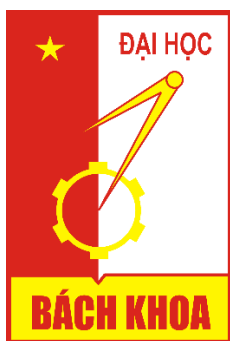


**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO THỰC HÀNH**  
**IT3103 – 744528 - 2024.1**  
**BÀI THỰC HÀNH 3**

Họ và tên sinh viên: Phạm Quốc Cường

MSSV: 20225604

Lớp: Việt Nhật 01 – K67

GVHD: Lê Thị Hoa

HTGD: Đặng Mạnh Cường

Hà Nội 11/2024

# BÁO CÁO THỰC HÀNH LAB 3

## LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Mục lục nội dung

BÁO CÁO THỰC HÀNH LAB 3 .....	2
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG .....	2
1. Working with method overloading.....	3
1.1. Overloading by differing types of parameter .....	3
1.1.1. Code .....	3
1.1.2. Aims Class .....	5
1.1.3. Result.....	5
1.2. Overloading by differing the number of parameters .....	6
1.2.1. Code .....	6
1.2.2. Result.....	7
2. Passing parameter .....	8
2.1. Code.....	8
2.2. Result .....	8
3. Class Member and Instance Member.....	11
3.1. Code.....	11
3.2. Result .....	13
4. Open the Cart class .....	14
4.1. Code.....	14
4.2. Result .....	17
5. Implement the Store class .....	18
5.1. Code.....	18
5.2. Result .....	19
6. String, StringBuilder and StringBuffer .....	20
6.1. Code.....	20
6.2. Result .....	21
7. Class Diagram .....	23

# 1. Working with method overloading

## 1.1. Overloading by differing types of parameter

### 1.1.1. Code

```

1 public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList){
2     for (int i = 0; i< dvdList.length; i++){
3         if (qtyOrdered < MAX_NUMBERS_ORDERED){
4             itemOrdered[qtyOrdered] = dvdList[i];
5             qtyOrdered++;
6             System.out.println("CuongPQ 5604 : The DVD " + dvdList[i].getTitle() + " has been added");
7         }
8         else{
9             System.out.println("CuongPQ 5604 : The cast is almost full! ");
10        }
11    }
12 }

```

Figure 1 Method `addDigitalVideoDisc(DigitalVideoDisc [] dvdList)`

Try to add a method `addDigitalVideoDisc` which allows to pass an arbitrary number of arguments for dvd:

```

1 public void addDigitalVideoDisc(DigitalVideoDisc... dvdList) {
2     for (DigitalVideoDisc disc : dvdList) {
3         if (qtyOrdered == MAX_NUMBERS_ORDERED) {
4             System.out.println("CuongPQ 5604 : The cart is almost full!");
5             break;
6         } else {
7             itemOrdered[qtyOrdered] = disc;
8             qtyOrdered++;
9             System.out.println("CuongPQ 5604 : The DVD " + '\'' + disc.getTitle() + '\'' + " has been added!");
10        }
11    }
12 }
13 }

```

Figure 2 method `addDigitalVideoDisc` which allows to pass an arbitrary number of arguments for dvd

Compare to an array parameter. What do you prefer in this case?

Tiêu chí	...	[]
----------	-----	----

<i>Cú pháp khi gọi phương thức</i>	<i>addDigitalVideoDisc(dvd1, dvd2, dvd3);</i>	<i>addDigitalVideoDisc(new DigitalVideoDisc[] { dvd1, dvd2, dvd3 });</i>
<i>Kích thước tham số</i>	<i>Không giới hạn, có thể truyền số lượng tham số tùy ý</i>	<i>Cần phải xác định số phần tử khi khai báo mảng</i>
<i>Độ dễ đọc</i>	<i>Cú pháp ngắn gọn, dễ đọc hơn</i>	<i>Cần khai báo mảng và rõ ràng hơn trong việc truyền tham số</i>
<i>Yêu cầu khi sử dụng</i>	<i>Không cần phải tạo mảng, có thể trực tiếp truyền các đối tượng</i>	<i>Cần phải tạo một mảng trước khi truyền vào phương thức</i>
<i>Tính linh hoạt</i>	<i>Linh hoạt, có thể truyền bất kỳ số lượng đối tượng nào</i>	<i>Ít linh hoạt hơn, phải truyền mảng có kích thước cố định</i>

*Tôi thích sử dụng addDigitalVideoDisc(DigitalVideoDisc... dvdList) trong trường hợp này vì nó giúp mã nguồn gọn gàng và dễ hiểu hơn.*

### 1.1.2. Aims Class

```
1 public class Aims {
2     public static void main(String[] args) {
3         //Create a new cart
4         Cart anOrder = new Cart();
5
6         //Create new dvd objects and add them to the cart
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King","Animation",
8             "Roger Allers",87,19.95f);
9         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars","Science Fiction",
10            "Geogre Lucas",87,24.95f);
11         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin","Animation",18.99f);
12
13
14         // Add DVD array method
15         DigitalVideoDisc[] dvdList = {dvd1, dvd2, dvd3};
16         anOrder.addDigitalVideoDisc(dvdList);
17
18
19         System.out.printf("CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
20         // System.out.println();
21     }
22 }
```

Figure 3 Aims Class

### 1.1.3. Result

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

CuongPQ 5604 : The DVD 'The Lion King' has been added!
CuongPQ 5604 : The DVD 'Star wars' has been added!
CuongPQ 5604 : The DVD 'Aladin' has been added!
CuongPQ 5604 : Total cost is: 63.89
```

Figure 4 Result

## 1.2. Overloading by differing the number of parameters

### 1.2.1. Code

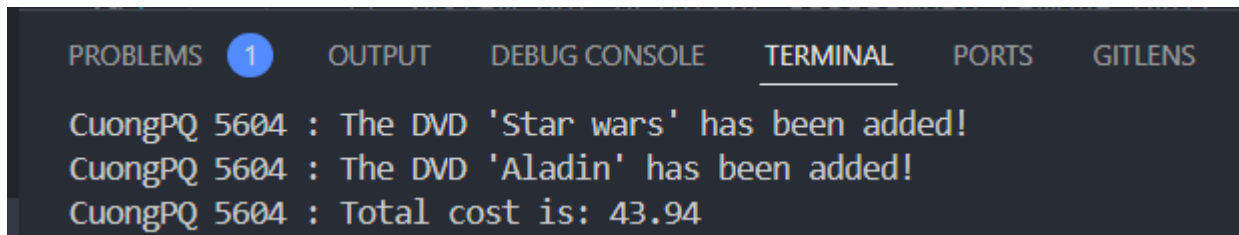
```
1 public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
2     if (qtyOrdered + 1 >= MAX_NUMBERS_ORDERED) {
3         System.out.println("CuongPQ 5604 : The cart is almost full!");
4     } else {
5         itemOrdered[qtyOrdered] = dvd1;
6         qtyOrdered++;
7         System.out.println("CuongPQ 5604 : The DVD " + '\'' + dvd1.getTitle() + '\'' + " has been added!");
8         itemOrdered[qtyOrdered] = dvd2;
9         qtyOrdered++;
10        System.out.println("CuongPQ 5604 : The DVD " + '\'' + dvd2.getTitle() + '\'' + " has been added!");
11    }
12 }
```

Figure 5 Method `addDigitalVideoDisc(DigitalVideoDisc [] dvdList)`

```
1 public class Aims {
2     public static void main(String[] args) {
3         //Create a new cart
4         Cart anOrder = new Cart();
5
6         //Create new dvd objects and add them to the cart
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King","Animation",
8             "Roger Allers",87,19.95f);
9         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars","Science Fiction",
10            "Geogre Lucas",87,24.95f);
11         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin","Animation",18.99f);
12
13         // Add 2 dvd method
14         DigitalVideoDisc[] dvdList = {dvd1, dvd2, dvd3};
15         anOrder.addDigitalVideoDisc(dvd2, dvd3);
16
17         System.out.printf("CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
18     }
19 }
20 }
```

Figure 6 Aims Class

### 1.2.2. Result

A screenshot of a terminal window with a dark background. At the top, there is a horizontal bar with several tabs: 'PROBLEMS', '1' (highlighted with a blue circle), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (underlined), 'PORTS', and 'GITLENS'. Below the tabs, the terminal displays three lines of text in a monospaced font: 'CuongPQ 5604 : The DVD 'Star wars' has been added!', 'CuongPQ 5604 : The DVD 'Aladin' has been added!', and 'CuongPQ 5604 : Total cost is: 43.94'.

*Figure 7 Result*


## 2. Passing parameter

### 2.1. Code

```
1 // Pham Quoc Cuong - 20225604
2
3 package src;
4 import src.DigitalVideoDisc;
5
6 public class TestPassingParameter {
7     public static void main(String[] args) {
8         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
9         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
10
11         swap(jungleDVD, cinderellaDVD);
12         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
13         System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
14
15         changeTitle(jungleDVD, cinderellaDVD.getTitle());
16         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
17     }
18
19     public static void swap(Object o1, Object o2) {
20         Object tmp = o1;
21         o1 = o2;
22         o2 = tmp;
23     }
24
25     public static void changeTitle(DigitalVideoDisc dvd, String title) {
26         String oldTitle = dvd.getTitle();
27         dvd.setTitle(title);
28         dvd = new DigitalVideoDisc(oldTitle);
29     }
30 }
```

Figure 8 Passing parameter code

### 2.2. Result



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
jungle dvd title: Jungle
cinderella dvd title: Cinderella
jungle dvd title: Cinderella
```



*Figure 9 Passing parameter result*

Question: Is JAVA a Pass by Value or a Pass by Reference programming language?

