

ĐẠI HỌC BÁCH KHOA HÀ NỘI
Trường Công nghệ Thông tin và Truyền thông

BÁO CÁO THỰC HÀNH LAB 03

Học phần: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mã học phần: IT3103

Mã lớp: 744523

Giảng viên hướng dẫn: TS. Đàm Quang Tuấn

Sinh viên:

Nguyễn Minh Tuấn

20226095

Hà Nội, tháng 11 năm 2024

Mục lục

Mục lục	2
I. Chụp màn hình của những đoạn code được viết mới.....	3
II. Chụp màn hình của những đoạn debug code và kết quả.....	Error! Bookmark not defined.
III. Cập nhật Use-Case Diagram và Class Diagram	6
1. Use-Case Diagram	6
2. Class Diagram	7
IV. Trả lời phần câu hỏi.....	8

I. Chụp màn hình của những đoạn code được viết mới

1. Working with method overloading

1.1. Overloading by differing types of parameter

```
//Method overloading khác kiểu tham số
/*public void addDigitalVideoDisc(DigitalVideoDisc dvdList[]) {
    for(DigitalVideoDisc disc : dvdList) {
        if(qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD " + "\"" + disc.getTitle() + "\"" + " has been added!");
        } else {
            System.out.println("The cart is full! Cannot add more discs.");
            break;
        }
    }
}
*/
public void addDigitalVideoDisc(DigitalVideoDisc... discs) { 2 usages
    for (DigitalVideoDisc disc : discs) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD \"" + disc.getTitle() + "\" has been added!");
        } else {
            System.out.println("The cart is full! Cannot add more discs.");
            break;
        }
    }
}
```

1.2. Overloading by differing the number of parameters

```

}*/
public void addDigitalVideoDisc(DigitalVideoDisc... discs) { 2 usages
    for (DigitalVideoDisc disc : discs) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD \"" + disc.getTitle() + "\" has been added!");
        } else {
            System.out.println("The cart is full! Cannot add more discs.");
            break;
        }
    }
}

// Method overloading khác số lượng tham số
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) { no usages
    if (qtyOrdered + 2 <= MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered++;
        itemsOrdered[qtyOrdered] = dvd2;
        qtyOrdered++;
        System.out.println("The DVD \"" + dvd1.getTitle() + "\" and \"" + dvd2.getTitle() + "\" have been added!");
    } else {
        System.out.println("The cart is not enough to add two cards!");
    }
}
}

```

2. Passing Parameter

```

public class TestPassingParameter {
    public static void main(String[] args){
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

        swap(jungleDVD, cinderellaDVD);

        System.out.println("Jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Jungle dvd title: " + jungleDVD.getTitle());
    }

    public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) { 1 usage
        DigitalVideoDisc tmp = new DigitalVideoDisc(dvd1.getTitle(), dvd1.getCategory(), dvd1.getDirector(), dvd1.getLength(), dvd1.getCost());

        dvd1.setTitle(dvd2.getTitle());
        dvd1.setCategory(dvd2.getCategory());
        dvd1.setDirector(dvd2.getDirector());
        dvd1.setLength(dvd2.getLength());
        dvd1.setCost(dvd2.getCost());

        dvd2.setTitle(tmp.getTitle());
        dvd2.setCategory(tmp.getCategory());
        dvd2.setDirector(tmp.getDirector());
        dvd2.setLength(tmp.getLength());
        dvd2.setCost(tmp.getCost());
    }

    public static void changeTitle(DigitalVideoDisc dvd, String title) { 1 usage
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
    }
}

```

3. Classifier Member and Instance Member

```
Cart.java TestPassingParameter.java DigitalVideoDisc.java Aims.java TestCart.java TestStore.java
2
3 public class DigitalVideoDisc 47 usages
4 {
5     private String title; 8 usages
6     private String category; 6 usages
7     private String director; 5 usages
8     private int length; 4 usages
9     private float cost; 6 usages
10    private int id; 5 usages
11    private static int nbDigitalVideoDiscs = 0; 9 usages
12    > public String getTitle() { return title; }
15    > public String getCategory() { return category; }
18    > public String getDirector() { return director; }
21    > public int getLength() { return length; }
24    > public float getCost() { return cost; }
27    > public int getId() { return id; }
30    > public static int getNbDigitalVideoDiscs() { return nbDigitalVideoDiscs; }
33    public DigitalVideoDisc(String title) { 3 usages
34        this.title = title;
35        nbDigitalVideoDiscs++;
36        this.id = nbDigitalVideoDiscs;
37    }
38    public DigitalVideoDisc(String title, String category, Float cost) { 3 usages
39        this.category = category;
40        this.title = title;
41        this.cost = cost;
42        nbDigitalVideoDiscs++;
43        this.id = nbDigitalVideoDiscs;
44    }
45    public DigitalVideoDisc(String director, String category, String title, Float cost) { no usages
46        this.director = director;
47        this.category = category;
48        this.title = title;
49        this.cost = cost;

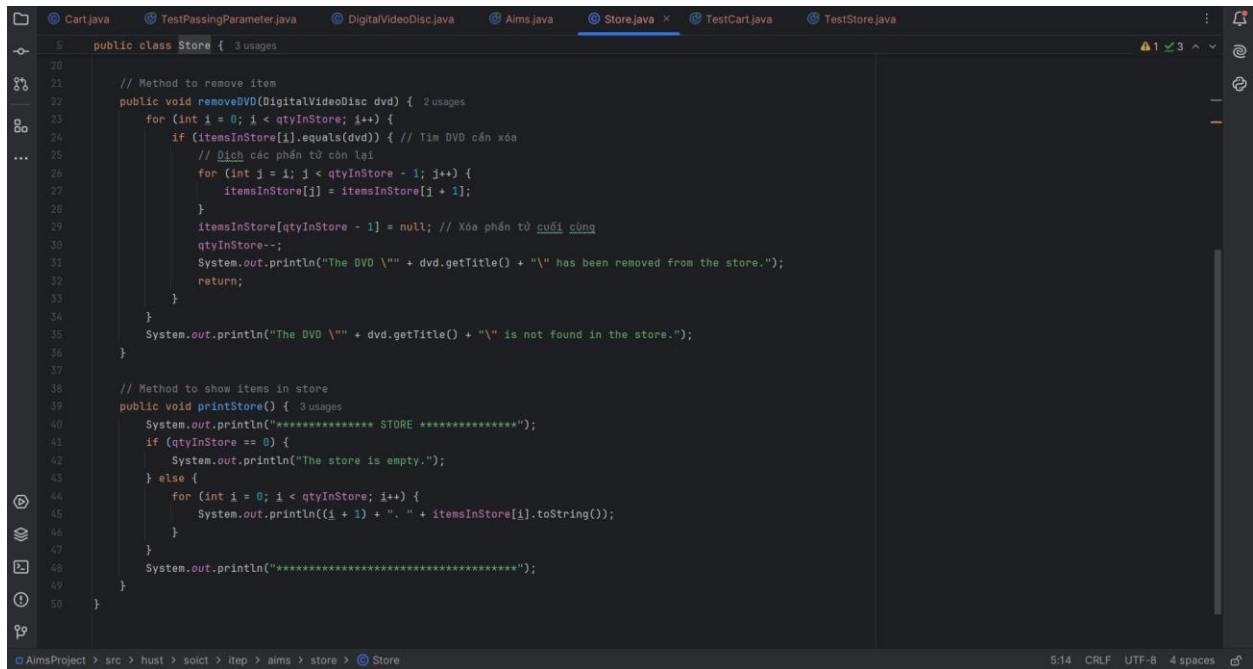
```

4. Open the Cart Class

```
Cart.java TestPassingParameter.java DigitalVideoDisc.java Aims.java TestCart.java TestStore.java
5 public class Cart { 6 usages
7     // Method to print cart
80    public void print() { 1 usage
81        System.out.println("*****CART*****");
82        System.out.println("Ordered Items:");
83        for (int i = 0; i < qtyOrdered; i++) {
84            System.out.println((i + 1) + ". " + itemsOrdered[i].toString());
85        }
86        float totalCost = totalCost();
87        System.out.println("Total cost: " + totalCost + " $");
88        System.out.println("*****");
89    }
90    // Method to search by id
91    public void searchById(int id) { 2 usages
92        boolean found = false;
93        for (int i = 0; i < qtyOrdered; i++) {
94            if (itemsOrdered[i].getId() == id) {
95                System.out.println("DVD with id " + "\"" + id + "\"" + " found:\n" + itemsOrdered[i].toString());
96                found = true;
97                break;
98            }
99        }
100        if (!found) {
101            System.out.println("No DVD found with ID: " + id);
102        }
103    }
104    // Method to search by title
105    public void searchByTitle(String title) { 2 usages
106        boolean found = false;
107        for (int i = 0; i < qtyOrdered; i++) {
108            if (itemsOrdered[i].isMatch(title)) {
