**Answer OOP4**

Họ và Tên : Trịnh Minh Đạt

MSSV: 20225701

Answer in 12. Sort media in the cart

1. What class should implement the Comparable interface?

Lớp Media nên implement giao diện Comparable, vì nó là lớp cơ sở cho các loại phương tiện trong giỏ hàng. Điều này sẽ cho phép tất cả các lớp con của Media (như Book, DigitalVideoDisc, và CompactDisc) kế thừa hoặc ghi đè phương thức compareTo() để xác định cách sắp xếp.

1. How should you implement the compareTo() method to reflect the ordering that we want?

Để sắp xếp theo **tiêu đề (title) rồi đến giá (cost)**:

**public** **class** MediaComparatorByCostTitle **implements** Comparator<Media> {

@Override

**public** **int** compare(Media m1, Media m2) {

**int** costComparison = Float.*compare*(m2.getCost(), m1.getCost()); // Higher cost first

**if** (costComparison == 0) {

**return** m1.getTitle().compareToIgnoreCase(m2.getTitle()); // Alphabetical order

}

**return** costComparison;

}

}

Để sắp xếp theo **giá (cost) rồi đến tiêu đề (title)**:

**public** **class** MediaComparatorByTitleCost **implements** Comparator<Media> {

@Override

**public** **int** compare(Media m1, Media m2) {

**int** titleComparison = m1.getTitle().compareToIgnoreCase(m2.getTitle());

**if** (titleComparison == 0) {

**return** Float.*compare*(m2.getCost(), m1.getCost()); // Higher cost first

}

**return** titleComparison;

}

}

1. 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, giao diện Comparable chỉ cho phép **một quy tắc sắp xếp tự nhiên** được định nghĩa thông qua phương thức compareTo(). Để có nhiều quy tắc sắp xếp, ta phải sử dụng giao diện Comparator để bổ sung logic sắp xếp khác.

1. Suppose the DVDs have a different ordering rule from the other media types (by title, then decreasing length, then cost). How would you modify your code to allow this?

Có thể ghi đè phương thức compareTo() trong lớp DigitalVideoDisc để phản ánh quy tắc sắp xếp đặc biệt:

package hust.soict.dsai.aims.comparator;

import java.util.Comparator;

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

public class MediaComparatorByTitleCost implements Comparator<Media> {

@Override

public int compare(Media m1, Media m2) {

int titleComparison = m1.getTitle().compareToIgnoreCase(m2.getTitle());

if (titleComparison == 0) {

return Float.compare(m2.getCost(), m1.getCost()); // Higher cost first

}

return titleComparison;

}

}