**Báo cáo thực hành Lập trình hướng đối tượng Lab 4**

Họ và tên: Đỗ Gia Huy

MSSV: 20215060

Mục lục

[I. Tạo lớp Book theo yêu cầu đề bài 3](#_Toc152787311)

[II. Tạo lớp trừu tượng Media và sửa lại lớp Book 4](#_Toc152787312)

[1. Tạo lớp trừu tượng Media 4](#_Toc152787313)

[2. Sửa lại lớp Book 5](#_Toc152787314)

[III. Tạo các lớp quan trọng khác 6](#_Toc152787315)

[1. Tạo class Disc extends Media 6](#_Toc152787316)

[1.1. Tạo class Disc 6](#_Toc152787317)

[1.2. Sửa lại lớp DigitalVideoDisc 7](#_Toc152787318)

[2. Tạo lớp Track 8](#_Toc152787319)

[3. Tạo lớp CompactDisc kế thừa Disc 9](#_Toc152787320)

[IV. Tạo interface Playable và chỉnh sửa các lớp 10](#_Toc152787321)

[1. Tạo interface Playable 10](#_Toc152787322)

[2. Sửa lại code các lớp 10](#_Toc152787323)

[2.1. Lớp CompactDisc 10](#_Toc152787324)

[2.2. Lớp DigitalVideoDisc 11](#_Toc152787325)

[V. Tạo hai lớp so sánh, sắp xếp phương tiện cho giỏ hàng 11](#_Toc152787326)

[1. Tạo 2 lớp so sánh 11](#_Toc152787327)

[2. Sủa lại file Media 13](#_Toc152787328)

[3. Trả lời câu hỏi 14](#_Toc152787329)

[VI. Cập nhật lớp Cart để làm việc với lớp Media 16](#_Toc152787330)

[VII. Cập nhật lớp Store để làm việc với lớp Media 17](#_Toc152787331)

[VIII. Đa hình với phương thức toString() 18](#_Toc152787332)

[1. Kiểm tra đa hình bằng các viết code trong thư mục Test 18](#_Toc152787333)

[2. Kết quả 18](#_Toc152787334)

[IX. Sửa lại lớp AIMS (lớp chính) rồi chạy thử 19](#_Toc152787335)

[X. Cập nhật các biểu đồ Diagram 19](#_Toc152787336)

[1. Biểu đồ Usecase Diagram 19](#_Toc152787337)

[2. Biểu đồ Class Diagram 20](#_Toc152787338)

1. Tạo lớp Book theo yêu cầu đề bài

A screenshot of a computer screen

Description automatically generated

Code này chưa tạo phương thức khởi dựng đầy đủ, vì vậy ta phải tạo 1 lớp trừu tượng Media để các lớp khác kế thừa lớp đó, trong đó có lớp Book này

1. Tạo lớp trừu tượng Media và sửa lại lớp Book
2. Tạo lớp trừu tượng Media

A screen shot of a computer

Description automatically generated

1. Sửa lại lớp Book

A screen shot of a computer screen

Description automatically generated

1. Tạo các lớp quan trọng khác
2. Tạo class Disc extends Media
   1. Tạo class Disc

A screen shot of a computer screen

Description automatically generated

* 1. Sửa lại lớp DigitalVideoDisc

A screen shot of a computer screen

Description automatically generated

1. Tạo lớp Track

A screen shot of a computer program

Description automatically generated

1. Tạo lớp CompactDisc kế thừa Disc

A screen shot of a computer screen

Description automatically generated

1. Tạo interface Playable và chỉnh sửa các lớp
2. Tạo interface Playable

A screen shot of a computer program

Description automatically generated

1. Sửa lại code các lớp
   1. Lớp CompactDisc

A screenshot of a computer program

Description automatically generated

* 1. Lớp DigitalVideoDisc

A screen shot of a computer program

Description automatically generated

1. Tạo hai lớp so sánh, sắp xếp phương tiện cho giỏ hàng
2. Tạo 2 lớp so sánh

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer screen

Description automatically generated

1. Sủa lại file Media

A screen shot of a computer screen

Description automatically generated

1. Trả lời câu hỏi

Câu hỏi (Tiếng Việt): Thay vì sử dụng Comparator để so sánh các mục trong giỏ hàng, chúng ta có thể sử dụng giao diện Comparable và ghi đè phương thức compareTo(). Giả sử chúng ta đang áp dụng Comparable interface approach.

* Lớp nào nên triển khai giao diện Comparable?
* Lớp nên triển khai giao diện Comparable là lớp chứa đối tượng mà bạn muốn so sánh, trong trường hợp này, là abstract class “Media”.
* Trong các lớp đó, phải triển khai phương thức compareTo() như thế nào để phản ánh đúng thứ tự mà chúng ta muốn?
* Để triển khai phương thức compareTo() thì cần so sánh các thuộc tính của đối tượng. Đối với yêu cầu sắp xếp theo title sau đó là cost, có thể triển khai như sau:

@Override

public int compareTo(Media other) {

if (!this.getTitle().equals(other.getTitle())) {

return this.getTitle().compareTo(other.getTitle());

} else {

return Double.compare(this.getCost(), other.getCost());

}

}

* Liệu có thể có hai quy tắc sắp xếp (theo title sau đó là cost và theo cost sau đó là title) nếu sử dụng cách tiếp cận này với giao diện Comparable?
* Không, với giao diện Comparable, bạn chỉ có thể có một quy tắc sắp xếp cho mỗi lớp. Điều này là do phương thức compareTo() chỉ trả về một giá trị int.
* Giả sử các đĩa DVD có một quy tắc sắp xếp khác biệt so với các loại phương tiện khác, tức là theo tiêu đề, sau đó là độ dài giảm dần, sau đó là giá. Bạn sẽ làm thế nào để điều chỉnh mã của mình để cho phép điều này?
* Có thể tạo một lớp riêng cho DVD và triển khai phương thức compareTo():

public class DVD extends Media implements Comparable<DVD> {

@Override

public int compareTo(DVD other) {

if (!this.getTitle().equals(other.getTitle())) {

return this.getTitle().compareTo(other.getTitle());

} else if (this instanceof DVD && other instanceof DVD) {

return Integer.compare(((DVD) other).getLength(), ((DVD) this).getLength());

} else {

return Double.compare(this.getCost(), other.getCost());

}

}

}

1. Cập nhật lớp Cart để làm việc với lớp Media

A screen shot of a computer screen

Description automatically generated

1. Cập nhật lớp Store để làm việc với lớp Media

A screen shot of a computer screen

Description automatically generated

1. Đa hình với phương thức toString()
2. Kiểm tra đa hình bằng các viết code trong thư mục Test

A screenshot of a computer screen

Description automatically generated

1. Kết quả

A screenshot of a computer

Description automatically generated

1. Sửa lại lớp AIMS (lớp chính) rồi chạy thử

Em đã làm và lưu trên git rồi.

1. Cập nhật các biểu đồ Diagram
2. Biểu đồ Usecase Diagram

A screenshot of a computer

Description automatically generated

1. Biểu đồ Class Diagram

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated