

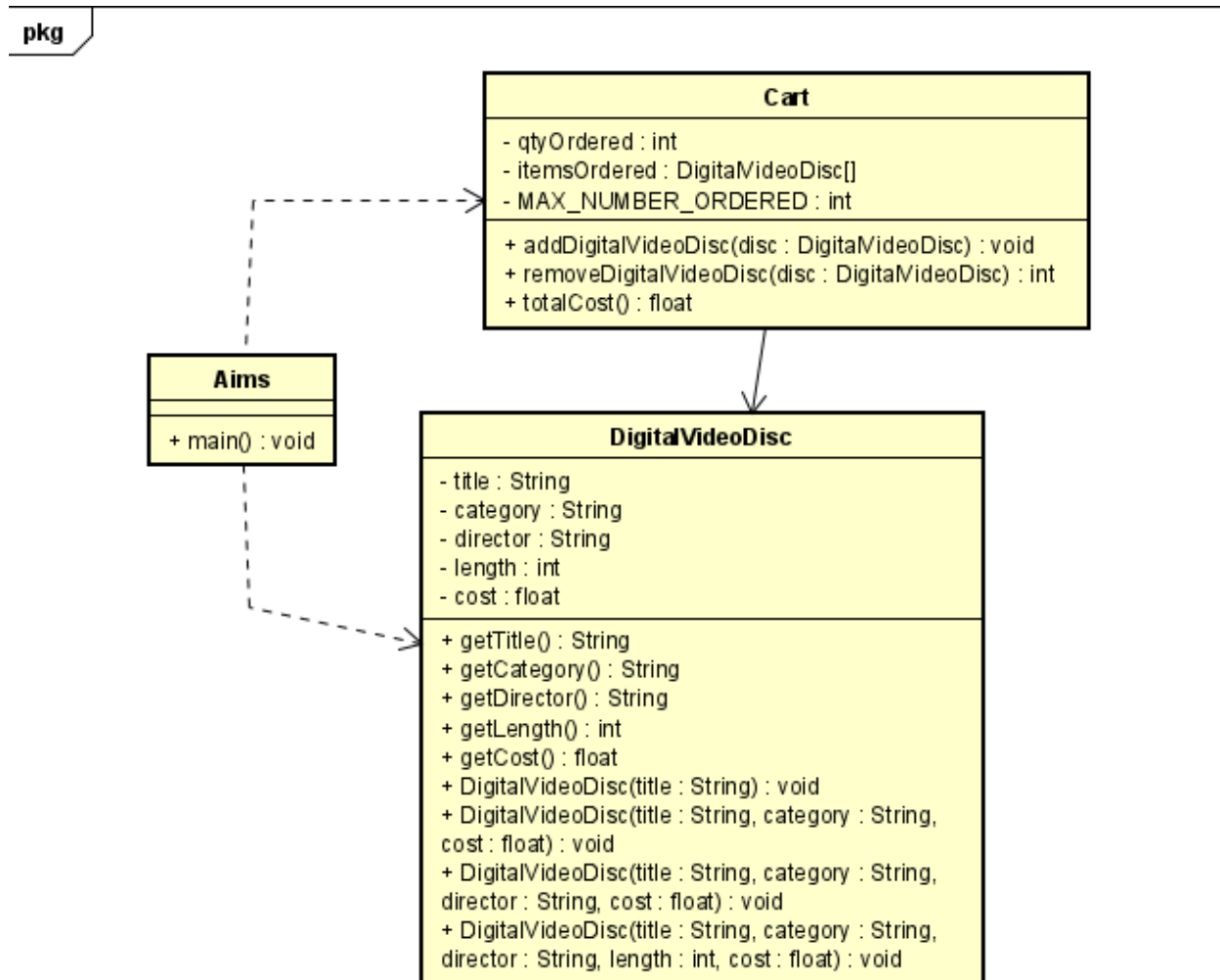
Báo cáo thực hành LAB 03

Họ và tên: Nguyễn Minh Khôi

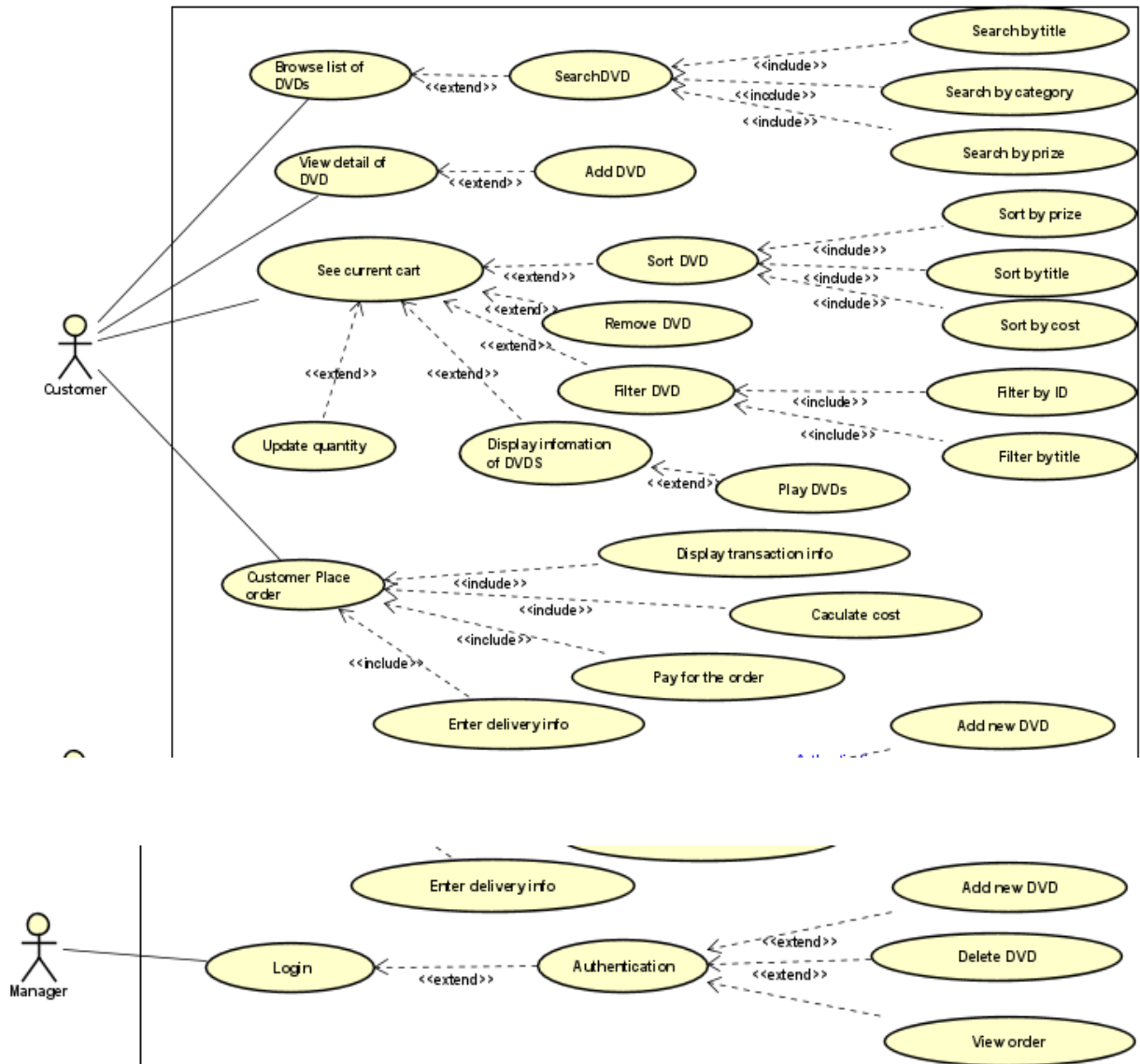
MSSV: 20225642

Mã lớp: 744523

1. Class Diagram



2. Use Case Diagram



3. New Written Code, code debugging and results

3.1. Working with new method overloading

```

//Function which add multiple DVDs with the same name in the cart
public int addDigitalVideoDisc(DigitalVideoDisc[] dvdList){ no usages
    int addingNumber = 0;
    for (DigitalVideoDisc disc : dvdList){
        if (qtyOrdered == MAX_NUMBER_ORDERED){
            System.out.println("The cart is almost full");
            break;
        } else {
            itemsOrdered[qtyOrdered - 1] = disc;
            qtyOrdered++;
            System.out.println("The DVD " + disc.getTitle() + " has been successfully added!");
            addingNumber++;
        }
    }
    return addingNumber;
}

```

```
//Function which add 2 DVDs with different names
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2){ no usages
    if (qtyOrdered + 1 >= MAX_NUMBER_ORDERED){
        System.out.println("The cart is almost full");
    } else {
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered++;
        System.out.println("The DVD " + dvd1.getTitle() + " has been successfully added!");

        itemsOrdered[qtyOrdered] = dvd2;
        qtyOrdered++;
        System.out.println("The DVD " + dvd2.getTitle() + " has been successfully added!");
    }
}
```

