Họ và tên: Nguyễn Hồng Quân

Mã số sinh viên: 20225908

Lớp: 744521 – Lập trình hướng đối tượng – IT3103

**LAP 04: Inheritance and Polymorphism**

I. Import the existing project into the workspace of Eclipse

II. Additional requirements of AIMS

III. Creating the Book class

package hust.soict.dsai.aims.media;

import java.util.ArrayList;

import java.util.List;

public class Book {

    private int id;

    private String title;

    private String category;

    private float cost;

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

    public Book() {

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    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;

    }

    public List<String> getAuthors() {

        return authors;

    }

    public void setAuthors(List<String> authors) {

        this.authors = authors;

    }

    public void addAuthor(String authorName) {

        for(String author: authors) {

            if(authorName.equals(author)) {

                System.out.println("Author Name da ton tai!!");

                return;

            }

        }

        authors.add(authorName);

    }

    public void removeAuthor(String authorName) {

        for(String author: authors) {

            if(authorName.equals(author)) {

                authors.remove(author);

                System.out.println("Author" + author + " da duoc xoa khoi danh sach");

                return;

            }

        }

        System.out.println("Author không tồn tại trong danh sách!");

    }

}

IV. Creating the abstract Media class

* Media class

package hust.soict.dsai.aims.media;

public abstract class Media{

    private int id;

    private String title;

    private String category;

    private float cost;

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    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;

    }

}

* Book class

package hust.soict.dsai.aims.media;

import java.util.ArrayList;

import java.util.List;

public class Book extends Media {

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

    public Book() {

    }

    public List<String> getAuthors() {

        return authors;

    }

    public void setAuthors(List<String> authors) {

        this.authors = authors;

    }

    public void addAuthor(String authorName) {

        for(String author: authors) {

            if(authorName.equals(author)) {

                System.out.println("Author Name da ton tai!!");

                return;

            }

        }

        authors.add(authorName);

    }

    public void removeAuthor(String authorName) {

        for(String author: authors) {

            if(authorName.equals(author)) {

                authors.remove(author);

                System.out.println("Author" + author + " da duoc xoa khoi danh sach");

                return;

            }

        }

        System.out.println("Author không tồn tại trong danh sách!");

    }

}

* DigitalVideoDisc class

package hust.soict.dsai.aims.media;

public class DigitalVideoDisc extends Media {

    private static int nbDigitalVideoDiscs = 0;

    private String director;

    private int length;

    public DigitalVideoDisc() {

    }

    public String getDirector() {

        return director;

    }

    public void setDirector(String director) {

        this.director = director;

    }

    public int getLength() {

        return length;

    }

    public void setLength(int length) {

        this.length = length;

    }

}

V. Creating the CompactDisc class

5.1. Create the Disc class extending the Media class

* Disc class

package hust.soict.dsai.aims.media;

public abstract class Disc extends Media {

    private int legth;

    private String director;

    public Disc() {

        super();

        // TODO Auto-generated constructor stub

    }

    public int getLegth() {

        return legth;

    }

    public void setLegth(int legth) {

        this.legth = legth;

    }

    public String getDirector() {

        return director;

    }

    public void setDirector(String director) {

        this.director = director;

    }

}

* CompactDisc class

package hust.soict.dsai.aims.media;

public abstract class CompactDisc extends Disc {

}

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

* Track class

package hust.soict.dsai.aims.media;

public class Track {

    private String title;

    private int legth;

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public int getLegth() {

        return legth;

    }

    public void setLegth(int legth) {

        this.legth = legth;

    }

    public Track(String title, int legth) {

        super();

        this.title = title;

        this.legth = legth;

    }

}

5.3. Open the CompactDisc class

package hust.soict.dsai.aims.media;

import java.util.ArrayList;

import java.util.List;

public abstract class CompactDisc extends Disc {

    private String artist;

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

    // Getter và Setter cho artist

    public String getArtist() {

        return artist;

    }

    public void setArtist(String artist) {

        this.artist = artist;

    }

    // Phương thức thêm track

    public void addTrack(Track track) {

        for (Track t : tracks) {

            if (track.getTitle().equals(t.getTitle())) {

                System.out.println("Track \"" + track.getTitle() + "\" đã tồn tại!");

                return;

            }

        }

        tracks.add(track);

        System.out.println("Track \"" + track.getTitle() + "\" đã được thêm thành công!");

    }

    // Phương thức xóa track

    public void removeTrack(Track track) {

        for (Track t : tracks) {

            if (track.getTitle().equals(t.getTitle())) {

                tracks.remove(t);

                System.out.println("Track \"" + track.getTitle() + "\" đã được xóa khỏi danh sách!");

                return;

            }

        }

        System.out.println("Track \"" + track.getTitle() + "\" không tồn tại trong danh sách!");

    }

    // Constructor mặc định

    public CompactDisc() {

        super();

    }

}

VI. Create the Playable interface

* Tạo interface Playable

package hust.soict.dsai.aims.media;

public interface Playable {

    public void play();

}

* Đối với class CompactDisc, thêm implements Playable sau extends Disc và thêm phương thức play()
* Thêm implenments Playable sau extends Disc:

public abstract class CompactDisc extends Disc implements Playable {

    private String artist;

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

* Thêm phương thức play()

public void play() {

    System.out.println("Now playing CD: " + getTitle());

    System.out.println("Artist: " + getArtist());

    System.out.println("Track list:");

    int totalLength = 0;

    for (Track track : tracks) {

        track.play();

        totalLength += track.getLength();

    }

    System.out.println("Total CD length: " + totalLength + " minutes");

}

* Đối với class DigitalVideoDisc, thêm implements Playable sau extends Disc và thêm phương thức play()
* Thêm implements Playable sau extends Disc

public class DigitalVideoDisc extends Disc implements Playable {

* Thêm phương thức play():

public void play() {

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

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

}

* Đối với class Track, chèn các từ khóa implements Playable sua từ khóa public class Track, rồi thêm phương thức play() cho Track
* Thêm từ khóa implements Playable

public class Track implements Playable {

    private String title;

    private int length;

    public String getTitle() {

        return title;

    }

* Thêm phương thức play()

public void play() {

    // TODO Auto-generated method stub

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

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

}

VII. Update the Cart class to work with Media

* Xóa mảng itemsOrdered, cũng như các phương thức add và remove của nó
* Tạo lại trường itemsOrdered, lần này là một đối tượng ArrayList thay vì một mảng
* Tạo addMedia() và removeMedia() để thay thế addDigitalVideoDisc() và remove DigitalVideoDisc()
* Cập nhật phương thức totalCost()

package hust.soict.dsai.aims.cart;

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

import java.util.ArrayList;

public class Cart {

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

    // Updated itemsOrdered to ArrayList of Media

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

    // Method to add a Media item to the cart

    public void addMedia(Media media) {

        if (itemsOrdered.size() >= MAX\_NUMBERS\_ORDERED) {

            System.out.println("The cart is full. Can't add more items.");

        } else {

            itemsOrdered.add(media);

            System.out.println("The item \"" + media.getTitle() + "\" has been added!");

        }

    }

    // Method to remove a Media item from the cart

    public void removeMedia(Media media) {

        if (itemsOrdered.isEmpty()) {

            System.out.println("Your cart is empty!");

        } else if (itemsOrdered.remove(media)) {

            System.out.println("The item \"" + media.getTitle() + "\" has been removed successfully!");

        } else {

            System.out.println("No matching item found!");

        }

    }

    // Method to calculate the total cost of items in the cart

    public float totalCost() {

        float sum = 0.0f;

        for (Media media : itemsOrdered) {

            sum += media.getCost();

        }

        return sum;

    }

    // Method to print the cart details

    public void print() {

        StringBuilder output = new StringBuilder("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CART\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nOrdered items:\n");

        for (int i = 0; i < itemsOrdered.size(); i++) {

            Media media = itemsOrdered.get(i);

            output.append(i + 1).append(". ").append(media.toString()).append("\n");

        }

        output.append("Total: ").append(totalCost()).append(" $\n");

        output.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

        System.out.println(output);

    }

    // Method to search an item by title

    public void searchByTitle(String title) {

        boolean found = false;

        for (Media media : itemsOrdered) {

            if (media.getTitle().equalsIgnoreCase(title)) {

                System.out.println("Result: " + media.toString());

                found = true;

                break;

            }

        }

        if (!found) {

            System.out.println("No match found!");

        }

    }

}

VIII. Update the Store class to work with Media

* Tương tự như lớp Cart, thay đổi thuộc tính itemsInStore[] của lớp Store thành kiểu ArrayList.
* Thay thế phương thức addDigitalVideoDisc() và removeDigitalVideoDisc() bằng addMedia() và removeMedia()

package hust.soict.dsai.aims.store;

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

import java.util.ArrayList;

public class Store {

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

    private boolean checkMedia(Media media) {

        return itemsInStore.contains(media);

    }

    public void addMedia(Media media) {

        if (!checkMedia(media)) {

            itemsInStore.add(media);

