Object-Oriented Programming

Lab 04: Inheritance and Polymorphism

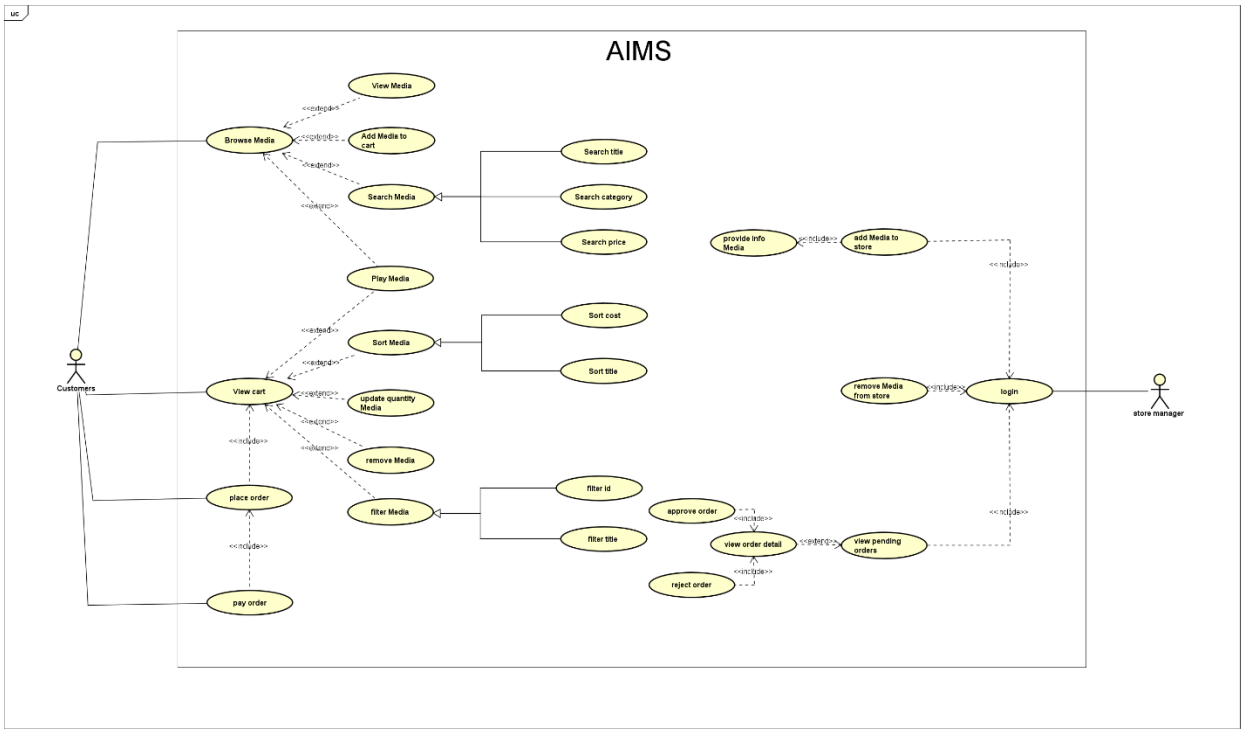
Họ và Tên: Nguyễn Bá Hoàng

MSSV: 20225844

Mã Lớp: 744520

1.Import the existing project into the workspace of Eclipse

2.Additional requirements of AIMS



3.Creating the Book class

```java
package hust.soict.hedspi.aims.media;

import java.util.ArrayList;
import java.util.List;

public class Book extends Media {  11 usages
    private List<String> authors = new ArrayList<String>();  4 usages

    public Book(int id, String title, String category, float cost) {  5 usages
        super(id, title, category, cost);
    }

    public void addAuthor(String authorName){  4 usages
        if(authors.contains(authorName) == false){
            authors.add(authorName);
        }else{
            System.out.println("da ton tai author");
        }
    }

    public void removeAuthor(String authorName){  no usages
        if(authors.contains(authorName) == true){
            authors.remove(authorName);
        }else{
            System.out.println("khong ton tai author de xoa");
        }
    }

    public String toString() {
        String Book_print = this.getTitle();

        if(this.getCategory() != null){
```

```java
    public String toString() {
        String Book_print = this.getTitle();

        if(this.getCategory() != null){
            Book_print += " - ";
            Book_print += this.getCategory();
        }

        return "Book - " + Book_print + ": " + this.getCost() + "$";
    }

}
```