109                System.out.println("DVD " + "\"" + title + "\"" + " found:\n" + itemsOrdered[i].toString());
110                found = true;
111            }
112        }
113    }

```

5. Implement the Store Class



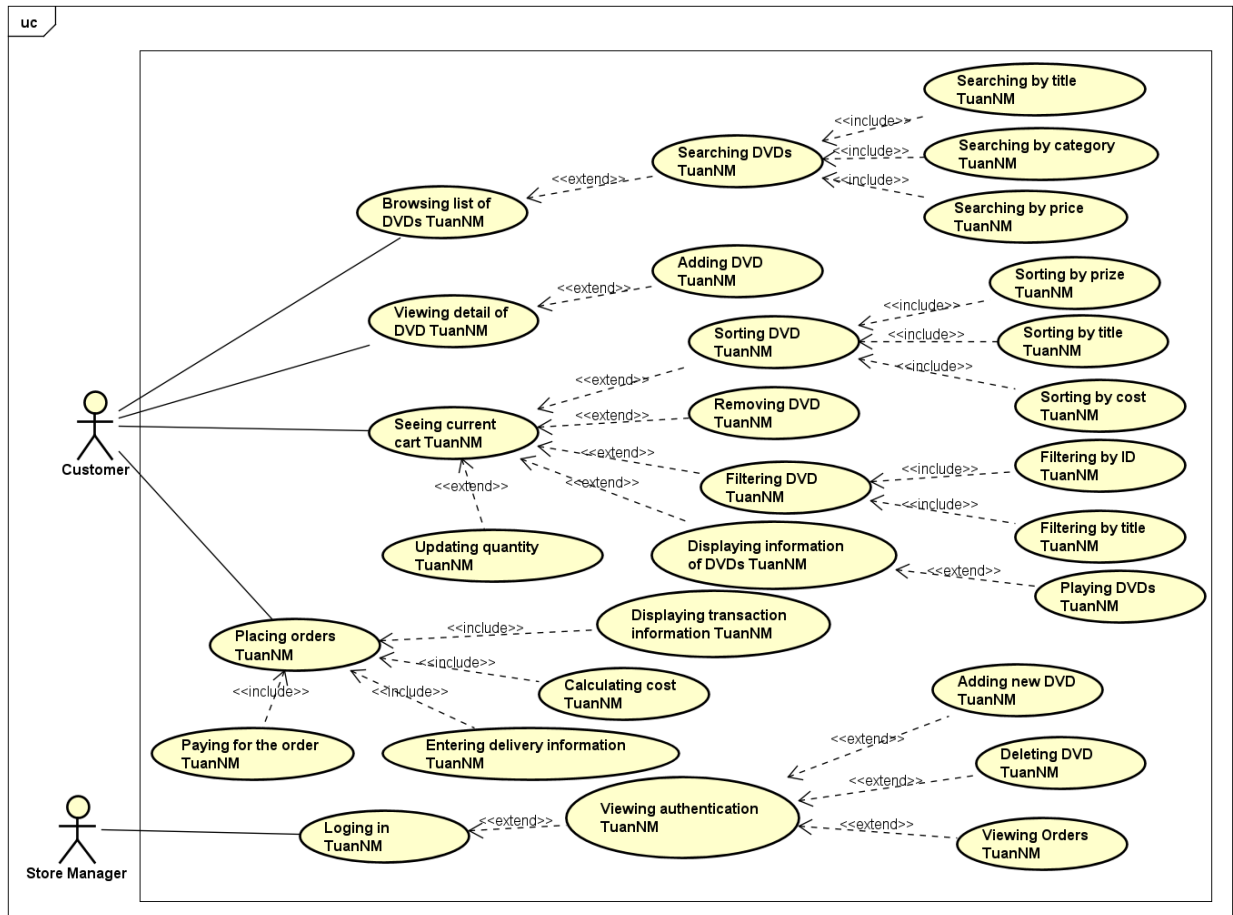
```
1 public class Store { 3 usages
20
21 // Method to remove item
22 public void removeDVD(DigitalVideoDisc dvd) { 2 usages
23     for (int i = 0; i < qtyInStore; i++) {
24         if (itemsInStore[i].equals(dvd)) { // Tìm DVD cần xóa
25             // Dịch các phần tử còn lại
26             for (int j = i; j < qtyInStore - 1; j++) {
27                 itemsInStore[j] = itemsInStore[j + 1];
28             }
29             itemsInStore[qtyInStore - 1] = null; // Xóa phần tử cuối cùng
30             qtyInStore--;
31             System.out.println("The DVD \"" + dvd.getTitle() + "\" has been removed from the store.");
32             return;
33         }
34     }
35     System.out.println("The DVD \"" + dvd.getTitle() + "\" is not found in the store.");
36 }
37
38 // Method to show items in store
39 public void printStore() { 3 usages
40     System.out.println("***** STORE *****");
41     if (qtyInStore == 0) {
42         System.out.println("The store is empty.");
43     } else {
44         for (int i = 0; i < qtyInStore; i++) {
45             System.out.println((i + 1) + ". " + itemsInStore[i].toString());
46         }
47     }
48     System.out.println("*****");
49 }
50 }
```

6. String, StringBuilder and StringBuffer

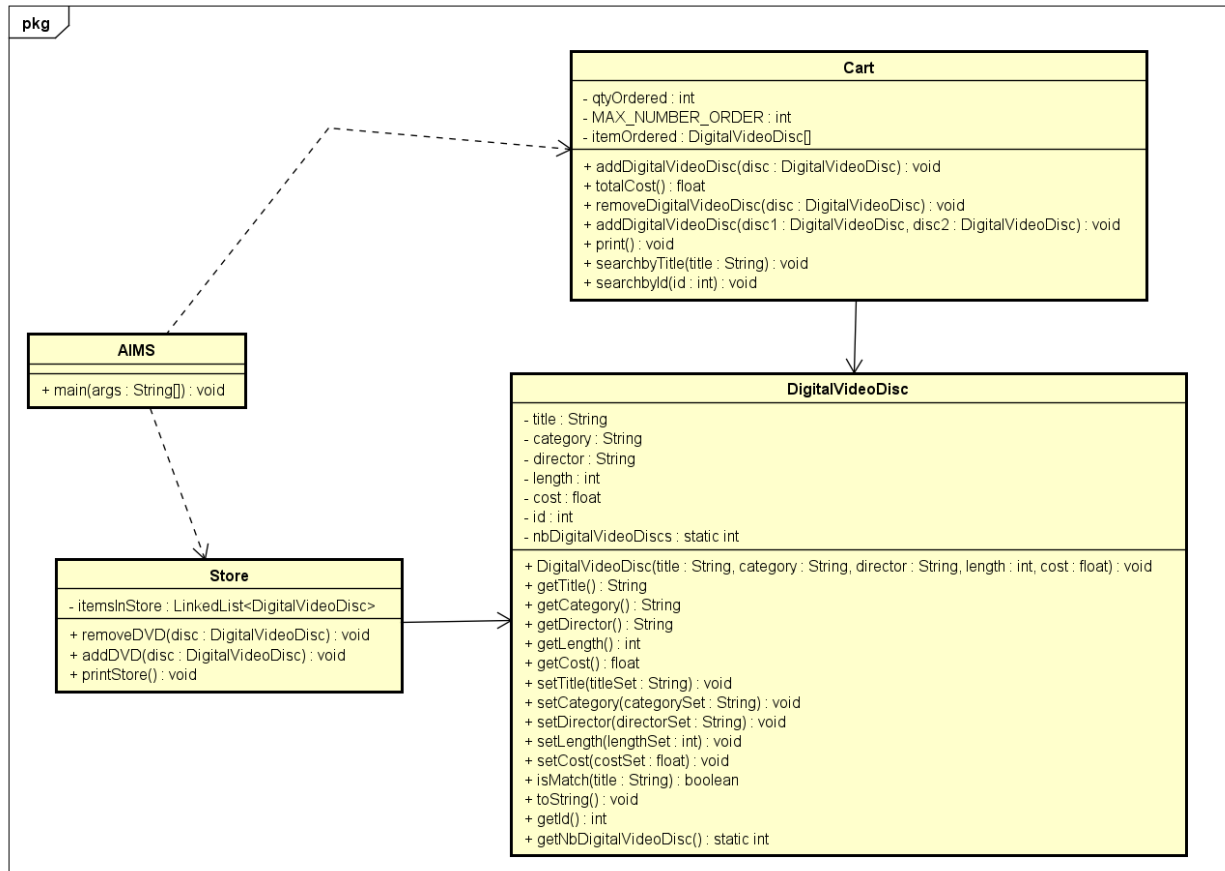
Toàn bộ code nằm trong phần OtherProjects.

II. Cập nhật Use-Case Diagram và Class Diagram

1. Use-Case Diagram



2. Class Diagram



III. Trả lời phần câu hỏi

- Question 1:
 - Question: Is JAVA a Pass by Value or a Pass by Reference programming language?
 - Câu hỏi (Dịch Tiếng Việt): Java là ngôn ngữ lập trình "Pass by Value" hay "Pass by Reference"?
 - Trả lời: Java là một ngôn ngữ lập trình "Pass by Value". Trong Java, khi ta truyền một tham số cho một phương thức, giá trị của tham số được sao chép và truyền vào phương thức. Điều này có nghĩa là nếu ta thay đổi giá trị của tham số bên trong phương thức, giá trị của biến gọi phương thức không bị ảnh hưởng.
- Question 2:
 - After the call of `swap(jungleDVD, cinderellaDVD)` why does the title of these two objects still remain?

- Trả lời: Sau khi thực hiện phương thức `swap(jungleDVD, cinderellaDVD)`, tiêu đề của hai đối tượng vẫn giữ nguyên vì trong Java, tham số được truyền vào phương thức là giá trị của đối tượng, không phải là tham chiếu đến đối tượng. Khi ta thay đổi giá trị của tham số bên trong phương thức (như việc đổi chỗ giữa o1 và o2), sự thay đổi này không ảnh hưởng đến giá trị của các đối tượng gốc.
- Question 3:
 - After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())` why is the title of the JungleDVD changed?
 - Trả lời: Sau khi gọi `changeTitle(jungleDVD, cinderellaDVD.getTitle())`, tiêu đề của jungleDVD bị thay đổi vì trong phương thức `changeTitle`, ta thực hiện thay đổi trực tiếp trên đối tượng dvd (được truyền vào phương thức) bằng cách gọi `dvd.setTitle(title)`. Điều này ảnh hưởng trực tiếp đến đối tượng gốc được truyền vào phương thức.