- Java là ngôn ngữ "**pass by value**". Khi truyền tham số kiểu nguyên thủy (int, float, v.v.), Java sẽ sao chép giá trị vào phương thức, nên thay đổi trong phương thức không ảnh hưởng đến giá trị ngoài.
- Còn với đối tượng, Java sẽ sao chép tham chiếu đến đối tượng, không phải chính đối tượng. Điều này có nghĩa là bạn có thể thay đổi trạng thái của đối tượng, nhưng không thể thay đổi tham chiếu (tức là không thể gán đối tượng mới cho tham chiếu gốc).

Question: After the call of swap(jungleDVD, cinderellaDVD) why does the title of these two objects still remain?

- Khi gọi phương thức swap(jungleDVD, cinderellaDVD), Java sao chép **tham chiếu** của các đối tượng jungleDVD và cinderellaDVD vào các tham số o1 và o2 trong phương thức. Tuy nhiên, Java sử dụng cơ chế **pass by value**, nghĩa là nó chỉ sao chép giá trị của tham chiếu (địa chỉ bộ nhớ) vào các tham số, chứ không sao chép chính các đối tượng.
- Vì vậy, mọi thay đổi đối với o1 và o2 trong phương thức chỉ ảnh hưởng đến bản sao tham chiếu của chúng trong phạm vi phương thức, mà không làm thay đổi tham chiếu gốc của jungleDVD và cinderellaDVD bên ngoài.

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

- Khi gọi dvd.setTitle(title), phương thức này thay đổi trực tiếp thuộc tính title của đối tượng mà jungleDVD đang tham chiếu đến.
- Tuy nhiên, việc gán dvd = new DigitalVideoDisc(oldTitle) chỉ thay đổi tham chiếu cục bộ dvd, không làm ảnh hưởng đến tham chiếu jungleDVD ngoài phương thức. Do đó, jungleDVD vẫn giữ đối tượng ban đầu, nhưng tiêu đề của đối tượng đó đã được thay đổi trước khi gán lại tham chiếu.

## 2.3. New Swap Method

```
1 // Please write a swap() method that can correctly swap the two objects:
2 public static void swap(DigitalVideoDisc Object1, DigitalVideoDisc Object2) {
3     DigitalVideoDisc tempObject = new DigitalVideoDisc(Object1.getTitle(), Object1.getCategory(),
4     Object1.getDirector(), Object1.getLength(), Object1.getCost() );
5
6     Object1.setTitle(Object2.getTitle()); // gan title của title 2 cho title 1
7     Object1.setCategory(Object2.getCategory()); // gan Category của Object2 cho Object1
8     Object1.setDirector(Object2.getDirector()); // gan directory của Object2 cho Object1
9     Object1.setLength(Object2.getLength()); // gan lenght của Object2 cho Object1
10    Object1.setCost(Object2.getCost()); // gan cost của Object2 cho Object1
11    Object2.setTitle(tempObject.getTitle());
12    Object2.setCategory(tempObject.getCategory());
13    Object2.setDirector(tempObject.getDirector());
14    Object2.setLength(tempObject.getLength());
15    Object2.setCost(tempObject.getCost());
16 }
```

