ disc

            if (itemsInStore.isEmpty())

            {   //Nếu list trống thì thông báo cart trống

                System.out.println("The shop is empty");

                return;

            }

            if (itemsInStore.remove(item))

            {   // Nếu loại bỏ thành công thì gửi thông báo

                System.out.println("The item is removed successfully");

            }

            else

            {   // Nếu không loại bỏ thành công thì gửi thông báo disc không có trong cart

                System.out.println("The item is not in the shop");

            }

    }

    public void showStore()

    {   // Method để liệt kê những gì có trong store và chi phí của chúng

        int i = 1;

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Store\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

        for (Media item : itemsInStore) {

             System.out.println(i + ".item - " + item.toString());

             i++;

        // Duyệt lần lượt qua Store và in kết quả

        }

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    }

    public Media searchByTitle(String Title)

    { // Method để tìm bằng title

        int count = 1;

        Media result = null;

        for (Media item : itemsInStore) {

            if (item.isMatch(Title))

                {// Duyệt lần lượt qua cart và kiểm tra xem title của dvd có chứa xâu dùng để tra không

                System.out.println("Item -" + item.toString());

                result = item;

                count ++;

                // nếu có thì in ra và tăng count

                }

       }

        if (count == 1)

        {   //Nếu count không đổi thì in ra không tìm được

            System.out.println("No results found");

            return null;

            //trả về null nếu không tìm thấy

        }

        return result;

        //trả về kết quả cuối thỏa mãn

        }

    }

1. Cart.java

//Nguyen Van Cuong - 20215006

package hust.soict.hedspi.aims;

import java.util.ArrayList;

import java.util.List;

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

public class Cart {

    public static final int MAX\_NUMBERS\_ORDERED = 20;

    private List<Media> itemOrdered =

            new ArrayList<>();

    // Sử dụng list để tiện cho việc remove

    public void addMedia (Media item)

    {   // Method cho việc thêm media

        if ( itemOrdered.size() == MAX\_NUMBERS\_ORDERED)

            { //Nếu kích thước của list = tối đa thì gửi thông báo

             System.out.println("The cart is almost full");

             return;

            }

        else

           {

            // Nếu không thì thêm media vào cart và thông báo thêm thành công

            itemOrdered.add(item);

            System.out.println("The item has been added");

           }

    }

    public void addMedia (Media ... item)

    {   // Method cho việc thêm disc với arg tùy ý

        if ( itemOrdered.size() == MAX\_NUMBERS\_ORDERED)

            { //Nếu kích thước của list = tối đa thì gửi thông báo

             System.out.println("The cart is almost full");

             return;

            }

        else

           {

            if ( itemOrdered.size() + item.length == MAX\_NUMBERS\_ORDERED)

             {  // Nếu cart không đủ chỗ để thêm list thì gửi thông báo

                System.out.println("There are not enough spots in the cart");

             }

            else

             {

                for (Media it : item) {

                    itemOrdered.add(it);

                // Dùng vòng lặp và thêm từ từ item từ list vào

                }

                System.out.println("The list of items has been added successfully");

             }

           }

    }

    public void addMedia (Media item1, Media item2 )

    {   // Method cho việc thêm media với 2 arg

        if ( itemOrdered.size() + 2 >  MAX\_NUMBERS\_ORDERED)

            { //Nếu cart không đủ chỗ để chứa thì gửi thông báo thì gửi thông báo

             System.out.println("The cart is almost full");

             return;

            }

        else

           {

            // Nếu không thì thêm media vào cart và thông báo thêm thành công

            itemOrdered.add(item1);

            itemOrdered.add(item2);

            System.out.println("Both items have been added");

           }

    }

    public void removeMedia (Media item)

    {       // Method cho việc loại bỏ media

            if (itemOrdered.isEmpty())

            {   //Nếu list trống thì thông báo cart trống

                System.out.println("The cart is empty");

                return;

            }

            if (itemOrdered.remove(item))

            {   // Nếu loại bỏ thành công thì gửi thông báo

                System.out.println("The disc is removed successfully");

            }

            else

            {   // Nếu không loại bỏ thành công thì gửi thông báo disc không có trong cart

                System.out.println("The disc is not in the cart");

            }

    }

    public float totalCost ()

    {   //Method tính tổng cost

        float total =0;

        for (Media item : itemOrdered) {

             total += item.getCost();

        }

        // Duyệt tuần tự qua cart và tính tổng cost

        return total;

    }

    public void showCart ()

    {   // Method để liệt kê những gì có trong cart và chi phí của chúng

        int i = 1;

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CART\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

        for (Media item : itemOrdered) {

             System.out.println(i + ".item - " + item.toString());

             i++;

        // Duyệt lần lượt qua cart và in kết quả

        }

        System.out.print("Total Cost: ");

        System.out.println(totalCost());

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    }

    public Media searchByTitle(String Title)

    { // Method để tìm bằng title

        int count = 1;

        Media result = null;

        for (Media item : itemOrdered) {

            if (item.isMatch(Title))

                {// Duyệt lần lượt qua cart và kiểm tra xem title của dvd có chứa xâu dùng để tra không

                System.out.println(count + ".item -" + item.toString());

                result = item;

                count ++;

                // nếu có thì in ra và tăng count

                }

       }

        if (count == 1)

        {   //Nếu count không đổi thì in ra không tìm được

            System.out.println("No results found");

            return null;

            //trả về null khi không tìm thấy

        }

        return result;

        //trả về kết quả cuối

    }

    public void searchById(int Id)

    {   int count = 1;

        for (Media item : itemOrdered) {

            if (Id == item.getId())

                {// Duyệt tuần tự qua cart và kiểm tra xem id có trùng không

                System.out.println(count + ".item -" + item.toString());

                count ++;

                //Nếu có thì in ra và tăng count rồi break vì id là duy nhất

                break;

                }

       }

        if (count == 1)

        {   // nếu count không đổi thì in ra là không tìm được

            System.out.println("No results found");

        }

    }

    public void sortByTitle()

    {

        java.util.Collections.sort(itemOrdered,Media.COMPARE\_BY\_TITLE\_COST);

    }

    public void sortByCost()

    {

        java.util.Collections.sort(itemOrdered, Media.COMPARE\_BY\_COST\_TITLE);

    }

}

# ***hust.soict.hedspi.aims.media***

1. *Media.java*

package hust.soict.hedspi.aims.media;

import java.util.Comparator;

public abstract class Media {

    private int id;

    private String title;

    private String category;

    private float cost;

    //Khởi tạo các thuộc tính của Media

    public static final Comparator<Media> COMPARE\_BY\_TITLE\_COST =

            new MediaComparatorByTitleCost();

    public static final Comparator<Media> COMPARE\_BY\_COST\_TITLE =

            new MediaComparatorByCostTitle();

    //Các thuộc tính comparator

    public Media(String title) {

        super();

        this.title = title;

    }

    public Media(String title, String category, float cost) {

        super();

        this.title = title;

        this.category = category;

        this.cost = cost;

    }

    //Các constructor

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public String getCategory() {

        return category;

    }

    public void setCategory(String category) {

        this.category = category;

    }

    public float getCost() {

        return cost;

    }

    public void setCost(float cost) {

        this.cost = cost;

    }

    // Các getters và setters

    public boolean equals(Media o)

    {   //override equals, nếu cùng title thì trả về true

        if (title.equals(o.title))

        {

            return true;

        }

        else

        {

            return false;

        }

    }

    public boolean isMatch(String tit) {

        //Method để kiểm tra trong title có chứa xâu tit không

            return getTitle().contains(tit);

        }

    public Media() {

    }

}

1. *Book.java*

package hust.soict.hedspi.aims.media;

import java.util.ArrayList;

import java.util.List;

public class Book extends Media {

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

    public Book(String title) {

        super(title);

    }

    public Book(String title, String category, float cost) {

        super(title,category,cost);

    }

    //Các constructor cho class

    public void addAuthor(String authorName)

    {   // Method cho việc thêm authorName

        if ( authors.contains(authorName))

            { //Kiểm tra xem tên tác giả có trong list chưa

             System.out.println("The author name is already presented");

             return;

            }

        else

           {

            // Nếu không thì thêm vào và thông báo thêm thành công

            authors.add(authorName);

            System.out.println("Author's name has been added successfully");

           }

    }

    public void removeAuthor(String authorName)

    {   // Method cho việc xoa authorName

        if ( authors.contains(authorName))

            { //Kiểm tra xem tên tác giả có trong list chưa

             authors.remove(authorName);

             System.out.println("The author's name has been removed successfully");

             return;

            }

        else

           {

            // Nếu không thì báo lỗi

            System.out.println("This author's name is not presented, can't remove it");

           }

    }

    public String toString()

        {// Phương thức chuyển các thông tin của Book thành string rồi trả lại

            String cd;

            cd = getTitle() + " - " + getCategory() + " - "  + getCost();

            return cd;

        }

    public Book() {

    }

}

1. *Disc.java*

package hust.soict.hedspi.aims.media;

public class Disc extends Media {

    private String director;

    private int length;

    //Các thuộc tính của Disc

    public String getDirector() {

        return director;

    }

    public int getLength() {

        return length;

    }

    //Các getter

    public void setDirector(String director) {

        this.director = director;

    }

    public void setLength(int length) {

        this.length = length;

    }

    //Các setter

    public Disc(String title) {

        super(title);

    }

    public Disc(String title, String category, float cost) {

        super(title,category,cost);

    }

    public Disc(String director, int length) {

        super();

        this.director = director;

        this.length = length;

    }

    public Disc() {

        super();

    }

    //Các constructor

}

1. *Track.java*

package hust.soict.hedspi.aims.media;

public class Track implements Playable {

    private String title;

    private int length;

    //Các thuộc tính của track

    public String getTitle() {

        return title;

    }

    public int getLength() {

        return length;

    }

    //Các getter

    public Track(String title, int length) {

        super();

        this.title = title;

        this.length = length;

    }

    //Constructor cho track

    public boolean equals(Track o)

    {//Override equals, nếu cùng title và length thì trả về true

        if (title.equals(o.title))

        {

            if (length == o.length )

            {

                return true;

            }

            else

            {

                return false;

            }

        }

        else

        {

            return false;

        }

    }

       public String toString()

        {// Phương thức chuyển các thông tin của DVD thành string rồi trả lại

            String track;

            track = getTitle() + " - " +getLength() ;

            return track;

        }

    public void play() {

        //Method thực hiện interface playable

        System.out.println("Playing DVD: " + this.getTitle());

        System.out.println("DVD length: " + this.getLength());

        }

}

1. *CompactDisc.java*

package hust.soict.hedspi.aims.media;

import java.util.ArrayList;

import java.util.List;

public class CompactDisc extends Disc implements Playable {

    private String artist;

    private List<Track> tracks = new ArrayList<Track>();

    //khởi tạo các thuộc tính của dvd

    public CompactDisc(String title) {

        super(title);

    }

    public CompactDisc(String title, String category, float cost) {

        super(title,category,cost);

    }

    public CompactDisc(String director, int length, String artist) {

        super(director, length);

        this.artist = artist;

    }

    //Các constructor cho CD

    public String getArtist() {

        return artist;

    }

    // Getter để lấy Artist

    public void addTrack(Track trackName)

    {   // Method cho việc thêm track

        if ( tracks.contains(trackName))

            { //Kiểm tra xem track có trong list chưa

             System.out.println("The track is already presented");

             return;

            }

        else

           {

            // Nếu không thì thêm vào và thông báo thêm thành công

            tracks.add(trackName);

            System.out.println("Track has been added successfully");

           }

    }

    public void removeTrack(Track trackName)

    {   // Method cho việc xoa track

        if ( tracks.contains(trackName))

            { //Kiểm tra xem tên có trong list chưa

             tracks.remove(trackName);

             System.out.println("The track has been removed successfully");

             return;

            }

        else

           {

            // Nếu không thì báo

            System.out.println("This track is not presented, can't remove it");

           }

    }

    public int getLength ()

    {   //Method tính length

        int total =0;

        for (Track track : tracks) {

             total += track.getLength();

        }

        // Duyệt tuần tự qua tracks và tính tổng thời gian

        return total;

    }

    public void play()

    {   //Method để play

        System.out.println("Total track: " + tracks.size());

        System.out.println("Total runtime " + getLength());

        for (Track track : tracks) {

             track.play();

        }

        // Duyệt tuần tự qua tracks và play

        return;

    }

       public String toString()

        {// Phương thức chuyển các thông tin của CD thành string rồi trả lại

            String cd;

            cd =    getTitle()+ " - " + getCategory() + " - " + getDirector()+ " - "  +getArtist() + " - "  +getLength();

            return cd;

        }

}

1. *DigitalVideoDisc.java*

package hust.soict.hedspi.aims.media;

public class DigitalVideoDisc extends Disc implements Playable {

    private static int nbDigitalVideoDiscs = 0;

    // Khởi tạo thuộc tính của DigitalVideoDisc

    public DigitalVideoDisc(String title) {

        super(title);

        setId(++nbDigitalVideoDiscs);

    }

    public DigitalVideoDisc(String title, String category, float cost) {

        super(title, category, cost);

        setId(++nbDigitalVideoDiscs);

    }

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

        super(title, category, cost);

        setDirector(director);

        setId(++nbDigitalVideoDiscs);

    }

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

        super(title, category, cost);

        setDirector(director);

        setLength(length);

        setId(++nbDigitalVideoDiscs);

    }

    //Các phương thức constructor cho DVD, dùng super tới Media

    public String toString() {

        // Phương thức chuyển các thông tin của DVD thành string rồi trả lại

        String dvd = getTitle() + " - " + getCategory() + " - " +

                getDirector() + " - "  + getLength() + " - " + getCost();

        return dvd;

    }

    public void play() {

        //Method play cho interface playable

        System.out.println("Playing DVD: " + getTitle());