3.2. Passing parameters

```
package test.disc;
import DVD.DigitalVideoDisc;

public class TestPassingParameter {
    public static void main(String[] args){
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "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 = dvd1;
        dvd1 = dvd2;
        dvd2 = tmp;
    }

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

- Results for TestPassingParameter:

```
\NguyenMinhKhoi_20225642_LAB02+03 test.disc.TestPassingParameter
Jungle dvd title: Jungle
Cinderella dvd title: Cinderella
Jungle dvd title: Cinderella
```

- Code Debugging:

The screenshot shows a debug console with the following content:

✓ "main"@1 ...: RUNNING Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)

main:15, TestPassingParameter (tes

- args = {String[0]@811} []
- ▼ jungleDVD = {DigitalVideoDisc@813}
 - > title = "Cinderella"
 - category = null
 - director = null
 - length = 0
 - cost = 0.0
 - id = 1
- ▼ cinderellaDVD = {DigitalVideoDisc@815}
 - ▼ title = "Cinderella"
 - > value = {byte[10]@824} [67, 105, 110, 100, 101, 114, 101, 108, 108, 97]
 - coder = 0
 - hash = 0
 - hashIsZero = false
 - category = null
 - director = null
 - length = 0
 - cost = 0.0
 - id = 2

3.3. Classifier Member and Instance Member

- Declare nbDigitalVideoDisc attribute

```
public class DigitalVideoDisc { 46 usages
    //Declare attributes
    private String title; 6 usages
    private String category; 5 usages
    private String director; 4 usages
    private int length; 3 usages
    private float cost; 5 usages
    private static int nbDigitalVideoDisc = 0; // Declared class attribute 4 usages
    private int id; //Instance attribute 6 usages
```

- Add id on every constructors

```

//Constructor by title
public DigitalVideoDisc(String title) { 3 usages
    this.title = title;
    this.id = ++nbDigitalVideoDisc;
}

//Constructor by category, title and cost
public DigitalVideoDisc(String title, String category, float cost) { 4 usages
    this.title = title;
    this.category = category;
    this.cost = cost;
    this.id = ++nbDigitalVideoDisc;
}

//Constructor by title, category , director, cost
public DigitalVideoDisc(String title, String category, String director, float cost) {
    this.title = title;
    this.category = category;
    this.director = director;
    this.cost = cost;
    this.id = ++nbDigitalVideoDisc;
}

// Constructor by all attributes
public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
    this.title = title;
    this.category = category;
    this.director = director;
    this.length = length;
    this.cost = cost;
    this.id = ++nbDigitalVideoDisc;
}

```

- Put id attribute on Getter & Setter

```

public int getId() { return id; } 6 usages
//Setter

public void setId(int id) { this.id = id; } no usages

```

3.4. Open the Cart class

- Create a new method to print the cart

```

//Print function
public void Print(){ 1 usage
    StringBuilder output = new StringBuilder();

    output.append("*****CART*****\n");
    output.append("Ordered Items:\n");
    for (int i = 0; i < qtyOrdered; i++){
        output.append(String.format("%d. [%s] - [%s] - [%s] - [%d]: %.2f $\n",
            i+1,
            itemsOrdered[i].getTitle(),
            itemsOrdered[i].getCategory(),
            itemsOrdered[i].getDirector(),
            itemsOrdered[i].getLength(),
            itemsOrdered[i].getCost()
        ));
    }
    output.append("Total cost: ").append("(").append(totalCost()).append(")");
    System.out.println(output.toString());
}

```

- Create a CartTest class

```

package test.cart;

import Cart.Cart;
import DVD.DigitalVideoDisc;

public class CartTest {
    public static void main(String[] args) {
        Cart cart = new Cart();
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation",
            director: "Roger Allers", length: 87, cost: 19.95f);
        cart.addDigitalVideoDisc(dvd1);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star wars", category: "Science Fiction",
            director: "Geogre Lucas", length: 87, cost: 24.95f);
        cart.addDigitalVideoDisc(dvd2);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", cost: 18.99f);
        cart.addDigitalVideoDisc(dvd3);

        cart.Print();

        //Test search by ID method
        cart.searchDVDsByID(3);
        cart.searchDVDsByID(97);