Figure 10 A swap() method that can correctly swap the two objects

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

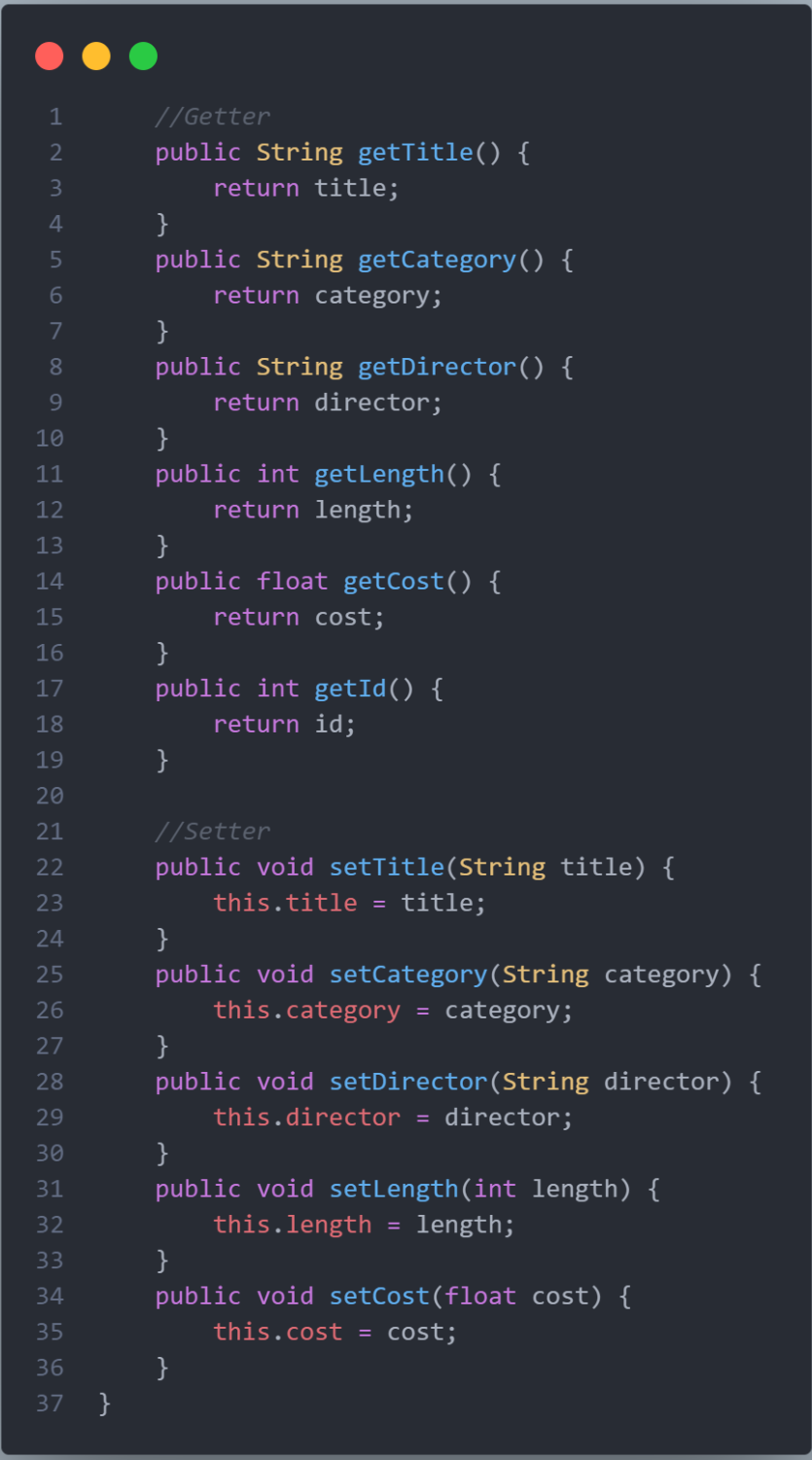
jungle dvd title: Cinderella
cinderella dvd title: Jungle
jungle dvd title: Jungle
```

Figure 11 Result

## 3. Class Member and Instance Member

### 3.1. Code

```
1  // Pham Quoc Cuong - 20225604
2  package src;
3
4  public class DigitalVideoDisc {
5      private static int nbDigitalVideoDiscs = 0; //class variable
6      private String title;
7      private String category;
8      private String director;
9      private int length;
10     private float cost;
11     private int id;
12
13     //Constructor by title
14     public DigitalVideoDisc(String title) {
15         super();
16         this.title = title;
17         this.id = ++nbDigitalVideoDiscs;
18     }
19     // Constructor by category, title and cost
20     public DigitalVideoDisc(String title, String category, float cost) {
21         this.title = title;
22         this.category = category;
23         this.cost = cost;
24         this.id = ++nbDigitalVideoDiscs;
25     }
26     // Constructor by director, category, title and cost
27     public DigitalVideoDisc(String title, String category, String director, float cost) {
28         this.title = title;
29         this.category = category;
30         this.director = director;
31         this.cost = cost;
32         this.id = ++nbDigitalVideoDiscs;
33     }
34     // Constructor by all attributes
35     public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
36         this.title = title;
37         this.category = category;
38         this.director = director;
39         this.length = length;
40         this.cost = cost;
41         this.id = ++nbDigitalVideoDiscs;
42     }
43 }
```



```
1      //Getter
2      public String getTitle() {
3          return title;
4      }
5      public String getCategory() {
6          return category;
7      }
8      public String getDirector() {
9          return director;
10     }
11     public int getLength() {
12         return length;
13     }
14     public float getCost() {
15         return cost;
16     }
17     public int getId() {
18         return id;
19     }
20
21     //Setter
22     public void setTitle(String title) {
23         this.title = title;
24     }
25     public void setCategory(String category) {
26         this.category = category;
27     }
28     public void setDirector(String director) {
29         this.director = director;
30     }
31     public void setLength(int length) {
32         this.length = length;
33     }
34     public void setCost(float cost) {
35         this.cost = cost;
36     }
37 }
```

Figure 12 Class member and Instance member

### 3.2. Result



The screenshot displays an IDE with a Java file open. The code defines three DVD objects and calculates their total cost. The terminal window shows the execution output, confirming the addition of three DVDs and the final total cost.

```
21 System.out.println("CuongPQ 5604 : The id of " + dvd1.getTitle() + " is " + dvd1.getId());
22 System.out.println("CuongPQ 5604 : The id of " + dvd2.getTitle() + " is " + dvd2.getId());
23 System.out.println("CuongPQ 5604 : The id of " + dvd3.getTitle() + " is " + dvd3.getId());
24 You, yesterday * apply-release-flow
25 System.out.printf(format:"CuongPQ 5604 : Total cost is: %.2f",anOrder.totalCost());
26
27
28
```


PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

CuongPQ 5604 : The DVD (1) 'The Lion King' have been added!  
CuongPQ 5604 : The DVD (2) 'Star wars' have been added!  
CuongPQ 5604 : The DVD (3) 'Aladin' have been added!  
CuongPQ 5604 : The id of The Lion King is 1  
CuongPQ 5604 : The id of Star wars is 2  
CuongPQ 5604 : The id of Aladin is 3  
CuongPQ 5604 : Total cost is: 63.89

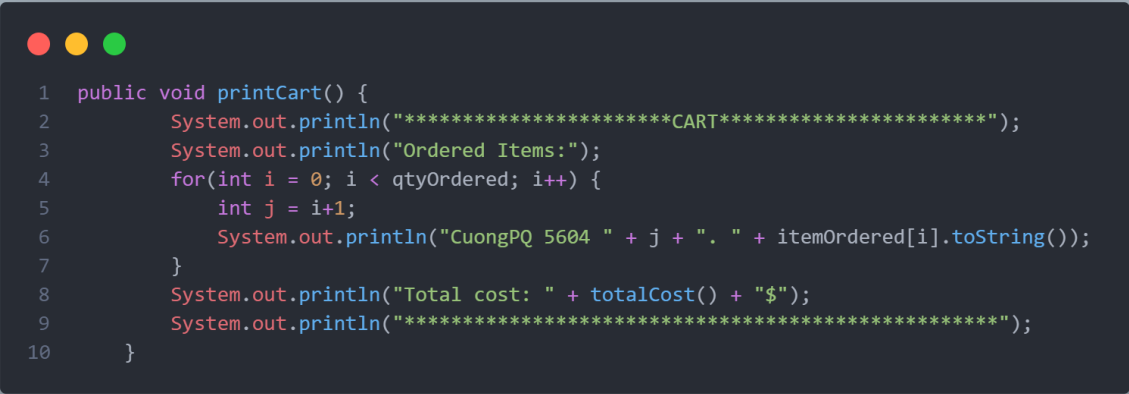
Figure 13 Result

## 4. Open the Cart class

### 4.1. Code




```
1 public String toString() {
2     return "Title='" + title + "'" +
3           "Category='" + category + "'" +
4           "Director='" + director + "'" +
5           "Length='" + length + "'" +
6           "Cost='" + cost + "$'";
7 }
```




```
1 public void printCart() {
2     System.out.println("*****CART*****");
3     System.out.println("Ordered Items:");
4     for(int i = 0; i < qtyOrdered; i++) {
5         int j = i+1;
6         System.out.println("CuongPQ 5604 " + j + ". " + itemOrdered[i].toString());
7     }
8     System.out.println("Total cost: " + totalCost() + "$");
9     System.out.println("*****");
10 }
```

Figure 14 Code method to print the content in Cart



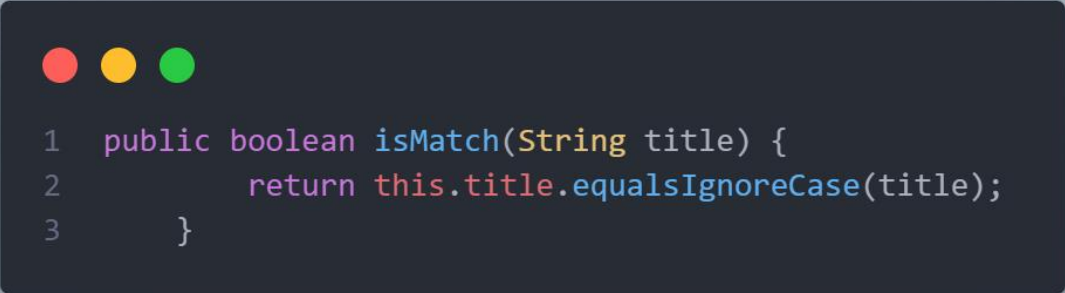
```
1 // search by ID method
2     public DigitalVideoDisc searchById(int id) {
3         for (DigitalVideoDisc disc : itemOrdered) {
4             if (disc != null && disc.getId() == id) {
5                 return disc;
6             }
7         }
8         return null;
9     }
```

Figure 15 Code method search by id



```
1 // search by title method
2     public DigitalVideoDisc searchByTitle(String title) {
3         for (DigitalVideoDisc disc : itemOrdered) {
4             if (disc != null && disc.isMatch(title)) {
5                 return disc;
6             }
7         }
8         return null;
9     }
```

Figure 16 Code method search by title



```
1 public boolean isMatch(String title) {
2     return this.title.equalsIgnoreCase(title);
3 }
```

Figure 17 IsMatch() in DigitalVideoDisc class



```
1 // Pham Quoc Cuong - 20225604
2 package src;
3
4 public class CartTest {
5     public static void main(String[] args){
6         //Creat a new cart
7         Cart cart = new Cart();
8         // Create new dvd object and add them to the cart
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
10        cart.addDigitalVideoDisc(dvd1);
11        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
12        cart.addDigitalVideoDisc(dvd2);
13        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
14        cart.addDigitalVideoDisc(dvd3);
15
16        cart.printCart();
17
18        // search by ID
19        int searchId = 3;
20        DigitalVideoDisc foundById = cart.searchById(searchId);
21        if (foundById != null) {
22            System.out.println("CuongPQ 5604 - Found by ID: " + searchId + " --- " + foundById.toString());
23        }
24        else {
25            System.out.println("CuongPQ 5604 - No DVD with ID: " + searchId);
26        }
27
28        // search by title
29        String searchTitle = "star wars";
30        DigitalVideoDisc foundByTitle = cart.searchByTitle(searchTitle);
31        if (foundByTitle != null) {
32            System.out.println("CuongPQ 5604 - Found by Title: " + searchTitle + " --- " + foundByTitle.toString());
33        }
34        else {
35            System.out.println("CuongPQ 5604 - No DVD with Title: " + searchTitle);
36        }
37    }
38 }
```

Figure 18 CartTest class



## 4.2. Result



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

CuongPQ 5604 : The DVD (1) 'The Lion King' have been added!
CuongPQ 5604 : The DVD (2) 'Star Wars' have been added!
CuongPQ 5604 : The DVD (3) 'Aladin' have been added!
*****CART*****
Ordered Items:
CuongPQ 5604 1. Title='The Lion King'Category='Animation'Director='Roger Allers'Length='87'Cost='19.95$'
CuongPQ 5604 2. Title='Star Wars'Category='Science Fiction'Director='George Lucas'Length='87'Cost='24.95$'
CuongPQ 5604 3. Title='Aladin'Category='Animation'Director='null'Length='0'Cost='18.99$'
Total cost: 63.89$
*****
CuongPQ 5604 - Found by ID: 3 --- Title='Aladin'Category='Animation'Director='null'Length='0'Cost='18.99$'
CuongPQ 5604 - Found by Title: star wars --- Title='Star Wars'Category='Science Fiction'Director='George Lucas'Length='87'Cost='24.95$'
```

*Figure 19 Result*

## 5. Implement the Store class

### 5.1. Code

```
1  package src;
2
3  import java.util.ArrayList;
4
5  public class Store {
6      // list of DVD in Store
7      private ArrayList<DigitalVideoDisc> itemsInStore = new ArrayList<>(); // Dynamic List for flexibility
8
9      // addDVD method
10     private boolean checkDVD(DigitalVideoDisc disc) {
11         for (DigitalVideoDisc digitalVideoDisc : itemsInStore) {
12             if (digitalVideoDisc.equals(disc)) {
13                 return true;
14             }
15         }
16         return false;
17     }
18
19     public void removeDVD(DigitalVideoDisc disc) {
20         if(checkDVD(disc)) {
21             itemsInStore.remove(disc);
22             System.out.println("CuongPQ 5604 - " + disc.getTitle() + " 've been deleted from the store !");
23         } else {
24             System.out.println("CuongPQ 5604 - There is no " + disc.getTitle() + " in the store !");
25         }
26     }
27
28     public void addDVD(DigitalVideoDisc disc) {
29         if(!checkDVD(disc)) {
30             itemsInStore.add(disc);
31             System.out.println("CuongPQ 5604 - " + disc.getTitle() + " 've been added to the store !");
32         } else {
33             System.out.println("CuongPQ 5604 - " + disc.getTitle() + " 'already exists in the store !");
34         }
35     }
36 }
37
38 // print DVD in Store
39 public void printStore() {
40     System.out.println("*****STORE*****");
41     if (itemsInStore.isEmpty()) {
42         System.out.println("CuongPQ 5604 - The store is currently empty.");
43     } else {
44         for (int i = 0; i < itemsInStore.size(); i++) {
45             System.out.println("CuongPQ 5604 - " + (i + 1) + ". " + itemsInStore.get(i).toString());
46         }
47     }
48     System.out.println("*****");
49 }
50 }
51
```

Figure 20 Store Class

```

1  package src;
2
3  public class StoreTest {
4      public static void main(String[] args){
5          //Creat a new cart
6          Store store = new Store();
7          // Create new dvd object and add them to the cart
8          DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
9          DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
10         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
11
12         store.addDVD(dvd1);
13         store.addDVD(dvd2);
14         store.addDVD(dvd3);
15
16         store.printStore();
17
18         store.removeDVD(dvd3);
19         store.removeDVD(dvd2);
20         store.removeDVD(dvd1);
21
22         store.printStore();
23     }
24 }

```

Figure 21 Store Test

## 5.2. Result

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

CuongPQ 5604 - The Lion King 've been added to the store !
CuongPQ 5604 - Star Wars 've been added to the store !
CuongPQ 5604 - Aladin 've been added to the store !
*****STORE*****
CuongPQ 5604 - 1. Title='The Lion King'Category='Animation'Director='Roger Allers'Length='87'Cost='19.95$'
CuongPQ 5604 - 2. Title='Star Wars'Category='Science Fiction'Director='George Lucas'Length='87'Cost='24.95$'
CuongPQ 5604 - 3. Title='Aladin'Category='Animation'Director='null'Length='0'Cost='18.99$'
*****
CuongPQ 5604 - Aladin 've been deleted from the store !
CuongPQ 5604 - Star Wars 've been deleted from the store !
CuongPQ 5604 - The Lion King 've been deleted from the store !
*****STORE*****
CuongPQ 5604 - The store is currently empty.
*****

```

Figure 22 Result

## 6. String, StringBuilder and StringBuffer

### 6.1. Code



```
1 // Pham Quoc Cuong - 20225604
2 package lab03.OtherProjects.hust.soict.hedspi.garbage;
3
4 import java.util.Random;
5
6 public class ConcatenationInLoops {
7     public static void main(String[] args) {
8         Random r = new Random(123);
9         long start = System.currentTimeMillis();
10        String s = "";
11        for (int i = 0; i < 65536; i++) s += r.nextInt(2);
12        System.out.println(System.currentTimeMillis() - start);
13
14        r = new Random(123);
15        start = System.currentTimeMillis();
16        StringBuilder sb = new StringBuilder();
17        for(int i = 0; i < 65536; i++)
18            sb.append(r.nextInt(2));
19        s += sb.toString();
20        System.out.println(System.currentTimeMillis() - start);
21    }
22 }
```

Figure 23 So sánh String và StringBuilder

## 6.2. Result

```

1 package lab03.ther.rojects.hust.soict.hedspi.garbage;
2
3 import java.io.IOException;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 public class GarbageCreator {
8     public static void main(String[] args) throws IOException {
9         String filename = "G:\\OOPLab\\IT3103.732871.2023.1.20215060.DoGiaHuy\\Lab02_03_04_05\\OtherProject\\src\\hust\\soict\\hedspi\\garbage\\test.txt";
10        byte[] inputBytes = { 0 };
11        long startTime, endTime;
12        inputBytes = Files.readAllBytes(Paths.get(filename));
13        startTime = System.currentTimeMillis();
14        String outputString = "";
15        for(byte b : inputBytes) {
16            outputString += (char)b;
17        }
18        endTime = System.currentTimeMillis();
19        System.out.println(endTime - startTime);
20    }
21 }
22

```

Figure 24 GarbageCreator class

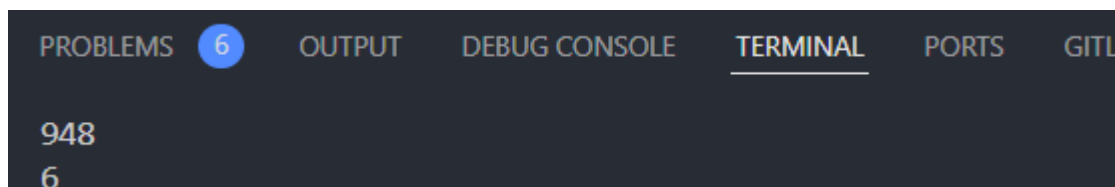


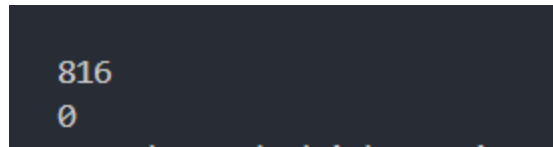
Figure 25 Result: program hangs

```

1 package lab03.ther.rojects.hust.soict.hedspi.garbage;
2
3 import java.io.IOException;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 public class NoGarbage {
8     public static void main(String[] args) throws IOException {
9         String filename = "G:\\OOPLab\\IT3103.732871.2023.1.20215060.DoGiaHuy\\Lab02_03_04_05\\OtherProject\\src\\hust\\soict\\hedspi\\garbage\\test.txt";
10        byte[] inputBytes = { 0 };
11        long startTime, endTime;
12
13        inputBytes = Files.readAllBytes(Paths.get(filename));
14        startTime = System.currentTimeMillis();
15        StringBuilder outputStringBuilder = new StringBuilder("");
16        for(byte b : inputBytes) {
17            outputStringBuilder.append((char)b);
18        }
19        endTime = System.currentTimeMillis();
20        System.out.println(endTime - startTime);
21    }
22 }

```

Figure 26 NoGarbage class



816  
0

*Figure 27 Result: Program finishes quickly*

## 7. Class Diagram

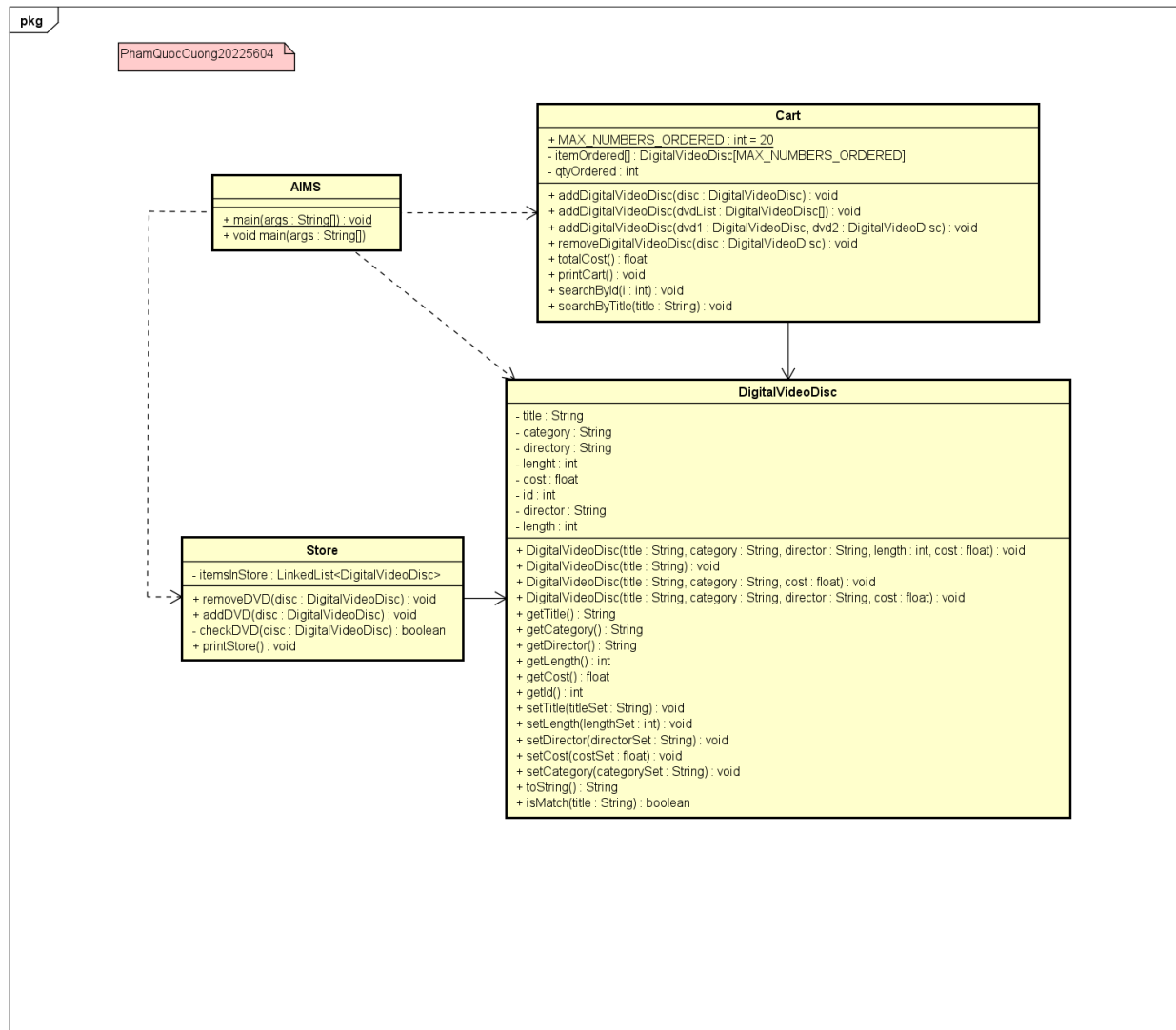


Figure 28 Class Diagram

## Table of Figures:

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList) .....	3
Figure 2 method addDigitalVideoDisc which allows to pass an arbitrary number of arguments for dvd .....	3
Figure 3 Aims Class.....	5
Figure 4 Result .....	5
Figure 5 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList) .....	6
Figure 6 Aims Class.....	6
Figure 7 Result .....	7
Figure 8 Passing parameter code .....	8
Figure 9 Passing parameter result.....	9
Figure 10 A swap() method that can correctly swap the two objects .....	10
Figure 11 Result .....	10
Figure 12 Class member and Instance member.....	12
Figure 13 Result .....	13
Figure 14 Code method to print the content in Cart .....	14
Figure 15 Code method search by id .....	15
Figure 16 Code method search by title .....	15
Figure 17 IsMatch() in DigitalVideoDisc class .....	16
Figure 18 CartTest class .....	16
Figure 19 Result .....	17
Figure 20 Store Class .....	18
Figure 21 Store Test.....	19
Figure 22 Result .....	19
Figure 23 So sánh String và StringBuilder.....	20
Figure 24 GarbageCreator class .....	21
Figure 25 Result: program hangs.....	21
Figure 26 NoGarbage class.....	21
Figure 27 Result: Program finishes quickly .....	22
Figure 28 Class Diagram.....	23