            System.out.println(media.getTitle() + " has been added to the store!");

        } else {

            System.out.println(media.getTitle() + " already exists in the store!");

        }

    }

    public void removeMedia(Media media) {

        if (checkMedia(media)) {

            itemsInStore.remove(media);

            System.out.println(media.getTitle() + " has been deleted from the store!");

        } else {

            System.out.println("There is no " + media.getTitle() + " in the store!");

        }

    }

    @Override

    public String toString() {

        StringBuilder string = new StringBuilder("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*STORE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nItems in the store:\n");

        if (itemsInStore.isEmpty()) {

            string.append("There are no items in the store!\n");

        } else {

            for (Media media : itemsInStore) {

                string.append(media.getTitle()).append(" - ").append(media.getCost()).append(" $\n");

            }

        }

        string.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

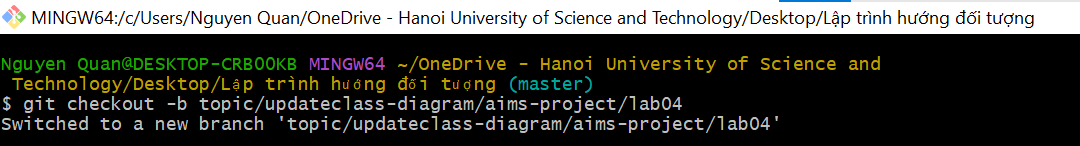
        return string.toString();

    }

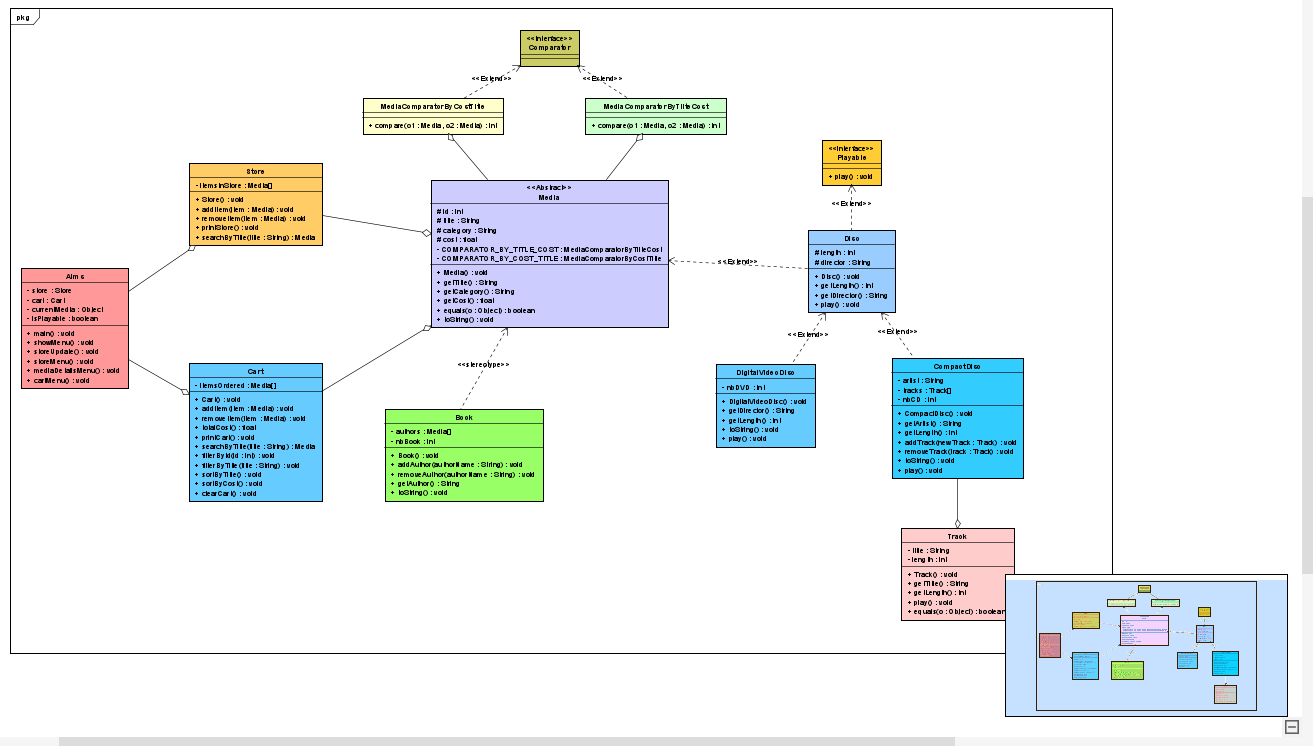
}

IX. Constructors of whole classes and parent classes

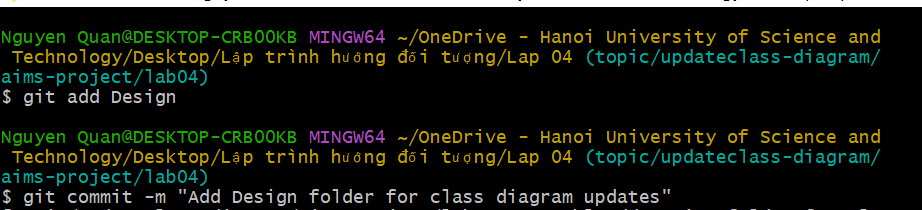
* Tạo branch topic/updateclass-diagram/aims-project/lab04



* Update diagram



* Đẩy lên branch và merge với master



* Which classes are aggregates of other classes? Checking all constructors of whole classes if they initialize for their parts?

Lớp media là lớp cha của lớp Store, Cart, Book, Disc

Lớp CompactDics,DigitalVideoDisc là con của lớp Disc

Viết constructors cho các lớp cha và lớp con. Xóa các phương thức setter dư thừa nếu có

package hust.soict.dsai.aims.media;

public abstract class Media{

    private int id;

    private String title;

    private String category;

    private float cost;

    public Media() {

    }

    public Media(String title) {

        this.title = title;

    }

    public Media(String title, float cost) {

        this(title);

        this.cost = cost;

    }

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

        this(title, cost);

        this.category = category;

    }

    public int getId() {

        return id;

    }

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public String getCategory() {

        return category;

    }

    public float getCost() {

        return cost;

    }

}

X. Unique item in a list

- Đối với lớp Media: tiêu đề là bằng nhau

Thêm hàm equals vào trong file media.java

public boolean equals(Object o) {

    if (this == o) return true;

    if (o == null || getClass() != o.getClass()) return false;

    Media media = (Media) o;

    return this.title != null && this.title.equals(media.title);

}

- Đối với lớp Track: tiêu đề và độ dài bằng nhau

Thêm hàm equals và trong class Track.java

public boolean equals(Object o) {

    Track track = (Track) o;

    return this.title.equals(track.title) && this.length == track.length;

}

**Câu hỏi:** Khi ghi đè phương thức equals() của lớp Object, bạn sẽ phải ép kiểu tham số Object obj thành kiểu Object mà bạn đang xử lý. Ví dụ, trong lớp Media, bạn phải ép kiểu Object obj thành một Media, và sau đó kiểm tra sự bằng nhau của hai thuộc tính của hai đối tượng theo yêu cầu như trên (nghĩa là tiêu đề cho Media; tiêu đề và độ dài cho Track). Nếu đối tượng truyền vào không phải là một thể hiện của Media, điều gì sẽ xảy ra?

Khi ghi đè phương thức equals() của lớp Object, bạn sẽ cần ép kiểu tham số Object obj thành kiểu của Object mà bạn đang xử lý. Ví dụ, trong lớp Media, bạn phải ép kiểu Object obj thành một đối tượng Media, sau đó kiểm tra tính bằng nhau của hai thuộc tính của hai đối tượng theo yêu cầu như trên (nghĩa là tiêu đề cho Media; tiêu đề và độ dài cho Track). Nếu đối tượng truyền vào không phải là một thể hiện của lớp Media, điều gì sẽ xảy ra?

Nếu đối tượng truyền vào không phải là một thể hiện của lớp Media, việc ép kiểu sẽ gây ra một ngoại lệ kiểu (ClassCastException). Điều này xảy ra khi bạn cố gắng ép kiểu một đối tượng sang một kiểu mà nó không thể chuyển đổi được.

Để tránh ngoại lệ này, trước khi thực hiện ép kiểu, bạn có thể kiểm tra xem đối tượng có phải là một thể hiện của lớp Media không bằng cách sử dụng toán tử instanceof

Ví dụ:

========================================================

if (obj instanceof Media) {

Media otherMedia = (Media) obj;

// Kiểm tra tính bằng nhau của các thuộc tính

// ...

} else {

// Xử lý trường hợp không phải là thể hiện của lớp Media

}

========================================================

Sử dụng instanceof giúp bạn kiểm tra trước khi thực hiện ép kiểu, giảm nguy cơ ngoại lệ kiểu khi xử lý đối tượng không mong muốn.