        //Test search by Title method
        cart.searchDVDsByTitle("The Lion King");
        cart.searchDVDsByTitle("Thien Ly Oi");
    }
}

```

Result in CartTest:

The DVD "The Lion King" has been successfully added!

The DVD "Star wars" has been successfully added!

The DVD "Aladin" has been successfully added!

*****CART*****

Ordered Items:

1. [The Lion King] - [Animation] - [Roger Allers] - [87]: 19.95 \$
2. [Star wars] - [Science Fiction] - [Geogre Lucas] - [87]: 24.95 \$
3. [Aladin] - [Animation] - [null] - [0]: 18.99 \$

Total cost: [63.89]

Result for ID 3 :

3. [Aladin] - [Animation] - [null] - [0]: 18.99 \$

No DVD found!

Result for Title The Lion King :

1. [The Lion King] - [Animation] - [Roger Allers] - [87]: 19.95 \$

No DVD found!

3.5. Implement the Store class

- Create a **Store** class, which contains one attribute **itemsInStore[]** – an array of DVDs available in the store.

```
package Store;

import java.util.HashMap;
import DVD.DigitalVideoDisc;

public class Store { 3 usages
    private DigitalVideoDisc[] itemsInStore; 7 usages
    private int nbItems; 9 usages

    // Constructor to initialize the store with a specified size
    public Store(int storeSize) { 1 usage
        itemsInStore = new DigitalVideoDisc[storeSize];
        nbItems = 0;
    }
}
```

- Implement 2 methods called addDVD and removeDVD

```

// Add a DVD to the store
public void addDVD(DigitalVideoDisc dvd) { 3 usages
    itemsInStore[nbItems] = dvd;
    nbItems++;
    System.out.println("DVD " + dvd.getId() + " have successfully added to the store.");
}

// Remove a DVD from the store
public void removeDVD(int id) { 2 usages
    boolean rightID = false;
    for (int i = 0; i < nbItems; i++) {
        if (itemsInStore[i].getId() == id) {
            // Shift the remaining DVDs to the left to remove the DVD
            for (int j = i; j < nbItems - 1; j++) {
                itemsInStore[j] = itemsInStore[j + 1];
            }
            itemsInStore[nbItems - 1] = null; // Nullify the last item
            nbItems--;
            rightID = true;
            System.out.println("DVD with ID " + id + " have succesfully removed from the store.");
            break;
        }
    }
    if (!rightID) {
        System.out.println("No DVD found with ID: " + id);
    }
}
}

```

- Make a new StoreTest class to test 2 new methods in Store class

```

package test.store;

import DVD.DigitalVideoDisc;
import Store.Store;

public class StoreTest {
    public static void main(String[] args) {

        // Make DVDs
        DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladdin", category: "Animation", cost: 14.99f);

        // Create a store a capacity of 4 DVDs
        Store store = new Store( storeSize: 4);

        // Add DVDs to the store
        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);

        // Display the DVDs in the store
        store.displayStore();

        // Remove a DVD by ID and display the store again
        store.removeDVD(dvd2.getId()); // Removing the second DVD
        store.displayStore();

        // Remove an undefined DVD
        store.removeDVD( id: 97);
    }
}

```

- Result for StoreTest:


```

DVD 1 have successfully added to the store.
DVD 2 have successfully added to the store.
DVD 3 have successfully added to the store.
***** DVDs in Store *****
1. The Lion King
2. Star Wars
3. Aladdin
*****
DVD with ID 2 have succesfully removed from the store.
***** DVDs in Store *****
1. The Lion King
2. Aladdin
*****
No DVD found with ID: 97

```

3.6. Re-organize your project

- Code from a new class ConcatenationInLoops

```

import java.util.*;
import static java.lang.System.currentTimeMillis;

public class ConcatenationInLoops {  ⚠ unknown
    public static void main(String[] args) {  ⚠ unknown
        Random r = new Random( seed: 123);
        long start = currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i++) s += r.nextInt( bound: 2);
        System.out.println(currentTimeMillis() - start);

        r = new Random( seed: 123);
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for(int i = 0; i < 65536; i++)
            sb.append(r.nextInt( bound: 2));
        s += sb.toString();
        System.out.println(System.currentTimeMillis() - start);
    }
}

```

- Result for ConcatenationInLoops:

657

8

- Make a text test.exe file to compare compile time between GarbageCreator and NoGarbage classes

```
1..Trăm năm trong cõi người ta,  
2..Chữ tài chữ mệnh khéo là ghét nhau.  
3..Trải qua một cuộc bể dâu,  
4..Những điều trông thấy mà đau đớn lòng.  
5.. Lạ gì bỉ sắc tư phong,  
6..Trời xanh quen thói má hồng đánh ghen.  
7..Cảo thơm lần giở trước đèn,  
8..Phong tình có lực còn truyền sử xanh.  
9.,Rằng năm Gia Tĩnh triều Minh,  
10.. Bốn phương phẳng lặng, hai kinh vững vàng.  
11..Có nhà viên ngoại họ Vương,  
12..Gia tư nghĩ cũng thường thường bực trung.  
13..Một trai con thứ rất lòng,  
14..Vương Quan là chữ, nổi dòng nho gia.  
15.. Đầu lòng hai ả tố nga,  
16. Thúy Kiều là chị, em là Thúy Vân.  
17. Mai cốt cách, tuyết tinh thần,  
18. Một người một vẻ, mười phân vẹn mười.  
19. Vân xem trang trọng khác vời,  
20.. Khuôn trăng đầy đặn, nét ngài nở nang.  
21.Hoa cười ngọc thốt đoan trang,  
22. Mây thua nước tóc, tuyết nhường màu da.  
23. Kiều càng sắc sảo, mặn mà,  
24. So bề tài, sắc, lại là phần hơn.  
25.. Làn thu thủy, nét xuân sơn,  
26. Hoa ghen thua thắm, liễu hờn kém xanh.
```

✓ 5482 ^ ^

- Code from a new class GarbageCreator

```
import java.nio.file.Paths;  
import java.nio.file.Files;  
import java.io.IOException;  
  
public class GarbageCreator {  
    public static void main(String[] args) throws IOException {  
        String filename = "D:\\\\NguyenMinhKhoi_20225642_00P\\\\NguyenMinhKhoi_20225642_LAB02\\\\OthersProject\\\\src\\\\hust\\\\soict\\\\hedspi\\\\garbage\\\\main\\\\";  
        byte[] inputBytes = { 0 };  
        long startTime, endTime;  
        inputBytes = Files.readAllBytes(Paths.get(filename));  
        startTime = System.currentTimeMillis();  
        String outputString = "";  
        for(byte b : inputBytes) {  
            outputString += (char)b;  
        }  
        endTime = System.currentTimeMillis();  
        System.out.println(endTime - startTime);  
    }  
}
```

⚠ 5 ✓ 3 ^ ^

- Result for GarbageCreator:

3233

- Code from a new class NoGarbage

```
import java.nio.file.Paths;  
import java.nio.file.Files;  
import java.io.IOException;  
  
public class NoGarbage {  
    public static void main(String[] args) throws IOException {  
        String filename = "D:\\\\NguyenMinhKhoi_20225642_00P\\\\NguyenMinhKhoi_20225642_LAB02\\\\OthersProject\\\\src\\\\hust\\\\soict\\\\hedspi\\\\garbage\\\\main\\\\";  
        byte[] inputBytes = { 0 };  
        long startTime, endTime;  
        inputBytes = Files.readAllBytes(Paths.get(filename));  
        startTime = System.currentTimeMillis();  
        StringBuilder outputStringBuilder = new StringBuilder();  
        for (byte b : inputBytes){  
            outputStringBuilder.append((char)b);  
        }  
        endTime = System.currentTimeMillis();  
        System.out.println(endTime - startTime);  
    }  
}
```

⚠ 4 ✓ 3 ^ ^

- Result for NoGarbage:

14

4. Trả lời câu hỏi

4.1. *Is JAVA a Pass by Value or a Pass by Reference programming language?*

- Java là ngôn ngữ lập trình “pass by value”. Ta thấy với đối tượng, giá trị được truyền là bản sao của tham chiếu (không phải bản sao của đối tượng), dẫn đến khả năng thay đổi trạng thái của đối tượng nhưng không thay đổi giá trị tham chiếu ban đầu.

4.2. After the call of **swap(jungleDVD, cinderellaDVD)** why does the title of these two objects still remain?

- Title của hai đối tượng vẫn giữ nguyên vì Java là ngôn ngữ truyền theo giá trị. Khi bạn truyền tham chiếu đối tượng (như jungleDVD và cinderellaDVD) vào một phương thức, Java sẽ truyền một bản sao của tham chiếu, chứ không phải tham chiếu thực tế.

4.3. After the call of **changeTitle(jungleDVD, cinderellaDVD.getTitle())** why is the title of the JungleDVD changed?

- Title của jungleDVD được thay đổi vì phương thức này sửa đổi đối tượng thực tế mà tham chiếu trỏ tới.