OOP Lab 04 Report

Pham Quang Huy – 20215207

1. **Create Book class**

Attributes:

```
public class Book extends Media {
    /* private int id;
    private String title;
    private String category;
    private float cost; */
    4 usages
    private ArrayList<String> authors = new ArrayList<String>();

    1 usage
    public int getId() {
        return id;
    }
```

Methods:

```
public boolean searchAuthor(String authorName){
    boolean found = false;
    for(String element:this.authors){
        if(authorName.equals(element))
            found = true;
    }
    return found;
}

2 usages
public void addAuthor(String authorName){
    boolean found = this.searchAuthor(authorName);
    if(found == false){
        this.authors.add(authorName);
    }
    else
        System.out.println("Already have");
}
```

2. Create the Abstract Media class

```java
abstract public class Media {
    8 usages
    protected int id;
    13 usages
    protected String title;
    9 usages
    protected String category;
    12 usages
    protected float cost;

    1 usage  2 overrides
    public int getId() {
        return id;
    }

    no usages  1 override
    public void setId(int id) {
        this.id = id;
    }

    3 usages  2 overrides
    public String getTitle() {
        return title;
    }
```

```java
    3 usages  2 overrides
    public void setCost(float cost) {
        this.cost = cost;
    }
    3 usages  3 implementations
    abstract public void display();
    no usages
    public static final Comparator<Media> COMPARE_BY_COST = new MediaComparatorByCost();
    no usages
    public static final Comparator<Media> COMPARE_BY_TITLE = new MediaComparatorByTitle();
}
```

3. Create the CompactCD Class:

```java
import java.lang.reflect.Array;
import java.util.ArrayList;

public class CompactDisc extends Disc implements Playable{
    3 usages
    private String artist;
    6 usages
    private ArrayList<Track> tracks = new ArrayList<~>();

    no usages
    public String getArtist() { return artist; }

    no usages
    public void setArtist(String artist) { this.artist = artist; }
    2 usages
    public boolean searchTrack(Track track){
        boolean found = this.tracks.contains(track);
        return found;
    }
    1 usage
    public void addTrack(Track track){
        boolean found = this.searchTrack(track);
        if(found == false)
            this.tracks.add(track);
        else
            System.out.println("The track already exists");
```

```java
1 usage
public void addTrack(Track track){
    boolean found = this.searchTrack(track);
    if(found == false)
        this.tracks.add(track);
    else
        System.out.println("The track already exists");
}

no usages
public void removeTrack(Track track){
    boolean found = this.searchTrack(track);
    if(found == false)
        System.out.println("No such track was found");
    else
        this.tracks.remove(track);
}
```

4. Create the Playable interface:

```java
3 usages  3 implementations
public interface Playable {
    3 usages  3 implementations
    public void play();
}
```

5. Updated the cart class:

```java
public class Cart {
    no usages
    public static final int MAX_NUMBERS_ORDERED = 20;

    2 usages
    public Cart() { System.out.println("An cart has been created"); }

    8 usages
    private ArrayList<Media>  itemsOrdered = new ArrayList<~>();
    7 usages
    public void addMedia(Media media){
        this.itemsOrdered.add(media);
    }

    1 usage
    public void removeMedia(Media media){
        boolean found = this.itemsOrdered.contains(media);
        if(found == true)
            this.itemsOrdered.remove(media);
        else
            System.out.println("No such was found");
    }
```

6. Updated the Store class

```java
import java.util.ArrayList;

public class Store {
    no usages
    public static final int MAX = 200;
    no usages
    private int qtyStored = 0;
    3 usages
    private ArrayList<Media> itemsOrdered = new ArrayList<~>();
    2 usages
    public void addMedia(Media media){
        this.itemsOrdered.add(media);
    }
    no usages
    public void removeMedia(Media media){
        boolean found = this.itemsOrdered.contains(media);
        if(found == true)
            this.itemsOrdered.remove(media);
        else
            System.out.println("No such was found");
    }
```

7. Sort media in the class

```java
import java.util.Comparator;

public class MediaComparatorByCost implements Comparator<Media> {
    @Override
    public int compare(Media o1, Media o2) {
        if(o1.getCost() > o2.getCost()) return 1;
        else if (o1.getCost() < o2.getCost()) return -1;
        else return 0;
    }

    public static void main(String[] args) {
        DigitalVideoDisc dvd = new DigitalVideoDisc( category: "Anime", title: "Doraemon", cost: 3.84f);
        Book book2 = new Book();
        book2.setCost(5.99f);
        MediaComparatorByCost compare1 = new MediaComparatorByCost();
        System.out.println(compare1.compare(dvd,book2));
    }
}
```

```java
import java.util.Comparator;

3 usages
public class MediaComparatorByTitle implements Comparator<Media>{
    @Override
    public int compare(Media o1, Media o2) {
        return o1.getTitle().compareTo(o2.getTitle());
    }
}
```

```java
1 usage
public void sortByTitle(){
    MediaComparatorByTitle compare = new MediaComparatorByTitle();
    Collections.sort(this.itemsOrdered,compare);
    for(Media element:this.itemsOrdered)
        element.display();
}

1 usage
public void sortByCost(){
    MediaComparatorByCost compare = new MediaComparatorByCost();
    Collections.sort(this.itemsOrdered,compare);
    for(Media element:this.itemsOrdered)
        element.display();
}
```

8. Console Application

```java
public class Aims {
    no usages
    public static void showMenu() {
        System.out.println("AIMS: ");
        System.out.println("--------------------------------");
        System.out.println("1. View store");
        System.out.println("2. Update store");
        System.out.println("3. See current cart");
        System.out.println("0. Exit");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3");
    }
    no usages
    public static void storeMenu() {
        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");
    }
```