XI. Polymorphism with toString() method

Khởi tạo tham số CD:

Title : To pimp a butterfly

Category: Rap

Cost: 19.99f

Director: kendrick lamar

Artist: KL

Khởi tạo tham số DVD:

Title : The Lion King

Category: Animation

Cost: 19.95f

Director: Roger Allers

Length: 98

Khởi tạo tham số Book:

Title: A song of ice and fire

Category: Novel

Cost: 42.34f

Hàm toString của CD:

@Override

     public String toString() {

        return "CD(" + this.id + "): " + this.title + " - " + this.category + " - " + this.director + " - " + this.artist + " - " + this.getLength() + " - " + this.cost + "$";

    }

Hàm toString của DVD:

public String toString() {

    return "DVD(" + this.id + "): " + this.title + " - " + this.category + " - " + this.director + " - "

            + this.length + " - " + this.cost + "$";

}

Hàm toString của Book

public String toString() {

    return "Book(" + this.id + "): " + this.title + " - " + this.getAuthor() + " - " + this.category + " - "

            + this.cost + "$";

}

Hàm main

package hust.soict.dsai.aims;

import hust.soict.dsai.aims.media.CompactDisc;

import hust.soict.dsai.aims.media.DigitalVideoDisc;

import hust.soict.dsai.aims.media.book;

import hust.soict.dsai.aims.media.media;

import java.util.ArrayList;

import java.util.List;

public class Aims {

    public static void main(String[] args) throws Exception {

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

        CompactDisc cd = new CompactDisc("To pimp a butterfly ", "Rap", 19.99f, "KL", "kendrick lamar");

        DigitalVideoDisc dvd = new DigitalVideoDisc("The Lion King", "Animation", 19.95f, "Roger Allers", 98);

        book book1 = new book("A song of ice and fire", "Novel", 42.34f);

        mediae.add(cd);

        mediae.add(dvd);

        mediae.add(book1);

        for (media m : mediae) {

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

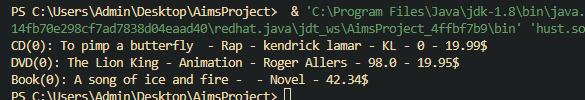
        }

        System.exit(0);

    }

}

Chạy chương trình



XII. Sort media in the cart

* Thêm vào media.java

public static final Comparator<Media> COMPARE\_BY\_TITLE\_COST = new MediaComparatorByTitleCost();

public static final Comparator<Media> COMPARE\_BY\_COST\_TITLE = new MediaComparatorByCostTitle();

* Tạo file MediaComparatorByCostTitle.java

package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<media> {

    @Override

    public int compare(media o1, media o2) {

        if(o1.getCost() - o2.getCost() > 0 ) {

            return -1;

        }else if(o1.getCost() - o2.getCost() < 0){

            return 1;

        }else {

            return o1.getTitle().compareTo(o2.getTitle());

        }

    }

}

* Tạo file MediaComparatorByTitleCost.java

package hust.soict.dsai.aims.media;

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<media>{

    @Override

    public int compare(media o1, media o2) {

        if(o1.getTitle().compareTo(o2.getTitle()) > 0 ) {

            return 1;

        }else if(o1.getTitle().compareTo(o2.getTitle()) < 0){

            return -1;

        }else {

            if(o1.getCost() - o2.getCost() < 0 ) {

                return 1;

            }else if(o1.getCost() - o2.getCost() > 0){

                return -1;

            }else {

                return 0;

            }

        }

    }

}

* Thêm hàm sortByTitle() và sortByCost() vào file Cart.java

public void sortByTitle() {

        Collections.sort(itemsOrdered, Media.COMPARE\_BY\_TITLE\_COST);

        print();

    }

    public void sortByCost() {

        Collections.sort(itemsOrdered, Media.COMPARE\_BY\_COST\_TITLE);

        print();

    }

Hàm main

package hust.soict.dsai.aims;

import hust.soict.dsai.aims.cart.Cart;

import hust.soict.dsai.aims.media.CompactDisc;

import hust.soict.dsai.aims.media.DigitalVideoDisc;

import hust.soict.dsai.aims.media.Track;

import hust.soict.dsai.aims.media.Book;

public class Aims {

    public static void main(String[] args) throws Exception {

        Cart cart1 = new Cart();

        CompactDisc cd = new CompactDisc("To pimp a butterfly", "Rap", 19.99f, "KL", "kendrick lamar");

        DigitalVideoDisc dvd = new DigitalVideoDisc("John wick 4", "Action", 25.46f, "Chad Stahelski", 36);

        Track track1 = new Track("Demons", 28);

        Track track2 = new Track("Angels", 44);

        cd.addTrack(track1, track2);

        Book book1 = new Book("A song of ice and fire", "Novel", 42.34f);

        Book book2 = new Book("To kill a mockingbird", "Novel", 3.26f);

        book1.addAuthor("JRR Martin");

        book2.addAuthor("Harper Lee");

        cart1.addMedia(cd);

        cart1.addMedia(dvd);

        cart1.addMedia(book1);

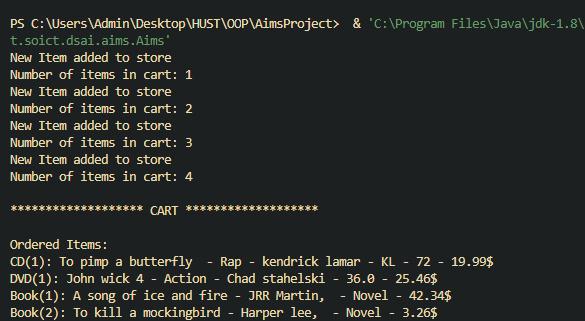
        cart1.addMedia(book2);

        cart1.print();

    }

}

Chạy chương trình:



Bây giờ chúng ta sẽ tiếp tục chạy thử hàm SortByCost trong Cart

B1:Thêm câu lệnh sau vào hàm main:

cart1.sortByCost();

B2: Chạy thử chương trình

Ta thấy được List đã được sắp xếp lại theo Cost từ lớn đến bé

Question:

Lớp nào nên thực hiện giao diện Comparable?

Trong trường hợp này, lớp Media nên thực hiện giao diện Comparable để định nghĩa một thứ tự tự nhiên cho các thể hiện của lớp Media.

Trong những lớp đó, phải thực hiện phương thức compareTo() như thế nào để phản ánh thứ tự mà chúng ta muốn?

Phương thức compareTo() trong lớp Media nên được thực hiện để kiểm soát thứ tự sắp xếp theo yêu cầu. Chúng ta có thể sử dụng các điều kiện để xác định thứ tự dựa trên yêu cầu sắp xếp hiện tại (tiêu đề sau đó giá, hoặc là giá sau đó tiêu đề).

Chúng ta có thể có hai quy tắc sắp xếp của mục (theo tiêu đề sau đó giá và theo giá sau đó tiêu đề) nếu chúng ta sử dụng giao diện Comparable không?

Không, giao diện Comparable chỉ cho phép một quy tắc sắp xếp tự nhiên cho lớp đó. Điều này có nghĩa là chúng ta chỉ có thể có một cách sắp xếp mặc định cho đối tượng của lớp Media khi sử dụng giao diện Comparable.

Để có hai quy tắc sắp xếp khác nhau (theo tiêu đề sau đó giá và theo giá sau đó tiêu đề), chúng ta sẽ cần sử dụng Comparator với phương thức thenComparing() để có thể xác định nhiều quy tắc sắp xếp khác nhau cho đối tượng Media.y tắc sắp xếp khác nhau cho đối tượng Media.

XIII. Create a complete console application in the Aims class

Hàm main

package hust.soict.dsai.aims;

import java.util.Scanner;

import hust.soict.dsai.aims.cart.cart;

import hust.soict.dsai.aims.media.CompactDisc;

import hust.soict.dsai.aims.media.DigitalVideoDisc;

import hust.soict.dsai.aims.media.Disc;

import hust.soict.dsai.aims.media.Track;

import hust.soict.dsai.aims.media.book;

import hust.soict.dsai.aims.media.media;

import hust.soict.dsai.aims.store.Store;

public class Aims {

    static Store store = new Store();

    static cart cart = new cart();

    static Object currentMedia = new Object();

    static boolean isPlayable = false;

    public static void main(String args[]) {

        CompactDisc cd  = new CompactDisc("To pimp a butterfly ","Rap",19.99f,"KL","kendrick lamar");

        DigitalVideoDisc dvd = new DigitalVideoDisc("John wick 4", "Action", 25.46f,"Chad stahelski",36);

        Track track1 = new Track(28,"Demons");

        Track track2 = new Track(44,"Angels");

        cd.addTrack(track1, track2);

        book book1 = new book("A song of ice and fire","Novel",42.34f);

        book book2 = new book("To kill a mockingbird","Novel",3.26f);

        book1.addAuthor("JRR Martin");

        book2.addAuthor("Harper lee","hello");

        store.addItem(cd, dvd, book1, book2);

        showMenu();

    }

Hàm showmenu()

public static void showMenu() {

    for (; ; ) {

        System.out.println("\nAIMS: ");

        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");

        Scanner keyboard = new Scanner(System.in);

        System.out.print("Your choice: ");

        int choice = keyboard.nextInt();

        if (choice == 1) {

            store.printStore();

            storeMenu();

        } else if (choice == 2) {

            storeUpdate();

        } else if (choice == 3) {

            cart.printCart();

            cartMenu();

        } else if (choice == 0) {

            break;

        } else {

            System.out.println("Invalid choice please choose again.");

        }

    }

    System.out.println("Thank you for using our service!");

}

Hàm storeUpdate:

public static void storeUpdate() {

    for (; ; ) {

        store.printStore();

        System.out.println("\nOptions: ");

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

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

        Scanner keyboard = new Scanner(System.in);

        System.out.print("Your choice: ");

        int choice = keyboard.nextInt();

        if (choice == 1) {

            int x;

            for (; ; ) {

                System.out.println("Type of media to add");

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

                System.out.println("1. Book");

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

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

                System.out.println("0. Back");

                keyboard = new Scanner(System.in);

                System.out.print("Your choice: ");

                x = keyboard.nextInt();

                if (x == 0) {

                    break;

                }

                if (x > 3 || x < 0) {

                    System.out.println("Invalid type of media.\n");

                    continue;

                }

                break;

            }

            if (x == 0) {

                break;

            }

            keyboard = new Scanner(System.in);

            System.out.print("Enter title: ");

            String title = keyboard.nextLine();

            keyboard = new Scanner(System.in);

            System.out.print("Enter cost: ");

            float cost = keyboard.nextFloat();

            keyboard = new Scanner(System.in);

            System.out.print("Enter category: ");

            String category = keyboard.nextLine();

            if (x == 1) {

                book book = new book(title, category, cost);

                System.out.println("Input '\*' when add all authors.");

                for (; ; ) {

                    keyboard = new Scanner(System.in);

                    System.out.print("Enter author: ");

                    String author = keyboard.nextLine();

                    if (author.equals("\*")) {

                        break;

                    }

                    book.addAuthor(author);

                }

                store.addItem(book);

            }

            if (x == 2) {

                keyboard = new Scanner(System.in);

                System.out.print("Enter director: ");

                String director = keyboard.nextLine();

                keyboard = new Scanner(System.in);

                System.out.print("Enter length: ");

                int length = keyboard.nextInt();

                DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, cost, director, length);

                store.addItem(dvd);

            }

            if (x == 3) {

                keyboard = new Scanner(System.in);

                System.out.print("Enter director: ");

                String director = keyboard.nextLine();

                keyboard = new Scanner(System.in);

                System.out.print("Enter artist: ");

                String artist = keyboard.nextLine();

                CompactDisc cd = new CompactDisc(title, category, cost, artist, director);

                System.out.println("Input '\*' when add all tracks.");

                for (; ; ) {

                    keyboard = new Scanner(System.in);

                    System.out.print("Enter track's title: ");

                    String trackTitle = keyboard.nextLine();

                    if (trackTitle.equals("\*")) {

                        break;

                    }

                    keyboard = new Scanner(System.in);

                    System.out.print("Enter track's length: ");

                    int trackLength = keyboard.nextInt();

                    Track track = new Track(trackLength, trackTitle);

                    cd.addTrack(track);

                }

                store.addItem(cd);

            }

        } else if (choice == 2) {

            keyboard = new Scanner(System.in);

            System.out.print("Enter title: ");

            String title = keyboard.nextLine();

            store.removeItem(store.searchByTitle(title));

        } else if (choice == 0) {

            break;

        } else {

            System.out.println("Invalid choice please choose again.");

        }

    }

}

Hàm storeMenu()

public static void storeMenu() {

    for (; ; ) {

        isPlayable = false;

        System.out.println("\nOptions: ");

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

        System.out.println("1. See a media is 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");

        Scanner keyboard = new Scanner(System.in);

        System.out.print("Your choice: ");

        int choice = keyboard.nextInt();

        if (choice == 1) {

            keyboard = new Scanner(System.in);

            System.out.print("Enter title: ");

            String title = keyboard.nextLine();

            media checkMedia = store.searchByTitle(title);

            if (checkMedia == null) {

                continue;

            }

            if (checkMedia instanceof Disc) {

                isPlayable = true;

            }

            currentMedia = checkMedia;

            mediaDetailsMenu();

        } else if (choice == 2) {

            keyboard = new Scanner(System.in);

            System.out.print("Enter title: ");

            String title = keyboard.nextLine();

            media checkMedia = store.searchByTitle(title);

            if (checkMedia == null) {

                continue;

            }

            cart.addItem(checkMedia);

        } else if (choice == 3) {

            keyboard = new Scanner(System.in);

            System.out.print("Enter title: ");

            String title = keyboard.nextLine();

            media checkMedia = store.searchByTitle(title);

            if (checkMedia == null) {

                continue;

            }

            if (checkMedia instanceof Disc) {

                Disc tmp = (Disc) checkMedia;

                tmp.play();

                continue;

            }

            System.out.println("This media type is not playable.");

        } else if (choice == 4) {

            cart.printCart();

            cartMenu();

        } else if (choice == 0) {

            break;

        } else {

            System.out.println("Invalid choice please choose again.");

        }

    }

}

Hàm mediaDetailMenu:

public static void mediaDetailsMenu() {

    for (; ; ) {

        System.out.println("\nOptions: ");

        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");

        Scanner keyboard = new Scanner(System.in);

        System.out.print("Your choice: ");

        int choice = keyboard.nextInt();

        if (choice == 1) {

            cart.addItem((media) currentMedia);

            break;

        } else if (choice == 2) {

            if (isPlayable) {

                Disc tmp = (Disc) currentMedia;

                tmp.play();

                continue;

            }

            System.out.println("This media type is not playable.");

        } else if (choice == 0) {

            break;

        } else {

            System.out.println("Invalid choice please choose again.");

        }

    }

}

Hàm cartMenu:

public static void cartMenu() {

    for (; ; ) {

        System.out.println("\nOptions: ");

        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");

        Scanner keyboard = new Scanner(System.in);

        System.out.print("Your choice: ");

        int choice = keyboard.nextInt();

        if (choice == 1) {

            int x;

            for (; ; ) {

                System.out.println("\nOptions: ");

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

                System.out.println("1. Filter by id");

                System.out.println("2. Filter by title");

                System.out.println("0. Back");

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

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

                keyboard = new Scanner(System.in);

                System.out.print("Your choice: ");

                x = keyboard.nextInt();

                if (x == 1) {

                    keyboard = new Scanner(System.in);

                    System.out.print("Enter id: ");

                    int id = keyboard.nextInt();

                    cart.filterById(id);

                    break;

                } else if (x == 2) {

                    keyboard = new Scanner(System.in);

                    System.out.print("Enter title: ");

                    String title = keyboard.nextLine();

                    cart.filterByTitle(title);

                    break;

                } else if (x == 0) {

                    break;

                } else {

                    System.out.println("Invalid choice please choose again.");

                }

            }

        } else if (choice == 2) {

            int x;

            for (; ; ) {

                System.out.println("\nOptions: ");

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

                System.out.println("1. Sort by title");

                System.out.println("2. Sort by cost");

                System.out.println("0. Back");

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

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

                keyboard = new Scanner(System.in);

                System.out.print("Your choice: ");

                x = keyboard.nextInt();

                if (x == 1) {

                    cart.sortByTitle();

                    break;

                } else if (x == 2) {

                    cart.sortByCost();

                    break;

                } else if (x == 0) {

                    break;

                } else {

                    System.out.println("Invalid choice please choose again.");

                }

            }

        } else if (choice == 3) {

            keyboard = new Scanner(System.in);

            System.out.print("Enter title: ");

            String title = keyboard.nextLine();

            cart.removeItem(cart.searchByTitle(title));

        } else if (choice == 4) {

            keyboard = new Scanner(System.in);

            System.out.print("Enter title: ");

            String title = keyboard.nextLine();

            media checkMedia = cart.searchByTitle(title);

            if (checkMedia instanceof Disc) {

                Disc tmp = (Disc) checkMedia;

                tmp.play();

                continue;

            }

            System.out.println("This type of media is not playable.");

        } else if (choice == 5) {

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

            cart.clearCart();

        } else if (choice == 0) {

            break;

        } else {

            System.out.println("Invalid choice please choose again.");

        }

    }

Hàm clearCart trong cart:

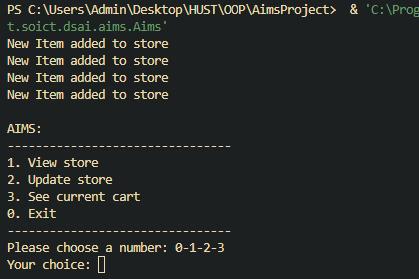
public void clearCart() {

    itemsOrdered.clear();

    System.out.println("Cart is cleared");

}

Chạy chương trình

****

=======================================================