4.Creating the abstract Media class

Book.java Media.java

```java
package hust.soict.hedspi.aims.media;


import java.util.Comparator;

public abstract class Media implements Comparable<Media> {  59 usages  4 inheritors
    private int id;  2 usages
    private String title;  6 usages
    private String category;  2 usages
    private float cost;  4 usages

    public static final Comparator<Media> COMPARE_BY_TITLE_COST = new MediaComparatorByTitleCost();  2 usages
    public static final Comparator<Media> COMPARE_BY_COST_TITLE = new MediaComparatorByCostTitle();  2 usages

    public Media(int id, String title, String category, float cost) {  2 usages
        this.id = id;
        this.title = title;
        this.category = category;
        this.cost = cost;
    }

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

    public String getTitle() {
        return title;
    }

    public String getCategory() {  6 usages
        return category;
    }
```

msProject > src > hust > soict > hedspi > aims > media > Media 3:1 CRLF UTF-8 4 spaces

```java

    public float getCost() {  10 usages
        return cost;
    }


    boolean equals(Media mediaCheck){  no usages
        if(mediaCheck == null){
            return false;
        }

        if(this.title == mediaCheck.title){
            return true;
        }else{
            return false;
        }
    }

    //compareTo theo title
    @Override  1 override
    public int compareTo(Media o){
        int compareTitle = this.title.compareTo(o.title);
        if(compareTitle == 0){
            int compareCost = Float.compare(o.cost,this.cost);
            return  compareCost;
        }
        return compareTitle;
    }
```

```java

    //compareTo theo cost
    /*
    @Override
    public int compareTo(Media o){
        int compareCost = Float.compare(o.cost,this.cost);

        if(compareCost == 0){
            int compareTitle = this.title.compareTo(o.title);
            return compareTitle;
        }
        return compareCost;
    }
    */


    public String toString() {  3 overrides
        String Media_print = this.getTitle();

        if(this.getCategory() != null){
            Media_print += " - ";
            Media_print += this.getCategory();
        }

        return "Media - " + Media_print + ": " + this.getCost() + "$";
    }

}
```

5. Creating the CompactDisc class

5.1. Create the Disc class extending the Media class

```java
package hust.soict.hedspi.aims.media;

public class Disc extends Media {  2 usages  2 inheritors
    private String director;  2 usages
    private int length = 0;  2 usages

    public Disc(int id, String title, String category, float cost, String director, int length) {  2 usages
        super(id, title, category, cost);
        this.director = director;
        this.length = length;
    }

    public String getDirector() {  4 usages
        return director;
    }

    public int getLength() {  5 usages
        return length;
    }
}
```

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

Disc.java | Track.java | CompactDisc.java

```java
package hust.soict.hedspi.aims.media;

public class Track implements Playable {  26 usages
    private String title;  4 usages
    private int length;  4 usages

    public Track(String title,int length) {  9 usages
        this.title = title;
        this.length = length;
    }

    public String getTitle() {
        return title;
    }

    public int getLength() {  4 usages
        return length;
    }

    public void play() {  7 usages
        System.out.println("Playing Track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    }


    boolean equals(Track trackCheck){  no usages
        if(trackCheck == null){
            return false;
        }

        if(this.title == trackCheck.title){
            if(this.length == trackCheck.length){
```

```java
            if(this.length == trackCheck.length){
                return true;
            }else {
                return false;
            }
        }else {
            return false;
        }
    }

}