        System.out.println("DVD length: " + getLength());

    }

}

1. *MediaComparatorByCostTitle.java*

package hust.soict.hedspi.aims.media;

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> {

    //Class Comparator theo cost rồi đến title

    private Comparator<Media> costComparator = Comparator.comparingDouble(Media::getCost);

    private Comparator<Media> titleComparator = Comparator.comparing(Media::getTitle);

    // Dùng hàm comparing theo cost và title

    private Comparator<Media> compoundComparator = costComparator.thenComparing(titleComparator);

    //Dùng thenComparing để compare theo cost rồi đến title

    public int compare(Media media1, Media media2) {

        return compoundComparator.compare(media1, media2);

    }

}

1. *MediaComparatorByTitleCost.java*

package hust.soict.hedspi.aims.media;

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {

  //Class Comparator theo title rồi đến cost

  private Comparator<Media> costComparator = Comparator.comparingDouble(Media::getCost);

    private Comparator<Media> titleComparator = Comparator.comparing(Media::getTitle);

 // Dùng hàm comparing theo cost và title

    private Comparator<Media> compoundComparator = titleComparator.thenComparing(costComparator);

  //Dùng thenComparing để compare theo cost rồi đến title

    public int compare(Media media1, Media media2) {

        return compoundComparator.compare(media1, media2);

    }

}

1. *Playable.java*

package hust.soict.hedspi.aims.media;

public interface Playable {

    public void play();

    //interface Playable

}

* 1. ***hust.soict.hedspi.test:*** *ToStringTest.java*

package hust.soict.hedspi.test;

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

import java.util.ArrayList;

import java.util.List;

public class ToStringTest {

    public static void main(String[] args) {

        List<Media> mediae = new ArrayList<Media>();

        CompactDisc  cd = new CompactDisc("Michael Bay",8,"Michael Jackson");

        DigitalVideoDisc dvd = new DigitalVideoDisc("The Lion King",

                "Animation", "Roger Allers", 87, 19.95f );

        Book book = new Book("The Gutenberg Galaxy","Literary analysis", 5.6f);

        //Khởi tạo các media khác nhau

        mediae.add(cd);

        mediae.add(dvd);

        mediae.add(book);

        //Thêm chúng vào list

        for(Media m: mediae) {

            System.out.println(m.toString());

            //Duyệt tuần tự và thử chúng với toString

        }

        //Test kiểm tra tính đa hình với hàm toString

    }}

# **Demo chương trình**

* 1. ToStringTest

A screenshot of a computer

Description automatically generated

* 1. AIMS

1. showMenu()

A screenshot of a computer

Description automatically generated

1. View store

A screenshot of a computer

Description automatically generated

1. See media’s detail:

A screenshot of a computer

Description automatically generated

A white screen with black text

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. Add to cart

A screenshot of a computer program

Description automatically generated

1. Play

A screenshot of a computer program

Description automatically generated

1. Add a medium to cart

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Play a medium

A black text on a white background

Description automatically generated

A screen shot of a computer

Description automatically generated

1. See current cart

A screenshot of a computer

Description automatically generated

1. Filter

A white background with black text

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Sort media

A screenshot of a computer program

Description automatically generated

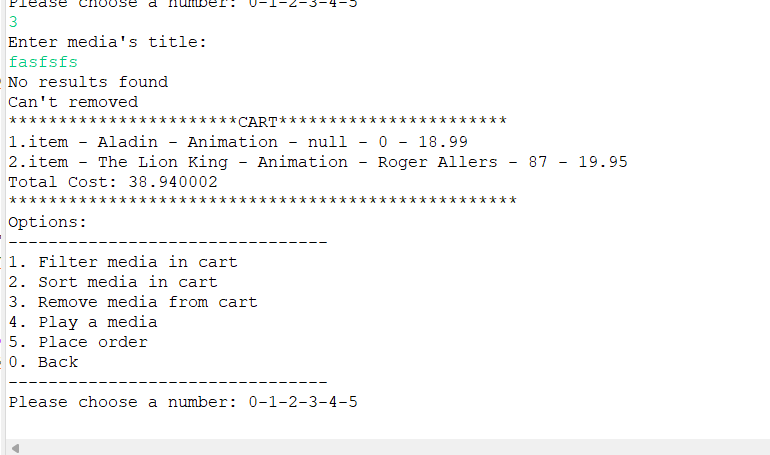
A screen shot of a computer

Description automatically generated

1. Remove media from cart

A screenshot of a computer

Description automatically generated



1. Play a media

A screenshot of a computer

Description automatically generated

1. Place order

A screenshot of a computer

Description automatically generated

1. Update store

A white paper with black text

Description automatically generated

A white background with black text

Description automatically generated

A white background with black text

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated