BÁO CÁO THỰC HÀNH LAB 4

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Họ và tên: Dương Văn Nhất

Mã số sinh viên: 20215106

Lớp: 732871

[linkgithub](https://github.com/nyatto03/IT3103.732871.2023.1.20215106.DuongVanNhat)

Table of Contents

[Design 2](#_Toc152875367)

[Creating the abstract Media class 2](#_Toc152875368)

[Book 3](#_Toc152875369)

[Disc 4](#_Toc152875370)

[CompactDisc 5](#_Toc152875371)

[DigitalVideoDisc 6](#_Toc152875372)

[Create the Playable interface 6](#_Toc152875373)

[Update the Cart class to work with Media 7](#_Toc152875374)

[Update the Store class to work with Media 9](#_Toc152875375)

[Constructors of whole classes and parent classes 9](#_Toc152875376)

[Polymorphism with toString() method 9](#_Toc152875377)

[Sort media in the cart 10](#_Toc152875378)

[Create a complete console application in the Aims class 11](#_Toc152875379)

# Design

A diagram of a network

Description automatically generated

# Creating the abstract Media class

Code:

package media;

public abstract class Media {

    private int id;

    private String title;

    private String category;

    private float cost;

    private static int nbMedia = 0;

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

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

        this.id = updateId();

        this.title = title;

        this.category = categoty;

        this.cost = cost;

    }

    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 boolean equals(Media media) {

        if (media == null || this == null)

            return false;

        if (!media.title.equals(this.title))

            return false;

        return true;

    }

    public int updateId() {

        nbMedia++;

        return nbMedia;

    }

}

# Book

Code:

package media;

import java.util.ArrayList;

import java.util.List;

public class Book extends Media {

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

    public List<String> getAuthors() {

        return authors;

    }

    public Book(String title, String category, float cost, List<String> authors) {

        super(title, category, cost);

        this.authors = authors;

    }

    public void setAuthors(List<String> authors) {

        this.authors = authors;

    }

    public void addAuthor(String authorName) {

        if (!authors.contains(authorName)) {

            authors.add(authorName);

        }

    }

    public void removeAuthor(String authorName) {

        if (authors.contains(authorName)) {

            authors.remove(authorName);

        }

    }

}

# Disc

package media;

public abstract class Disc extends Media {

    private int length;

    private String director;

    public int getLength() {

        return length;

    }

    public String getDirector() {

        return director;

    }

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

        super(title, category, cost);

        this.length = length;

        this.director = director;

    }

}

# CompactDisc

package 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<>();

  public String getArtist() {

    return artist;

  }

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

    super(title, category, cost, length, director);

    this.artist = artist;

  }

  public void addTrack(Track track) {

    if (tracks.contains(track)) {

      System.out.println("Track already exists");

    } else {

      tracks.add(track);

    }

  }

  public void removeTrack(Track track) {

    if (tracks.contains(track)) {

      tracks.remove(track);

    } else {

      System.out.println("Track does not exist");

    }

  }

  public int getLength() {

    int totalLength = 0;

    for (Track track : tracks) {

      totalLength += track.getLength();

    }

    return totalLength;

  }

  public void play() {

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

    for (Track track : tracks) {

      track.play();

    }

  }

}

# DigitalVideoDisc

package media;

public class DigitalVideoDisc extends Disc implements Playable {

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

        super(title, category, cost, length, director);

    }

    public void play() {

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

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

    }

}

# Create the Playable interface

package media;

public interface Playable {

    public void play();

}

DigitalVideoDisc.java

public class DigitalVideoDisc extends Disc implements Playable

{

// class body

public void play() {

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

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

    }

}

public class CompactDisc extends Disc implements Playable

{

// class body

public void play() {

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

    for (Track track : tracks) {

      track.play();

    }

  }

}

public class Track implements Playable

{

// class body

public void play() {

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

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

    }

}

# Update the Cart class to work with Media

import media.\*;

import java.util.ArrayList;

public class Cart {

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

    public void addMedia(Media media) {

        itemsOrdered.add(media);

    }

    public void removeMedia(Media media) {

        itemsOrdered.remove(media);

    }

    public float totalCost() {

        float total = 0;

        for (Media media : itemsOrdered) {

            total += media.getCost();

        }

        return total;

    }

    public Media findById(int id) {

        for (Media media : itemsOrdered) {

            if (media.getId() == id) {

                return media;

            }

        }

        return null;

    }

    public void printItem(Media media) {

        System.out.println(media.getTitle() + " - " + media.getCost());

    }

    public Media findByTitle(String title) {

        for (Media media : itemsOrdered) {

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

                return media;

            }

        }

        return null;

    }

    public void removeByTitle(String title) {

        Media mediaToRemove = findByTitle(title);

        if (mediaToRemove != null) {

            itemsOrdered.remove(mediaToRemove);

        }

    }

}

# Update the Store class to work with Media

import media.\*;

import java.util.ArrayList;

public class Store {

    private ArrayList<Media> itemsInStore;

    public Store() {

        itemsInStore = new ArrayList<>();

    }

    public void addMedia(Media media) {

        itemsInStore.add(media);

        System.out.println("The media with name " + media.getTitle() + " has been added");

    }

    public void removeMedia(Media media) {

        if (itemsInStore.contains(media)) {

            itemsInStore.remove(media);

            System.out.println(media.getTitle() + " has been deleted");

        }

    }

    // other methods like find, print inventory

}

# Constructors of whole classes and parent classes

//

# Polymorphism with toString() method

import media.\*;

import java.util.ArrayList;

public class PolymorphismTest {

    public static void main(String[] args) {

        ArrayList<Media> mediaList = new ArrayList<>();

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

        authors.add("J.K. Rowling");

        Book book = new Book("Harry Potter", "Fiction", 10, authors);

        CompactDisc cd = new CompactDisc("Album", "Music", 15, 80, "Director", "Artist");

        authors.add("J.K. Rowling\_2");

        Book book2 = new Book("Harry Potter 2", "Fiction 2", 10, authors);

        DigitalVideoDisc dvd = new DigitalVideoDisc("Movie", "Movies", "Director", 120, 20);

        // Add different media types

        mediaList.add(book);

        mediaList.add(cd);

        mediaList.add(book2);

        mediaList.add(dvd);

        // Iterate and print using toString()

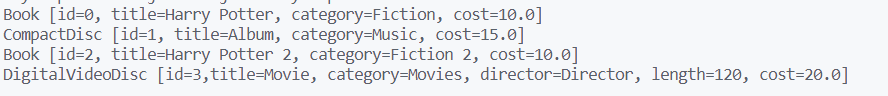
        for (Media m : mediaList) {

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

        }

    }

}



Giải thích: duyệt danh sách, phương thức toString() sẽ được gọi cho mỗi đối tượng. Tuy nhiên phương thức thực thi cụ thể sẽ phụ thuộc vào loại đối tượng:

* Đối với Sách, phương thức toString() của lớp Sách sẽ được gọi.
* Đối với CD, phương thức của lớp CD sẽ được gọi.
* Và tương tự với DVD.

Đây là ví dụ minh họa đa hình - gọi cùng một phương thức (toString()) nhưng hành vi thực thi phụ thuộc vào kiểu đối tượng cụ thể tại thời điểm chạy.

# Sort media in the cart

Khi sử dụng phương thức Comparable:

1. Lớp nào cần phải implement Comparable:

* Lớp Media cần phải implement interface Comparable bởi vì chúng ta muốn so sánh các đối tượng Media.

1. Cách triển khai phương thức compareTo():

* Trong lớp Media ghi đè phương thức compareTo() để định nghĩa logic sắp xếp.

1. Hai qui tắc sắp xếp:

* Thêm một enum nhằm chỉ định kiểu sắp xếp cho phương thức compareTo()
* Triển khai compareTo() trả về kết quả sắp xếp dựa vào tham số enum đó.

1. Qui tắc sắp xếp khác nhau cho DVD:

* Tạo lớp DVD kế thừa từ lớp Media
* Ghi đè lại phương thức compareTo() trong lớp DVD để định nghĩa qui tắc sắp xếp cho DVD
* Gọi compareTo() của DVD khi so sánh đối tượng DVD

Điều này cho phép DVD có qui tắc sắp xếp riêng trong khi vẫn duy trì interface Comparable thống nhất.

# Create a complete console application in the Aims class

import java.util.Scanner;

public class Aims {

    private static Store store = new Store();

    private static Cart cart = new Cart();

    public static void main(String[] args) {

        try (Scanner scanner = new Scanner(System.in)) {

            showMenu();

            int choice = scanner.nextInt();

            while (choice != 0) {

                switch (choice) {

                    case 1:

                        viewStoreMenu();

                        break;

                    case 2:

                        updateStoreMenu();

                        break;

                    case 3:

                        viewCartMenu();

                        break;

                }

                showMenu();

                choice = scanner.nextInt();

            }

        }

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

    }

    public static void showMenu() {

        // display main menu options

    }

    public static void viewStoreMenu() {

        // display store items

        displayStoreSubmenu();

    }

    public static void updateStoreMenu() {

        // add/remove items from store

    }

    public static void viewCartMenu() {

        // display cart items

        displayCartSubmenu();

    }

    public static void displayStoreSubmenu() {

        // display store submenu options

    }

    public static void displayCartSubmenu() {

        // display cart submenu options

    }

    public static void addToCart() {

        // add item to cart

    }

    public static void removeFromCart() {

        // remove item from cart

    }

    // other methods

}