```

5.3. Open the CompactDisc class

```java
package hust.soict.hedspi.aims.media;

import java.util.ArrayList;
import java.util.List;

public class CompactDisc extends Disc implements Playable {  20 usages
    private String artist;  5 usages
    private List<Track> tracks = new ArrayList<Track>();  7 usages

    public CompactDisc(int id, String title, String category, float cost, String director, String artist) {  5 usages
        super(id, title, category, cost, director, length: 0);
        this.artist = artist;
    }

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

    public void addTrack(Track tracksInput){  9 usages
       for(Track tmp :tracks){
           if(tmp != null) {
               if (tmp.getTitle() == tracksInput.getTitle()) {
                   if(tmp.getLength() == tracksInput.getLength()) {
                       System.out.println("track da ton tai");
                       return;
                   }
               }
           }
       }
       tracks.add(tracksInput);
    }
```

```java

    public void removeTrack(Track tracksInput){  no usages
        for(Track tmp :tracks){
            if(tmp != null) {
                if (tmp.getTitle() == tracksInput.getTitle()) {
                    tracks.remove(tmp);
                    return;
                }
            }
        }
        System.out.println("track khong ton tai");
    }


    public int getLength_CD(){  3 usages
        int count_length =0;
        for(Track tmp :tracks) {
            if (tmp != null) {
                count_length += tmp.getLength();
            }
        }
        return count_length;
    }
```

```java
    public void play() {  7 usages
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("artist: " + artist);
        System.out.println("Number track in CD: " + tracks.size());

        System.out.println("CD length: " + this.getLength_CD());

        //play each tracks
        for(Track tmp :tracks){
            if(tmp != null){
                tmp.play();
            }
        }

    }
```

```java
    public String toString() {
        String CD_print = this.getTitle();

        if(this.getCategory() != null){
            CD_print += " - ";
            CD_print += this.getCategory();
        }

        if(this.getDirector() != null){
            CD_print += " - ";
            CD_print += this.getDirector();
        }

        if(this.getLength_CD() != 0){
            CD_print += " - ";
            CD_print += this.getLength_CD();
        }

        if(this.artist != null){
            CD_print += " - ";
            CD_print += this.artist;
        }

        return "CD - " + CD_print + ": " + this.getCost() + "$";
    }

}
```

6. Create the Playable interface

-Playable interface:

```java
package hust.soict.hedspi.aims.media;

public interface Playable {  3 usages  3 implementations
    public void play();  7 usages  3 implementations
}

```

- DigitalVideoDisc class:

```java
    public void play() {  7 usages  new *
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());
    }

```

- CompactDisc class:

```java
    public void play() {  7 usages
        System.out.println("Playing CD: " + this.getTitle());
        System.out.println("artist: " + artist);
        System.out.println("Number track in CD: " + tracks.size());

        System.out.println("CD length: " + this.getLength_CD());

        //play each tracks
        for(Track tmp :tracks){
            if(tmp != null){
                tmp.play();
            }
        }

    }
```

- Track class:

```java
    public void play() {  7 usages
        System.out.println("Playing Track: " + this.getTitle());
        System.out.println("Track length: " + this.getLength());
    }

```

7. Update the Cart class to work with Media

Cart.java × Store.java

```java
package hust.soict.hedspi.aims.Cart;

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

import java.util.ArrayList;
import java.util.Collections;

public class Cart {  6 usages  hello24680 *

    private ArrayList<Media> itemsOrdered = new ArrayList<Media>();  13 usages

    public void addMedia(Media mediaNew){  2 usages  hello24680 *
        for(Media tmp :itemsOrdered){
            if(tmp != null) {
                if (tmp.getTitle() == mediaNew.getTitle()) {
                    System.out.println("media da ton tai");
                    return;
                }
            }
        }
        itemsOrdered.add(mediaNew);
    }

    public void removeMedia(Media mediaRemove){  1 usage  new *
        for(Media tmp :itemsOrdered){
            if(tmp != null) {
                if (tmp.getTitle() == mediaRemove.getTitle()) {
                    itemsOrdered.remove(tmp);
                    return;
                }
            }
        }
```

```java
        System.out.println("Media khong ton tai");
    }

    public float totalCost(){  1 usage  hello24680 *
        float count_cost = 0;
        for(Media tmp :itemsOrdered){
            if(tmp != null) {
                count_cost += tmp.getCost();
            }
        }
        return count_cost;
    }

    public void sortByTitle(){  1 usage  new *
        Collections.sort(itemsOrdered, Media.COMPARE_BY_TITLE_COST);
        for(Media tmp: itemsOrdered){
            if(tmp != null){
                System.out.println(tmp.toString());
            }
        }
    }

    public void sortByCost() {  1 usage  hello24680 *
        Collections.sort(itemsOrdered, Media.COMPARE_BY_COST_TITLE);
        for(Media tmp: itemsOrdered){
            if(tmp != null){
                System.out.println(tmp.toString());
            }
        }
    }
```

```java
    }

    public void printCart() {  3 usages  hello24680 *
        System.out.println("**********************CART**********************");
        System.out.println("Ordered Items:");

        for (Media tmp : itemsOrdered) {
            System.out.println(tmp.toString());
        }

        System.out.println("Total cost: " + totalCost() + "$");
        System.out.println("*************************************************");
    }

    public Media searchByTitle(String title) {  3 usages  hello24680 *
        for (Media tmp : itemsOrdered) {
            if (tmp.getTitle().equals(title))
                return tmp;
        }

        return null;
    }

    public Media searchById(int id) {  1 usage  hello24680 *
        for (Media tmp : itemsOrdered) {
            if (tmp.getId() == id)
                return tmp;
        }

        return null;
    }
```

```java
    }

    public void empty() {  1 usage  new *
        itemsOrdered.clear();
    }
}
```

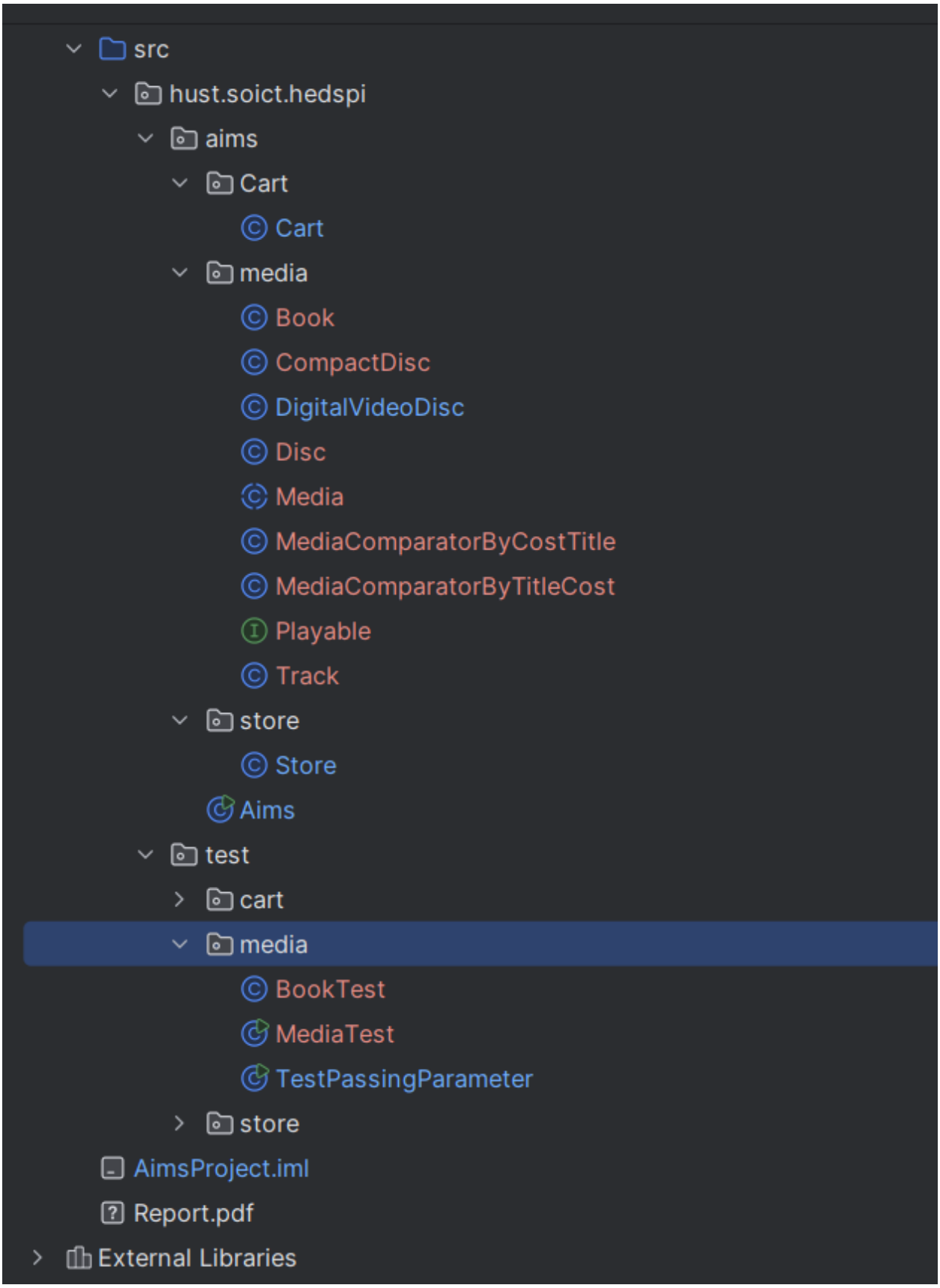
8. Update the Store class to work with Media

```java
public class Store {  6 usages  hello24680 *
    private ArrayList<Media> itemsInStore = new ArrayList<Media>();  6 usages


    public void addMedia(Media mediaNew){  12 usages  hello24680 *
        for(Media tmp :itemsInStore){
            if(tmp != null) {
                if (tmp.getTitle() == mediaNew.getTitle()) {
                    System.out.println("media da ton tai");
                    return;
                }
            }
        }
        itemsInStore.add(mediaNew);
    }

    public void removeMedia(Media mediaRemove){  1 usage  hello24680 *
        for(Media tmp :itemsInStore){
            if(tmp != null) {
                if (tmp.getTitle() == mediaRemove.getTitle()) {
                    itemsInStore.remove(tmp);
                    return;
                }
            }
        }
        System.out.println("Media khong ton tai");
    }
```

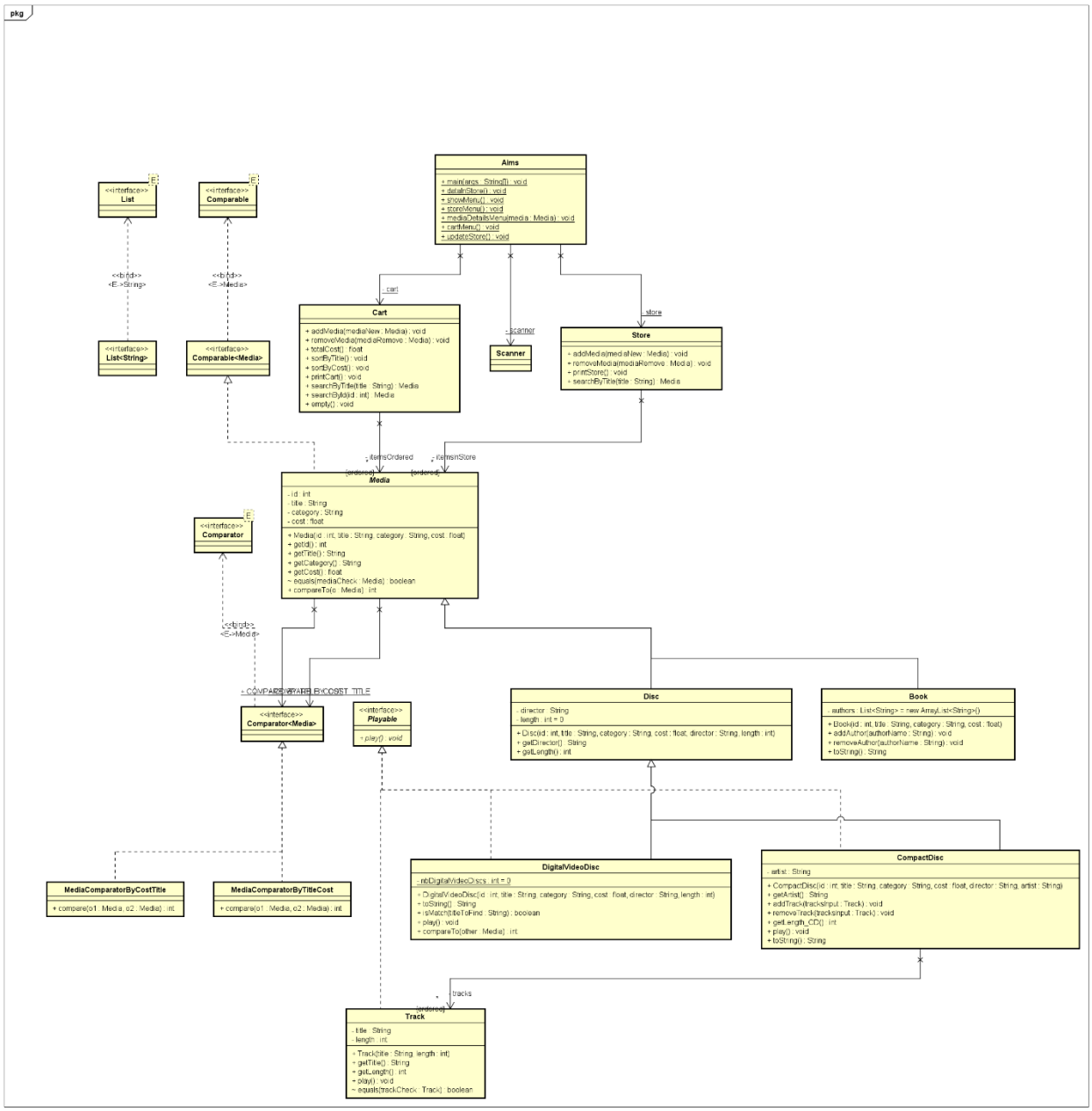
```java
    public void printStore() {  6 usages  hello24680 *
        for (Media tmp : itemsInStore) {
            System.out.println(tmp.toString());
        }
    }

    public Media searchByTitle(String title) {  4 usages  hello24680 *
        for (Media tmp : itemsInStore) {
            if (tmp.getTitle().equals(title)) {
                return tmp;
            }
        }
        return null;
    }


}
```

9. Constructors of whole classes and parent classes

- Cấu trúc project


src
hust.soict.hedspi
aims
Cart
Cart
media
Book
CompactDisc
DigitalVideoDisc
Disc
Media
MediaComparatorByCostTitle
MediaComparatorByTitleCost
Playable
Track
store
Store
Aims
test
cart
media
BookTest
MediaTest
TestPassingParameter
store
AimsProject.iml
Report.pdf
External Libraries

- Class Diagram



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

Trả lời: + media là cha của disc và book; disc là cha của DVDs và CD.

+ các constructor đã được khởi tạo như dưới đây:

- Constructors của các lớp cha và con:

Media:

```
public Media(int id, String title, String category, float cost) {  2 usages
    this.id = id;
    this.title = title;
    this.category = category;
    this.cost = cost;
}
```

Disc:

```
public Disc(int id, String title, String category, float cost, String director, int length) {  2
    super(id, title, category, cost);
    this.director = director;
    this.length = length;
}
```

Book:

```
public Book(int id, String title, String category, float cost) {  5
    super(id, title, category, cost);
}
```

DVDs:

```
public DigitalVideoDisc(int id, String title, String category, float cost, String director, int length) {
    super(id, title, category, cost, director, length);
}
```

CD:

```
public CompactDisc(int id, String title, String category, float cost, String director, String artist) {
    super(id, title, category, cost, director, length: 0);
    this.artist = artist;
}
```

10. Unique item in a list

- Media class:

```
boolean equals(Media mediaCheck){  no usage
    if(mediaCheck == null){
        return false;
    }

    if(this.title == mediaCheck.title){
        return true;
    }else{
        return false;
    }
}
```

- Track class:

```java
boolean equals(Track trackCheck){  no usages
    if(trackCheck == null){
        return false;
    }

    if(this.title == trackCheck.title){
        if(this.length == trackCheck.length){
            return true;
        }else {
            return false;
        }
    }else {
        return false;
    }
}
```

Câu hỏi: If the passing object is not an instance of Media, what happens?

Trả lời: **Nếu đối tượng truyền vào không phải là một instance của Media hoặc không phải con của Media thì sẽ không thực hiện được phép so sánh và java sẽ báo lỗi tại nơi truyền parameter vào phương thức.**

11. Polymorphism with toString() method

-test:

```java
public static void main(String[] args){
    List<Media> mediae = new ArrayList<Media>();

    //create media
     //CD
    CompactDisc CD1 = new CompactDisc( id: 1, title: "MU", category: "1 mon qua", cost: 1.5f, director: "siu", artist: "hoang1");
     //dvd
    DigitalVideoDisc DVD1 = new DigitalVideoDisc( id: 2, title: "Barsa", category: "null", cost: 0.5f, director: "siuu", length: 1);
     //book
    Book Book3 = new Book( id: 3, title: "Che", category: "null", cost: 3f);

    //add vao mediae
    mediae.add(CD1);
    mediae.add(DVD1);
    mediae.add(Book3);

    //test in ra man hinh
    System.out.println("\n");
    System.out.println("test add");
    for(Media m: mediae){
        System.out.println(m.toString());
    }
```

-ket qua:

```
test add
CD - MU - 1 mon qua - siu - hoang1: 1.5$
DVD - Barsa - null - siuu - 1: 0.5$
Book - Che - null: 3.0$

Process finished with exit code 0
```

-Nx:toString() được override để hiển thị chi tiết các thuộc tính
trong các lớp CD, DVD và Book cua media cha. Do đó java sẽ sử dụng toString()
trong các lớp cd,dvd,book thay vì của media.

-media:

```java
public String toString() {  3 overrides
    String Media_print = this.getTitle();

    if(this.getCategory() != null){
        Media_print += " - ";
        Media_print += this.getCategory();
    }

    return "Media - " + Media_print + ": " + this.getCost() + "$";
}
```

-cd:

```java
public String toString() {
    String CD_print = this.getTitle();

    if(this.getCategory() != null){
        CD_print += " - ";
        CD_print += this.getCategory();
    }

    if(this.getDirector() != null){
        CD_print += " - ";
        CD_print += this.getDirector();
    }

    if(this.getLength_CD() != 0){
        CD_print += " - ";
        CD_print += this.getLength_CD();
    }

    if(this.artist != null){
        CD_print += " - ";
        CD_print += this.artist;
    }

    return "CD - " + CD_print + ": " + this.getCost() + "$";
}
```

-dvd:

```java
public String toString() {
    String dvd_print = this.getTitle();

    if(this.getCategory() != null){
        dvd_print += " - ";
        dvd_print += this.getCategory();
    }

    if(this.getDirector() != null){
        dvd_print += " - ";
        dvd_print += this.getDirector();
    }

    if(this.getLength() != 0){
        dvd_print += " - ";
        dvd_print += this.getLength();
    }
    return "DVD - " + dvd_print + ": " + this.getCost() + "$";
}
```

-book:

```java
public String toString() {
    String Book_print = this.getTitle();

    if(this.getCategory() != null){
        Book_print += " - ";
        Book_print += this.getCategory();
    }

    return "Book - " + Book_print + ": " + this.getCost() + "$";
}
```

12. Sort media in the cart

MediaComparatorByCostTitle:

```java
package hust.soict.hedspi.aims.media;

import java.util.Comparator;

public class MediaComparatorByCostTitle implements Comparator<Media> {  1 usage
    @Override
    public int compare(Media o1, Media o2) {
        int compareCost = Float.compare(o2.getCost(),o1.getCost());

        if(compareCost == 0){
            int compareTitle = o1.getTitle().compareTo(o2.getTitle());
            return compareTitle;
        }
        return compareCost;
    }
}
```

MediaComparatorByTitleCost:

```java
package hust.soict.hedspi.aims.media;

import java.util.Comparator;

public class MediaComparatorByTitleCost implements Comparator<Media> {  1 usage
    public int compare(Media o1, Media o2) {
        int compareTitle = o1.getTitle().compareTo(o2.getTitle());

        if(compareTitle == 0){
            int compareCost = Float.compare(o2.getCost(),o1.getCost());
            return compareCost;
        }
        return compareTitle;
    }
}
```

Kết quả:

```
compare by title
DVD - Barsa - null - siuu - 1: 0.5$
Book - Che - null: 3.0$
CD - MU - 1 mon qua - siu - hoang1: 1.5$


compare by cost
Book - Che - null: 3.0$
CD - MU - 1 mon qua - siu - hoang1: 1.5$
DVD - Barsa - null - siuu - 1: 0.5$


compareTo by title
DVD - Barsa - null - siuu - 1: 0.5$
Book - Che - null: 3.0$
CD - MU - 1 mon qua - siu - hoang1: 1.5$


compareTo DVD theo title->length->cost
DVD - Barsa - null - siuu - 2: 2.0$
DVD - Barsa - null - siuu - 2: 0.5$
DVD - Barsa - null - siuu - 1: 0.5$
Book - Che - null: 3.0$
CD - MU - 1 mon qua - siu - hoang1: 1.5$

Process finished with exit code 0
```

Câu hỏi:

a. What class should implement the Comparable interface?

- Lớp chứa đối tượng cần so sánh, chẳng hạn Media hoặc các lớp con của nó như DigitalVideoDisc, Book, CompactDisc.

b. In those classes, how should you implement the compareTo() method be to reflect the ordering that we want?

-triển khai như trên đây:

```java
//compareTo theo title
@Override  1 override
public int compareTo(Media o){
    int compareTitle = this.title.compareTo(o.title);
    if(compareTitle == 0){
        int compareCost = Float.compare(o.cost,this.cost);
        return  compareCost;
    }
    return compareTitle;
}


//compareTo theo cost
/*
@Override
public int compareTo(Media o){
    int compareCost = Float.compare(o.cost,this.cost);

    if(compareCost == 0){
        int compareTitle = this.title.compareTo(o.title);
        return compareTitle;
    }
    return compareCost;
}
*/
```

c. Can we have two ordering rules of the item (by title then cost and by cost then title) if we use this Comparable interface approach?

- Không ta không thể, Comparable chỉ cho phép định nghĩa một quy tắc sắp xếp duy nhất thông qua phương thức compareTo().

Nếu cần nhiều quy tắc thì khi đó ta phải sử dụng Comparator như đã triển khai trong MediaComparatorByCostTitle và MediaComparatorByTitleCost.

d. Suppose the DVDs has a different ordering rule from the other media types, that is by title, then decreasing length, then cost. How would you modify your code to allow this?

-em đã làm như trên đây:

```java
//compareTo DVD theo title->length->cost
@Override  new *
public int compareTo(Media other) {
    if (other instanceof DigitalVideoDisc) {

        DigitalVideoDisc otherDVD = (DigitalVideoDisc) other;

        // So sanh title
        int titleComparison = this.getTitle().compareTo(otherDVD.getTitle());
        if (titleComparison != 0) {
            return titleComparison;
        }

        // So sanh length
        int lengthComparison = Integer.compare(otherDVD.getLength(), this.getLength());
        if (lengthComparison != 0) {
            return lengthComparison;
        }

        // So sanh cost
        return Float.compare(otherDVD.getCost(), this.getCost());
    }

    // cac loai khac su dung quy tac cua lop cha media
    return super.compareTo(other);
}
```

13. Create a complete console application in the Aims class

(do aims class dài hơn 500 dòng và nhiều tổ hợp kiểm tra nên sẽ không viết ở đây mà để code trên git để mọi người check ạ)