

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH
IT3103-744527-2024.1
BÀI THỰC HÀNH -LAB01

Họ và tên sv: **Vương Quốc Huy**

MSSV: **20225637**

Lớp: **Việt Nhật 07-K67**

GVHD: Lê Thị Hoa

HTGD: Đặng Mạnh Cường

Hà Nội 9/2024

BÁO CÁO THỰC HÀNH LAB 4
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Table of Contents

1. Tạo lớp Book	3
2. Tạo lớp abstract Media	4
3. Tạo lớp CompactDisc	6
3.1 Tạo lớp Disc	6
3.2 Tạo lớp Track.....	7
3.3 Chỉnh sửa lớp CompactDisc	8
4. Tạo interface Playable	9
5. Chỉnh sửa lớp Cart để hoạt động với Media.....	10
6. Chỉnh sửa lớp Store để hoạt động với Media	11
7. Các hàm tạo của toàn lớp và lớp cha	12
8. Item duy nhất trong một list	12
9. Test đa hình với phương thức toString().....	13
10. Sắp xếp media trong giỏ hàng.....	14
10.1 Tạo các class so sánh	14
10.2 Test với Collections.sort().....	15
11. Tạo console app hoàn chỉnh	16
12. Class Diagram.....	25

1. Tạo lớp Book

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  public class Book {
5      private int id;
6      private String title;
7      private String category;
8      private float cost;
9      private List<String> authors = new ArrayList<String>();
10
11
12  public Book() {
13  }
14
15
16  public int getId() {
17      return id;
18  }
19
20  public void setId(int id) {
21      this.id = id;
22  }
23
```

```
    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 addAuthor(String authorName) {
    if (authors.contains(authorName)) {
        System.out.println("Author already existed");
        return false;
    } else {
        authors.add(authorName);
        return true;
    }
}

public boolean removeAuthor(String authorName) {
    if (!authors.contains(authorName)) {
        System.out.println("Author doesn't exist");
        return false;
    } else {
        authors.remove(authorName);
        return true;
    }
}

```

2. Tạo lớp abstract Media

```

1  public abstract class Media {
2      private int id;
3      private String title;
4      private String category;
5      private float cost;
6
7      public Media(){}
8
9      public Media(String title, float cost){
10         this.title = title;
11         this.cost = cost;
12     }
13     public String getTitle() {
14         return title;
15     }
16
17     public Media(String title, String category, float cost){
18         this.title = title;
19         this.category = category;
20         this.cost = cost;
21     }
22     public Media(int id, String title, String category, float cost){
23         this(title, category, cost);
24         this.id = id;
25     }

```

```

26
27  ✓ public boolean equals(Object o){
28      if(!(o instanceof Media)) return false;
29      return ((Media) o).getTitle().equals(this.title);
30  }
31  ✓ public int getId(){
32      return id;
33  }
34  }

```

- Kế thừa lớp Media cho Book và DigitalVideoDisc

```

1  ✓ import java.util.ArrayList;
2  import java.util.List;
3
4  public class Book extends Media {
5      private int id;
6      private String title;
7      private String category;
8      private float cost;
9      private List<String> authors = new ArrayList<String>();
10

```

```

49      public boolean addAuthor(String authorName) {
50          if (authors.contains(authorName)) {
51              System.out.println("Author already existed");
52              return false;
53          } else {
54              authors.add(authorName);
55              return true;
56          }
57      }
58
59
60      public boolean removeAuthor(String authorName) {
61          if (!authors.contains(authorName)) {
62              System.out.println("Author doesn't exist");
63              return false;
64          } else {
65              authors.remove(authorName);
66              return true;
67          }
68      }
69  }

```

```
public class DigitalVideoDisc extends Media{
    // private String title;
    // private String category;
    private String director;
    private int length;
    // private float cost;
    private static int nbDigitalVideoDisc = 0;
    //private int id;
}
```

3. Tạo lớp CompactDisc

3.1 Tạo lớp Disc

- Tạo lớp Disc kế thừa lớp Media, có thêm 2 thuộc tính length và director và tạo constructor.

```
public class Disc extends Media{
    private int length;
    private String director;

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

    public Disc(String title, float cost){
        super(title, cost);
    }

    public int getLength(){
        return length;
    }
    public String getDirector(){
        return director;
    }
    public void setDirector(String director){
        this.director = director;
    }
    public void setLength(int length){
        this.length = length;
    }
}
```

- Sửa đổi để lớp DigitalVideoDisc kế thừa lớp Disc

```

public class DigitalVideoDisc extends Disc{
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;
    private static int nbDigitalVideoDisc = 0;
    private int id;
    public DigitalVideoDisc(String title){
        super();
        setTitle(title);
        nbDigitalVideoDisc++;
        setId(nbDigitalVideoDisc);
    }
    public DigitalVideoDisc(String title, float cost){
        super(title, cost);
    }
    public DigitalVideoDisc(String category, String title, float cost){
        this(title);
        setCategory(category);
        setCost(cost);
    }

    public DigitalVideoDisc(String director, String category, String title, float cost){
        this(category, title, cost);
        setDirector(director);
    }
}

```

```

public DigitalVideoDisc(String title, float cost){
    super(title, cost);
}
public DigitalVideoDisc(String category, String title, float cost){
    this(title);
    setCategory(category);
    setCost(cost);
}

public DigitalVideoDisc(String director, String category, String title, float cost){
    this(category, title, cost);
    setDirector(director);
}
public DigitalVideoDisc(String title, String category, String director, int length, float cost){
    this(director, category, title, cost);
    setLength(length);
}
}

```

- Tạo lớp CompactDisc kế thừa lớp Disc

```

public class CompactDisc extends Disc{
    private String artist;
    public CompactDisc(int id, String title, String category, float cost, int length, String director){
        super(id, title, category, cost, length, director);
    }
    public CompactDisc(String title, float cost){
        super(title, cost);
    }
}

```

3.2 Tạo lớp Track

- Tạo lớp Track có 2 thuộc tính title, length và tạo constructor.

```

1  public class Track{
2      private String title;
3      private int length;
4
5      public Track(String title, int length){
6          this.title = title;
7          this.length = length;
8      }
9      public String getTitle(){
10         return title;
11     }
12     public int getLength(){
13         return length;
14     }
15 }

```

3.3 Chỉnh sửa lớp CompactDisc

- Thêm 2 thuộc tính private artist và tracks, tạo getter cho artist, tạo constructor

```

1  import java.util.List;
2  import java.util.ArrayList;
3  public class CompactDisc extends Disc{
4      private String artist;
5      private List<Track> tracks = new ArrayList<>();
6      public CompactDisc(int id, String title, String category, float cost, int length, String director){
7          super(id, title, category, cost, length, director);
8      }
9      public CompactDisc(String title, float cost){
10         super(title, cost);
11     }
12     public CompactDisc(int id, String title, String category, float cost, int length, String director, String artist){
13         super(id, title, category, cost, length, director);
14         this.artist = artist;
15         this.tracks.addAll(tracks);
16     }
17     public boolean addTrack(Track track){
18         if (tracks.contains(track)){
19             System.out.println("Track already existed");
20             return false;
21         } else {
22             tracks.add(track);
23             System.out.println("Track added successfully");
24             return true;
25         }
26     }

```

```

    public boolean removeTrack(Track track){
        if(!tracks.contains(track)){
            System.out.println("No such track found");
            return false;
        } else {
            tracks.remove(track);
            System.out.println("Track removed successfully");
            return true;
        }
    }
    public int getLength(){
        int length = 0;
        for(Track track : tracks){
            length += track.getLength();
        } return length;
    }
}

```


- Tạo các phương thức addTrack và removeTrack.

```
public boolean addTrack(Track track){
    if (tracks.contains(track)){
        System.out.println("Track already existed");
        return false;
    } else {
        tracks.add(track);
        System.out.println("Track added successfully");
        return true;
    }
}

public boolean removeTrack(Track track){
    if(!tracks.contains(track)){
        System.out.println("No such track found");
        return false;
    } else {
        tracks.remove(track);
        System.out.println("Track removed successfully");
        return true;
    }
}
```

- Tạo phương thức getLength trả về tổng length của tất cả các track có trong CD.

```
public int getLength(){
    int length = 0;
    for(Track track : tracks){
        length += track.getLength();
    } return length;
}
```

4. Tạo interface Playable

- Tạo interface Playable và thêm prototype public void play();

```
1 public interface Playable{
2     public void play();
3 }
4
```

- Implement Playable với các lớp CompactDisc, DigitalVideoDisc và Track

```
public class DigitalVideoDisc extends Disc implements Playable{
    private String title;
    private String category;
```

```
import java.util.List;
import java.util.ArrayList;
public class CompactDisc extends Disc implements Playable{
    private String artist;
    private List<Track> tracks = new ArrayList<>();
```

```
public class Track implements Playable{
    private String title;
    private int length;
```

- Implement phương thức play() cho lớp DigitalVideoDisc và Track

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

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

- Implement phương thức play() cho lớp CompactDisc, play tất cả track có trong CD đó.

```
public void play(){
    for(Track track : tracks){
        track.play();
    }
}
```

5. Chỉnh sửa lớp Cart để hoạt động với Media

- Thay thế 2 phương thức addDigitalVideoDisc() và removeDigitalVideoDisc() bằng addMedia() và removeMedia(), thay thế mảng itemsOrdered bằng một đối tượng ArrayList

```
import java.util.List;
import java.util.ArrayList;
public class Cart{
    public static final int MAX_NUMBERS_ORDERED = 20;
    private List<Media> itemsOrdered = new ArrayList<>();
    private int qtyOrdered = 0;
```

```

public boolean addMedia(Media media){
    if(itemsOrdered.size() == MAX_NUMBERS_ORDERED){
        System.out.println("The cart is already full");
        return false;
    }
    else{
        itemsOrdered.add(media);
        System.out.println("Item added");
        return true;
    }
}

public boolean removeMedia( Media media){
    if(!itemsOrdered.contains(media)){
        System.out.println("No such item found");
        return false;
    } else{
        itemsOrdered.remove(media);
        System.out.println("Item removed");
        return true;
    }
}

```

- Chỉnh sửa phương thức totalCost().

```

public float totalCost(){
    float sum = 0f;
    for(Media item : itemsOrdered){
        sum += item.getCost();
    } return sum;
}

```

6. Chỉnh sửa lớp Store để hoạt động với Media

- Thay thế mảng itemsInStore bằng đối tượng ArrayList. Thay đổi các phương thức làm việc với DigitalVideoDisc ban đầu thành làm việc với Media.

```

import java.util.List;
import java.util.ArrayList;
public class Store {
    public static final int MAX_NUMBERS_DVD = 100;
    private List<Media> itemsInStore = new ArrayList<>();
    // private int qtyOrdered = 0;

    public boolean addMedia(Media media){
        if(itemsInStore.size() == MAX_NUMBERS_DVD){
            System.out.println("Store is already full");
            return false;
        }
        else{
            itemsInStore.add(media);
            System.out.println("Item added successfully");
            return true;
        }
    }
    public boolean removeMedia( Media media){
        if(!itemsInStore.contains(media)){
            System.out.println("Item not found");
            return false;
        } else{
            itemsInStore.remove(media);
            System.out.println("Item removed successfully");
            return true;
        }
    }
}

```

```

}
public boolean removeMedia(String title){
    for(Media item : itemsInStore){
        if(item.getTitle().equals(title)){
            itemsInStore.remove(item);
            System.out.println("Item removed");
            return true;
        }
    }
    System.out.println("item not found");
    return false;
}
}

```

7. Các hàm tạo của toàn lớp và lớp cha

8. Item duy nhất trong một list

- Ghi đè phương thức equals() của lớp Media.

```
@Override
public boolean equals(Object o){
    if(!(o instanceof Media)) return false;
    return ((Media) o).getTitle().equals(this.title);
}
```

- Ghi đè phương thức equals() của lớp Track.

```
@Override
public boolean equals(Object o) {
    if (!(o instanceof Track)) return false;
    Track other = (Track) o;
    return other.getTitle().equals(title) && other.getLength() == length;
}
```

- Câu hỏi: Khi ghi đè phương thức equals() của lớp Object, chúng ta phải cast tham số o thành kiểu đang cần xử lý. Ví dụ, đối với lớp Media, chúng ta cần cast o thành kiểu Media. Nếu truyền vào một đối tượng không phải instance của Media, điều gì sẽ xảy ra? → Nếu không xử lý các kiểu khác với Media, quá trình biên dịch sẽ xảy ra lỗi incompatible type.

9. Test đa hình với phương thức toString()

```
public class ToStringTest {
    Run main | Debug main
    public static void main(String[] args) {
        List<Media> ms = new ArrayList<>();
        ms.add(new DigitalVideoDisc("Doraemon", "Anime", "Fujio", 15f));
        ms.add(new Book(0, "Why we sleep", "Scientific", 9f));
        ms.add(new CompactDisc(1, "Gao ranger", "Super sentai", 20f, 2, "Peter"));

        for(Media m : ms) {
            System.out.println(m.toString());
        }
    }
}
```

10. Sắp xếp media trong giỏ hàng

10.1 Tạo các class so sánh

```
J MediaComparatorByCostTitle.java > ...
1  import java.util.Comparator;
2
3  public class MediaComparatorByCostTitle implements Comparator<Media>{
4      @Override
5      public int compare(Media a, Media b) {
6          return Comparator.comparingDouble(Media::getCost)
7              .reversed()
8              .thenComparing(Media::getTitle)
9              .compare(a, b);
10     }
11 }
12
```

```
J MediaComparatorByTitleCost.java > MediaComparatorByTitleCost
1  import java.util.Comparator;
2
3  public class MediaComparatorByTitleCost implements Comparator<Media> {
4      @Override
5      public int compare(Media a, Media b) {
6          return Comparator.comparing(Media::getTitle)
7              .thenComparingDouble((o) -> -o.getCost())
8              .compare(a, b);
9      }
10 }
```

10.2 Test với Collections.sort()

J ComparatorTest.java > ...

```
1  import java.util.ArrayList;
2  import java.util.Collections;
3  import java.util.List;
4
5  public class ComparatorTest {
6      Run main | Debug main
7      public static void main(String[] args) {
8          List<Media> m = new ArrayList<>();
9          m.add(new Book(0, "B-Book", "History book", 10f));
10         m.add(new Book(1, "C-Book", "Music book", 18f));
11         m.add(new Book(1, "C-Book", "Cooking book", 20f));
12         m.add(new Book(2, "A-Book", "Cartoon", 3f));
13
14         System.out.println("Sort by cost then title: ");
15         Collections.sort(m, Media.COMPARE_BY_COST_TITLE);
16         m.forEach(System.out::println);
17
18         System.out.println("\nSort by title then cost: ");
19         Collections.sort(m, Media.COMPARE_BY_TITLE_COST);
20         m.forEach(System.out::println);
21     }
```

11. Tạo console app hoàn chỉnh

```
public class AimsProject{
    public static void main(String[] args){
        //    System.out.println(anOrder.totalCost());
        //    anOrder.listID();
        public static void showMenu() {
            while (true) {
                System.out.println("Vuong Quoc Huy 20225637");
                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("-----");

                int choice = scanner.nextInt();
                switch (choice) {
                    case 0:
                        break;
                    case 1:
                        storeMenu();
                        continue;
                    case 2:
                        updateStoreMenu();
                        continue;
                    case 3:
                        cartMenu();
                        continue;
                    default:
                        System.out.println("Invalid choice, please try again");
                        break;
                }
            }
            break;
        }
    }
}
```

```
public static void updateStoreMenu() {
    while (true) {
        System.out.println("Options:");
        System.out.println("-----");
        System.out.println("1. Add a media");
        System.out.println("2. Remove a media");
        System.out.println("0. Back");
        System.out.println("-----");
        System.out.println("---> ");

        int choice = scanner.nextInt();
        switch (choice) {
            case 0:
                break;
            case 1:
                Media newItem = createNewMedia();
                store.addMedia(newItem);
                continue;
            case 2:
                System.out.println("Enter the media title: ");
                scanner.nextLine();
                String title = scanner.nextLine();
                store.removeMedia(title);
                continue;
            default:
                System.out.println("Invalid choice, please try again");
                break;
        }
    }
}
```



```
public static Media createNewMedia() {  
    Media item = null;  
    System.out.println("Enter the Media info");  
    System.out.println("Title: ");  
    scanner.nextLine();  
    String title = scanner.nextLine();  
    System.out.println("Cost: ");  
    float price = scanner.nextFloat();  
  
    while (true) {  
        System.out.println("Chosse a media type: ");  
        System.out.println("1. Book");  
        System.out.println("2. DVD");  
        System.out.println("3. CD");  
        System.out.println("0. Back");  
  
        int choice = scanner.nextInt();  
  
        switch (choice) {  
            case 0:  
                break;  
            case 1:  
                item = new Book(title, price);  
                break;  
            case 2:  
                item = new DigitalVideoDisc(title, price);  
                break;  
            case 3:  
                break;  
        }  
    }  
}
```

```

        case 3:
            item = new CompactDisc(title, price);
            break;
        default:
            System.out.println("Invalid choice, please try again");
            break;
    }
    break;
}
return item;
}

public static void storeMenu() {
    while (true) {
        store.printItemsInStore();
        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 media");
        System.out.println("4. See current cart");
        System.out.println("0. Exit");
        System.out.println("-----");
        System.out.println("Please choose a number: 0, 1, 2, 3, 4");
        System.out.println("--->");

        String title;
        Media item;
        int choice = scanner.nextInt();
        switch (choice) {

```

```
switch (choice) {  
    case 0:  
        break;  
    case 1:  
        System.out.println("Enter the media title: ");  
        scanner.nextLine();  
        title = scanner.nextLine();  
        item = store.searchByTitle(title);  
        if(item != null) {  
            mediaDetailMenu(item);  
        }  
        continue;  
    case 2:  
        System.out.println("Enter the media title: ");  
        scanner.nextLine();  
        title = scanner.nextLine();  
        item = store.searchByTitle(title);  
        if(item != null) {  
            cart.addMedia(item);  
        }  
        continue;  
    case 3:  
        System.out.println("Enter the media title: ");  
        scanner.nextLine();  
        title = scanner.nextLine();  
        item = store.searchByTitle(title);  
        if(item instanceof Playable) {  
            ((Playable) item).play();  
        } else if (item != null) {  
            System.out.println("This item is not playable");  
        }  
    }  
}
```

```

        System.out.println("This item is not playable");
    }
    continue;
case 4:
    cart.printItemsInCart();
    continue;
default:
    System.out.println("Invalid choise, please try again");
    continue;
}
break;
}
}

public static void mediaDetailMenu(Media item) {
    while (true) {
        System.out.println(item.toString());
        System.out.println("Options: ");
        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 chosse a number: 0, 1, 2");
        System.out.println("---->");

        int choice = scanner.nextInt();
        switch (choice) {
            case 0:

```

```

                case 0:
                    break;
                case 1:
                    cart.addMedia(item);
                    continue;
                case 2:
                    if(item instanceof Book) {
                        System.out.println("Books are not playable");
                    } else if (item instanceof Playable) {
                        ((Playable) item).play();
                    }
                    continue;
                default:
                    System.out.println("Invalid choice, please try again");
                    break;
            }
        }
        break;
    }
}

public static void cartMenu() {
    cart.printItemsInCart();
    while (true) {
        System.out.println("Options: ");
        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 media");
        System.out.println("5. Place order");

```

```
System.out.println("5. Place order");
System.out.println("0. Back");
System.out.println("Please choose a number: 0, 1, 2, 3, 4, 5");
System.out.println("--->");

int choice = scanner.nextInt();
scanner.nextLine();
String title = null;
int id;
Media item = null;
switch (choice) {
    case 0:
        break;
    case 1:
        System.out.println("1. Filter by ID");
        System.out.println("2. Filter by title");
        System.out.print("Choose an option: ");
        int filterChoice = scanner.nextInt();
        scanner.nextLine(); // Xử lý ký tự xuống dòng
        switch (filterChoice) {
            case 1:
                System.out.print("Enter the ID: ");
                id = scanner.nextInt();
                scanner.nextLine(); // Xử lý ký tự xuống dòng
                System.out.println(cart.searchById(id));
                break;
            case 2:
                System.out.print("Enter the title: ");
                title = scanner.nextLine();
                System.out.println(cart.searchByTitle(title));
        }
    }
}
```

```

        System.out.println(cart.searchByTitle(title));
        break;
    default:
        System.out.println("Invalid filter option");
        break;
    }
    break;
case 2:
    System.out.println("1. Sort by cost");
    System.out.println("2. Sourt by title");
    System.out.println("Choose an option: ");
    int sortChoice = scanner.nextInt();
    scanner.nextLine();
    switch (sortChoice) {
        case 1:
            cart.sortByCost();
            cart.printItemsInCart();
            break;
        case 2:
            cart.sortByTitle();
            cart.printItemsInCart();
            break;
        default:
            System.out.println("Invalid sort option");
            break;
    }
case 3:
    System.out.println("Enter the title: ");
    title = scanner.nextLine();
    item = cart.searchByTitle(title);

```

```

        item = cart.searchByTitle(title);
        if(item != null) {
            cart.removeMedia(item);
        }
        cart.printItemsInCart();
        continue;
    case 4:
        System.out.print("Enter a title: ");
        title = scanner.nextLine();
        item = cart.searchByTitle(title);

        if (item == null) {
            System.out.println("Item not found.");
            continue;
        }

        if (item instanceof Playable) {
            ((Playable) item).play();
        } else {
            System.out.println("Item not playable.");
        }
        continue;
    case 5:
        System.out.println("Order info:");
        System.out.println("Total cost: " + cart.totalPrice());
        System.out.println("1. Continue\t2. Back");
        choice = scanner.nextInt();
        scanner.nextLine();
        switch (choice) {
            case 1:

```

```
System.out.println("Order Info: ");
System.out.println("Total cost: " + cart.totalPrice());
System.out.println("1. Continue\t2. Back");
choice = scanner.nextInt();
scanner.nextLine();
switch (choice) {
    case 1:
        cart.emptyCart();
        System.out.println("Order placed successfully");
        break;
    case 2:
        break;
}
continue;
default:
    System.out.println("Invalid choice, please try again");
    break;
}
break;
}
```


12. Class Diagram

