

BÁO CÁO THỰC HÀNH LAP 3 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Nguyễn Thị Minh Châu - 20214997

Mục lục

1. Working with method overloading.....	2
1.1. Overloading by differing types of parameter.....	2
1.2. Overloading by differing the number of parameters	2
2. Passing parameter	2
Result:	3
3. Classifier Member and Instance Member.....	4
Result:	6
4. Open the Cart class.....	6
Code:.....	6
Result:	8
5. Implement the Store class	9
Code:.....	9
Result:	12
6. String, StringBuilder and StringBuffer.....	12
Code:.....	12
Result:	13
7. Answer the Question	13
8. Class Diagram, UseCase Diagram	14

1. Working with method overloading

1.1. Overloading by differing types of parameter

```
55 @ v public void addDigitalVideoDisc(DigitalVideoDisc []dvdList) {  
56     // If cart is full  
57     v if (qtyOrdered + dvdList.length > 20) {  
58         System.out.println("The cart is almost full");  
59         return;  
60     }  
61  
62     // Add to cart  
63     System.arraycopy(dvdList, srcPos: 0, itemsOrdered, qtyOrdered, dvdList.length);  
64  
65     // Increase the qtyOrdered  
66     qtyOrdered += dvdList.length;  
67  
68     // Notify  
69     System.out.println("The list has been added");  
70 }
```

1.2. Overloading by differing the number of parameters

```
73 public void addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2) {  
74     // If cart is full  
75     if (qtyOrdered >= 19) {  
76         System.out.println("The cart is almost full");  
77         return;  
78     }  
79  
80     // Increase the qtyOrdered  
81     qtyOrdered += 2;  
82  
83     // Add to cart  
84     itemsOrdered[qtyOrdered - 2] = dvd1;  
85     itemsOrdered[qtyOrdered - 1] = dvd1;  
86  
87     // Notify  
88     System.out.println("The disc has been added");  
89 }
```

2. Passing parameter

```

1 package lab02.AimsProject;
2
3 public class TestPassingParameter {
4     6 usages
5     static class DVDWrapper {
6         7 usages
7         DigitalVideoDisc disc;
8
9         2 usages
10        DVDWrapper(DigitalVideoDisc disc) {
11            this.disc = disc;
12        }
13    }
14
15    public static void main(String[] args) {
16        // TODO Auto-generated method stub
17        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title: "Jungle");
18        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title: "Cinderella");
19        DVDWrapper jungleDVDWrapper = new DVDWrapper(jungleDVD);
20        DVDWrapper cinderellaDVDWrapper = new DVDWrapper(cinderellaDVD);
21
22        swap(jungleDVDWrapper, cinderellaDVDWrapper);
23        System.out.println("jungle dvd title: " + jungleDVDWrapper.disc.getTitle());
24        System.out.println("cinderella dvd title: " + cinderellaDVDWrapper.disc.getTitle());
25
26        changeTitle(jungleDVD, cinderellaDVD.getTitle());
27        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
28    }
29
30    1 usage
31    public static void swap(DVDWrapper dvd1, DVDWrapper dvd2){
32        DigitalVideoDisc tmp = dvd1.disc;
33        dvd1.disc = dvd2.disc;
34        dvd2.disc = tmp;
35    }
36
37    1 usage
38    public static void changeTitle(DigitalVideoDisc dvd, String title) {
39        String oldTitle = dvd.getTitle();
40        dvd.setTitle(title);
41        dvd = new DigitalVideoDisc(oldTitle);
42    }
43 }

```

Result:

```
jungle dvd title: Cinderella  
cinderella dvd title: Jungle  
jungle dvd title: Cinderella  
  
Process finished with exit code 0
```

3. Classifier Member and Instance Member

Code:

```
5      private int id;  
      3 usages  
6      private String title;  
      3 usages  
7      private String category;  
      3 usages  
8      private String director;  
      3 usages  
9      private int length;  
      3 usages  
10     private double cost;  
11  
      4 usages  
12     private static int nbDigitalVideoDiscs = 0;  
13
```

```
15     public DigitalVideoDisc(String title) {
16         this.title = title;
17         nbDigitalVideoDiscs++;
18         id = nbDigitalVideoDiscs;
19     }
```

```
4  ▶  public static void main(String[] args) {
5      //Create a new cart
6      Cart anOrder = new Cart();
7
8      //Create new dvd objects and add them to the cart
9      DigitalVideoDisc dvd1 = new DigitalVideoDisc ( title: "The Lion King",
10         category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
11      anOrder.addDigitalVideoDisc (dvd1);
12
13      DigitalVideoDisc dvd2 = new DigitalVideoDisc ( title: "Star Wars",
14         category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
15      anOrder.addDigitalVideoDisc (dvd2);
16
17      DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin",
18         category: "Animation", cost: 18.99f);
19      anOrder.addDigitalVideoDisc (dvd3);
20
21      //print number of dvd in cart
22      System.out.println("Number of dvd is: ");
23      System.out.println(DigitalVideoDisc.getNbDigitalVideoDiscs());
24
25      //print id of dvd3
26      System.out.println("Id of dvd3 is: ");
27      System.out.println(dvd3.getId());
28  }
29  }
30  }
```

Result:

```
The disc has been added
The disc has been added
The disc has been added
Number of dvd is:
3
Id of dvd3 is:
3

Process finished with exit code 0
```

4. Open the Cart class

Code:

```
135
136 // Method to print the list of ordered items of a cart,
137 // the price of each item, and the total price
    usage new *
138 public void printCart() {
139     System.out.println("*****CART*****");
140     System.out.println("Ordered Items:");
141     for (DigitalVideoDisc dvd : itemsOrdered) {
142         if (dvd != null)
143             dvd.printDVD();
144     }
145     System.out.println("Total cost: " + totalCost());
146     System.out.println("*****");
147 }
148
```

```
149 // Method to search for DVDs in the cart by ID and display the search results
    2 usages new *
150 public void searchByID(int id) {
151     boolean found = false;
152     for (DigitalVideoDisc dvd: itemsOrdered) {
153         if (dvd != null && dvd.getId() == id) {
154             found = true;
155             dvd.printDVD();
156         }
157     }
158     if (!found) {
159         System.out.println("Not found!");
160     }
161 }
162
163 // Method to search for DVDs in the cart by title and print the results.
    2 usages new *
164 public void searchByTitle(String title) {
165     boolean found = false;
166     for (DigitalVideoDisc dvd: itemsOrdered) {
167         if (dvd != null && dvd.isMatch(title)) {
168             found = true;
169             dvd.printDVD();
170         }
171     }
172     if (!found) {
173         System.out.println("Not found!");
174     }
175 }
176 }
```

```

36
37 // Method to finds out if the corresponding disk is a match given the title.
   1 usage new *
38 @ public boolean isMatch(String title) {
39     return title.equals(this.title);
40 }
41
42 // Method to print a dvd
   3 usages new *
43 public void printDVD() {
44     System.out.println(id + ". DVD - "
45         + title + " - "
46         + category + " - "
47         + director + " - "
48         + length + ": "
49         + cost + "$");
50 }

```

```

1 package lab02.AimsProject;
2
3 public class CartTest {
4     public static void main(String[] args) {
5         //Create a new cart
6         Cart cart = new Cart();
7         //Create new dvd objects and add them to the cart
8         DigitalVideoDisc dvd1 = new DigitalVideoDisc ( title: "The Lion King",
9             category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
10        cart.addDigitalVideoDisc (dvd1);
11        DigitalVideoDisc dvd2 = new DigitalVideoDisc ( title: "Star Wars",
12            category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
13        cart.addDigitalVideoDisc (dvd2);
14        DigitalVideoDisc dvd3 = new DigitalVideoDisc ( title: "Aladin",
15            category: "Animation", cost: 18.99f);
16        cart.addDigitalVideoDisc (dvd3);
17        //Test the print method
18        cart.printCart();
19        //To-do: Test the search methods here
20        cart.searchByID(1);
21        cart.searchByID(4);
22        cart.searchByTitle("Star Wars");
23        cart.searchByTitle("Harry Potter");
24    }
25 }

```

Result:


```
The disc has been added
The disc has been added
The disc has been added
*****CART*****
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.950000762939453$
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.950000762939453$
3. DVD - Aladin - Animation - null - 0: 18.989999771118164$
Total cost: 63.89
*****
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.950000762939453$
Not found!
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.950000762939453$
Not found!

Process finished with exit code 0
```

5. Implement the Store class

Code:

```
1 package lab02.AimsProject;
2
3 import java.util.ArrayList;
4
5 2 usages
6 public class Store {
7     // Attribute
8     5 usages
9     private ArrayList<DigitalVideoDisc> itemsInStore = new ArrayList<>();
10
11     // Constructor
12     1 usage
13     public Store() {
14     }
15
16     // Method to add a dvd
17     3 usages
18     public void addDVD(DigitalVideoDisc disc) {
19         // Add to store
20         itemsInStore.add(disc);
21
22         // Notify
23         System.out.println("The disc has been added");
24     }
25
26     // Method to remove a dvd
27     2 usages
28     public void removeDVD(DigitalVideoDisc disc) {
29         // Search for disc
30         int indexOfRemoved = itemsInStore.indexOf(disc);
31
32         // If not found
33         if (indexOfRemoved == -1) {
34             System.out.println("The disc is not found");
35         }
36         return;
37     }
38 }
```

```
31     }
32
33     // Remove
34     itemsInStore.remove(indexOfRemoved);
35
36     // Notify
37     System.out.println("The disc has been removed");
38 }
39
40 // Getter and Setter
no usages
41 public ArrayList<DigitalVideoDisc> getItemsInStore() {
42     return itemsInStore;
43 }
44
no usages
45 public void setItemsInStore(ArrayList<DigitalVideoDisc> itemsInStore) {
46     this.itemsInStore = itemsInStore;
47 }
48 }
49
```

```
1 package lab02.AimsProject;
2
3 public class StoreTest {
4     public static void main(String[] args) {
5         //Create a new cart
6         Store store = new Store();
7
8         //Create new dvd objects and add them to the cart
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc ( title: "The Lion King",
10             category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
11         store.addDVD(dvd1);
12
13         DigitalVideoDisc dvd2 = new DigitalVideoDisc ( title: "Star Wars",
14             category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
15         store.addDVD(dvd2);
16
17         DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin",
18             category: "Animation", cost: 18.99f);
19         store.addDVD(dvd3);
20
21         // Remove success
22         store.removeDVD(dvd1);
23
24         // Remove failed
25         DigitalVideoDisc dvd4 = new DigitalVideoDisc( title: "Harry Potter",
26             category: "Fiction", cost: 18.99f);
27         store.removeDVD(dvd4);
28     }
29 }
30
```

Result:

```
The disc has been added
The disc has been added
The disc has been added
The disc has been removed
The disc is not found

Process finished with exit code 0
```

6. String, StringBuilder and StringBuffer

Code:

```
1 package lab03.OtherProjects.Garbage;
2
3 import java.util.*;
4
5 public class ConcatenationInLoops {
6     public static void main(String[] args) {
7         Random r = new Random( seed: 123);
8         long start = System.currentTimeMillis();
9         String s = "";
10        for (int i = 0; i < 65536; i++)
11            s += r.nextInt( bound: 2);
12        System.out.println(System.currentTimeMillis() - start); // This prints roughly 4500.
13        r = new Random ( seed: 123);
14        start = System.currentTimeMillis();
15        StringBuilder sb = new StringBuilder();
16        for (int i = 0; i < 65536; i++)
17            sb.append(r.nextInt( bound: 2));
18        s = sb.toString();
19        System.out.println(System.currentTimeMillis() - start); // This prints 5.
20    }
21 }
```

Result:

487

10

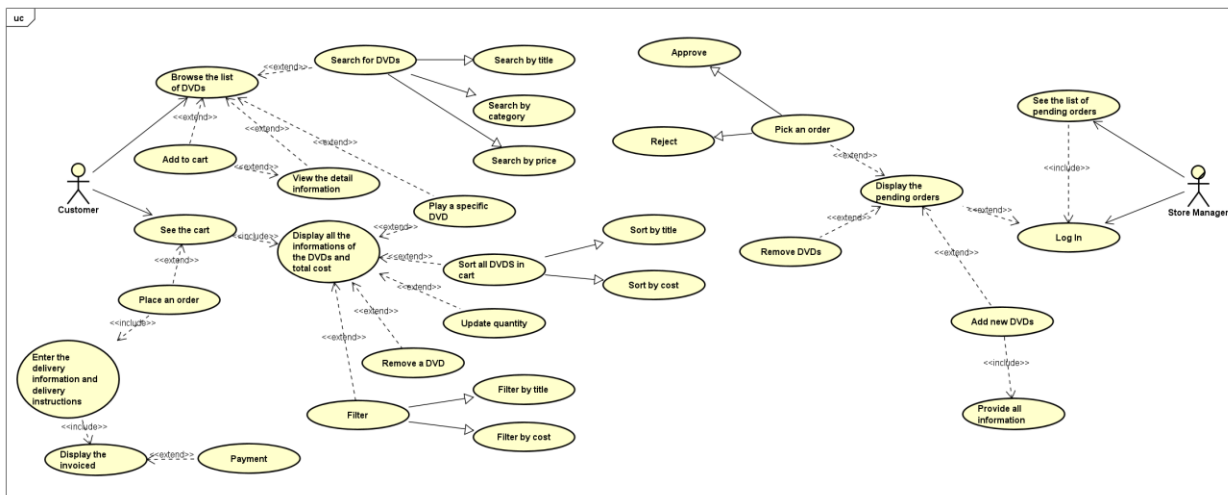
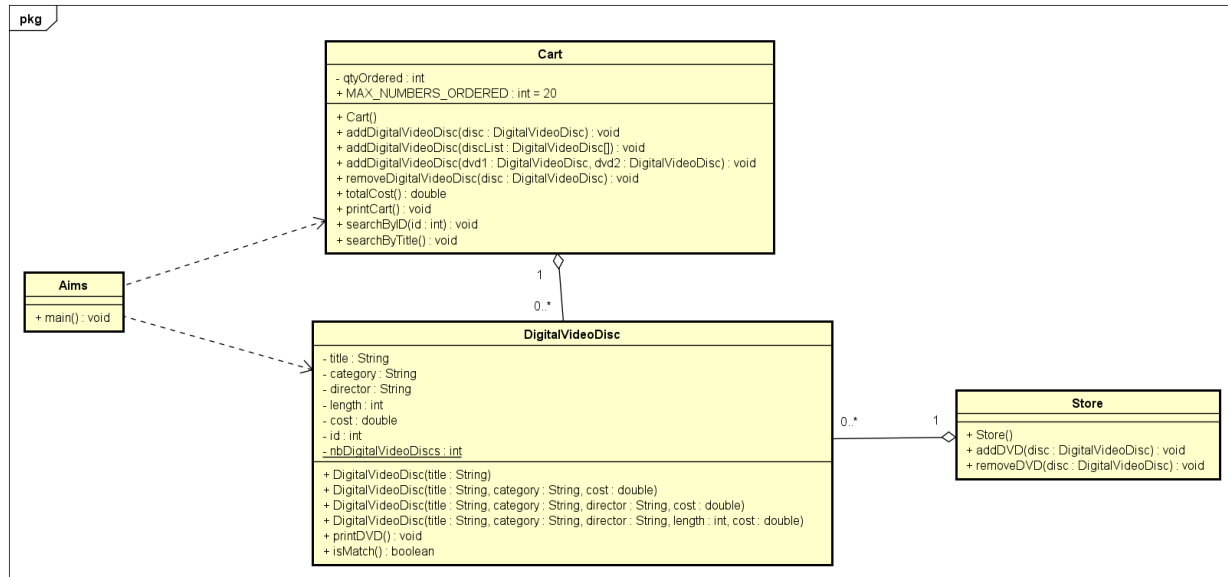
Process finished with exit code 0

7. Answer the Question

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

=> JAVA is a Pass by Value programming language.

8. Class Diagram, UseCase Diagram



Link git: <https://github.com/mchau98/IT3103.732871.2023.1.4997.NguyenThiMinhChau>