BÁO CÁO THỰC HÀNH LAB 3
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục nội dung

Mục lục hình ảnh

1. Working with method overloading

1.1 Overloading by differing types of parameter

```
54      // Method to add list new DVDs
        no usages  Dl-huyentm *
55  @   public void addDigitalVideoDisc(DigitalVideoDisc []dvdList) {
56          // If cart is full
57          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      }
```

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

1.2 Overloading by differing the number of parameters

```
72      // Method to add two new DVD
        no usages  new *
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      }
```

Figure 2 Method addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2)

2. Passing parameter

Code:

```java
package lab02.AimsProject;

public class TestPassingParameter {
    6 usages
    static class DVDWrapper {
        7 usages
        DigitalVideoDisc disc;

        2 usages
        DVDWrapper(DigitalVideoDisc disc) {
            this.disc = disc;
        }
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "Cinderella");
        DVDWrapper jungleDVDWrapper = new DVDWrapper(jungleDVD);
        DVDWrapper cinderellaDVDWrapper = new DVDWrapper(cinderellaDVD);

        swap(jungleDVDWrapper, cinderellaDVDWrapper);
        System.out.println("jungle dvd title: " + jungleDVDWrapper.disc.getTitle());
        System.out.println("cinderella dvd title: " + cinderellaDVDWrapper.disc.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
    }
    1 usage
    public static void swap(DVDWrapper dvd1, DVDWrapper dvd2){
        DigitalVideoDisc tmp = dvd1.disc;
        dvd1.disc = dvd2.disc;
        dvd2.disc = tmp;
    }

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

Figure 3 Passing parameter code

Result:

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

Process finished with exit code 0
```

Figure 4 Passing parameter result

3. Classifier Member and Instance Member

Code:

```java
package lab03.AimsProject;

1 usage  new *
public class DigitalVideoDisc {
    // Attribute
    4 usages
    private int id;
    5 usages
    private String title;
    4 usages
    private String category;
    4 usages
    private String director;
    4 usages
    private int length;
    4 usages
    private double cost;

    4 usages
    private static int nbDigitalVideoDiscs = 0;

    // Constructor
    2 usages  new *
    public DigitalVideoDisc(String title) {
        this.title = title;
        nbDigitalVideoDiscs++;
```

DigitalVideoDisc.java

```java
        this.title = title;
        nbDigitalVideoDiscs++;
        id = nbDigitalVideoDiscs;
    }

    2 usages  new *
    public DigitalVideoDisc(String title, String category, double cost) {
        this(title);
        this.category = category;
        this.cost = cost;
    }

    2 usages  new *
    public DigitalVideoDisc(String title, String category, String director, double cost) {
        this(title, category, cost);
        this.director = director;
    }

    1 usage  new *
    public DigitalVideoDisc(String title, String category, String director, int length, double
        this(title, category, director, cost);
        this.length = length;
    }

    // Method to print a dvd
    no usages  new *
```

```java
    // Method to print a dvd
    no usages  new *
    public void printDVD() {
        System.out.println(id + ". DVD - "
                + title + " - "
                + category + " - "
                + director + " - "
                + length + ": "
                + cost + "$");
    }

    // Method to finds out if the corresponding disk is a match given the title.
    no usages  new *
    public boolean isMatch(String title) {
        return title.equals(this.title);
    }

    // Getter and Setter
    no usages  new *
    public String getTitle() {
        return title;
    }
```

```
55      }                                                          1 18
56
        no usages  new *
57      public String getCategory() {
58          return category;
59      }
60
        no usages  new *
61      public String getDirector() {
62          return director;
63      }
64
        no usages  new *
65      public int getLength() {
66          return length;
67      }
68
        no usages  new *
69      public double getCost() {
70          return cost;
71      }
72
        no usages  new *  1 related problem
73      public void setTitle(String title) {
74          this.title = title;
75      }
```

```
76
        no usages  new *
77      public void setCategory(String category) {
78          this.category = category;
79      }
80
        no usages  new *
81      public void setDirector(String director) {
82          this.director = director;
83      }
84
        no usages  new *
85      public void setLength(int length) {
86          this.length = length;
87      }
88
        no usages  new *
89      public void setCost(double cost) {
90          this.cost = cost;
91      }
92
        no usages  new *
93      public int getId() {
94          return id;
```

```
95      }
96
        no usages  new *
97      public void setId(int id) {
98          this.id = id;
99      }
100
        no usages  new *
101     public static int getNbDigitalVideoDiscs() {
102         return nbDigitalVideoDiscs;
103     }
104
        no usages  new *
105     public static void setNbDigitalVideoDiscs(int nbDigitalVideoDiscs) {
106         DigitalVideoDisc.nbDigitalVideoDiscs = nbDigitalVideoDiscs;
107     }
108 }
```

Figure 5 Classifier Member and Instance Member Code

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
```

Figure 6 Classifier Member and Instance Member Result

4. Open the Cart class

Code:

```java
135
136     // Method to print the list of ordered items of a cart,
137     // the price of each item, and the total price
        1 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
```

```java
149     // Method to search for DVDs in the cart by ID and display the search resul
        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 }
```

Figure 7 Method in class Cart to print the list

```
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      }
```

Figure 8 Method in class DigitalVideoDisc to check title and print a dvd

```
1   package lab02.AimsProject;                                          2
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  }
```

Figure 9 Test code

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
```

Figure 10 Result Open the Cart class

5. Implement the Store class

Code:

```java
package lab02.AimsProject;

import java.util.ArrayList;

2 usages
public class Store {
    // Attribute
    5 usages
    private ArrayList<DigitalVideoDisc> itemsInStore = new ArrayList<>();

    // Constructor
    1 usage
    public Store() {
    }

    // Method to add a dvd
    3 usages
    public void addDVD(DigitalVideoDisc disc) {
        // Add to store
        itemsInStore.add(disc);

        // Notify
        System.out.println("The disc has been added");
    }

    // Method to remove a dvd
    2 usages
    public void removeDVD(DigitalVideoDisc disc) {
        // Search for disc
        int indexOfRemoved = itemsInStore.indexOf(disc);

        // If not found
        if (indexOfRemoved == -1) {
            System.out.println("The disc is not found");
            return;
```

Figure 11 Code class Store

```java
package lab02.AimsProject;

public class StoreTest {
    public static void main(String[] args) {
        //Create a new cart
        Store store = new Store();

        //Create new dvd objects and add them to the cart
        DigitalVideoDisc dvd1 = new DigitalVideoDisc ( title: "The Lion King",
                category: "Animation", director: "Roger Allers", length: 87, cost: 19.95f);
        store.addDVD(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc ( title: "Star Wars",
                category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
        store.addDVD(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin",
                category: "Animation", cost: 18.99f);
        store.addDVD(dvd3);

        // Remove success
        store.removeDVD(dvd1);

        // Remove failed
        DigitalVideoDisc dvd4 = new DigitalVideoDisc( title: "Harry Potter",
                category: "Fiction", cost: 18.99f);
        store.removeDVD(dvd4);
    }
}
```

Figure 12 Code test class Store

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
```

Figure 13 Result Implement the Store class

6. String, StringBuilder and StringBuffer

Code:

```java
package lab03.OtherProjects.Garbage;

import java.util.*;

public class ConcatenationInLoops {
    public static void main(String[] args) {
        Random r = new Random( seed: 123);
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i++)
            s += r.nextInt( bound: 2);
        System.out.println(System.currentTimeMillis() - start); // This prints roughly 4500.
        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); // This prints 5.
    }
}
```

Figure 14 Code ConcatenationInLoops

Result:

```
487
10
```

Figure 15 Result ConcatenationInLoops

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