

LAB 3 REPORT

1. Working with method overloading

1.1 Overloading by differing types of parameter

```

public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList)
{
    // add to cart
    for(DigitalVideoDisc disc : dvdList)
    {
        if(qtyOrdered>20) {
            System.out.println("The cart is full ");
            return ;
        }
        else {
            itemOrdered[qtyOrdered]=disc;
            qtyOrdered++;
            System.out.println(disc.getTitle()+" has been added!");
        }
    }
}

```

1.2 Overloading by differing the number of parameters

```

84
85 public void addDigitalVideoDisc( DigitalVideoDisc dvd1 , DigitalVideoDisc dvd2 )
86 {
87     if(qtyOrdered>20)
88     {
89         System.out.println("The cart is full");
90         return ;
91     }
92
93     qtyOrdered+=2;
94     itemOrdered[qtyOrdered-2]=dvd1;
95     itemOrdered[qtyOrdered-1]=dvd2;
96     System.out.println("The list has been added");
97 }
98 }
99
100

```

- Implementations of new methods:

These methods will add a list of DVDs to the current cart. Please note that we cannot compile both because they have the same

name and signature (one list as a parameter)

```

17
18 Cart anOrder = new Cart();
19 DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
20     "Animation", "Roger Allers", 87, 19.95f);
21 DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
22     "Science Fiction", "George Lucas", 87, 24.95f);
23 DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
24     "Animation", 18.99f);
25 DigitalVideoDisc [] dvdList = new DigitalVideoDisc[3];
26 dvdList[0] = new DigitalVideoDisc("The abc", "Muc King", 14.2f);
27 dvdList[1] = new DigitalVideoDisc("The def", "Lee King", 18.5f);
28 dvdList[2] = new DigitalVideoDisc("the xyz", "Yasuo King", 20.4f);
29
30 anOrder.addDigitalVideoDisc(dvdList);
31 anOrder.addDigitalVideoDisc(dvd1, dvd2);
32
33 | System.out.println(anOrder.toString());
34
35 System.out.println("Total cost is: "+anOrder.totalCost());
36 }
37 }

```

```

<terminated> Aims (1) [Java Application] C:\Users\PC_KyAnh\.p2\po
The list has been adder 2dvd!
Title: The abc, Cost: $14.2
Title: The def, Cost: $18.5
Title: the xyz, Cost: $20.4
Title: The Lion King, Cost: $19.95
Title: Star Wars, Cost: $24.95

Total cost is: 98.0

```

2. Passing parameter

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

Java là ngôn ngữ lập trình Pass-by-Value (Truyền tham trị).

```

2 public class testPassingParameter {
3 public static void main(String[] args) {
4     DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
5     DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
6     swap(jungleDVD, cinderellaDVD);
7     System.out.println("Jungle dvd title: "+jungleDVD.getTitle());
8     System.out.println("cinderella dvd title: "+cinderellaDVD.getTitle());
9     changeTitle(jungleDVD, cinderellaDVD.getTitle());
10    System.out.println("Jung le dvd title: "+jungleDVD.getTitle());
11
12
13 }
14 public static void swap (Object o1 , Object o2)
15 {
16     Object tmp = o1;
17     o1 = o2;
18     o2 = tmp;
19 }
20 public static void changeTitle(DigitalVideoDisc dvd , String title)
21 {
22     String oldTitle = dvd.getTitle();
23     dvd.setTitle(title);
24     dvd = new DigitalVideoDisc (oldTitle);
25 }
26 }
27

```

```

Jungle dvd title: Jungle
cinderella dvd title: Cinderella
Jung le dvd title: Cinderella

```

Questions:

- After the call of `swap(jungleDVD, cinderellaDVD)` why does the title of these two objects still remain?
 - ➔ Sau khi gọi `swap(jungleDVD, cinderellaDVD)`, tiêu đề của hai đối tượng vẫn không thay đổi vì Java sử dụng cơ chế “pass-by-value” (truyền tham số theo giá trị) khi xử lý các tham số, kể cả với các đối tượng. Điều này có nghĩa là hàm `swap` chỉ hoán đổi các bản sao của tham chiếu, chứ không phải bản thân các tham chiếu gốc. Do đó, vị trí của các đối tượng ban đầu không bị thay đổi.
- After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())` why is the title of the `JungleDVD` changed?

- Khi phương thức `changeTitle(jungleDVD, cinderellaDVD.getTitle())` được gọi, tiêu đề của đối tượng mà tham chiếu `jungleDVD` trỏ tới sẽ thay đổi, vì phương thức này trực tiếp tác động lên đối tượng đó. Câu lệnh `dvd = new DigitalVideoDisc(oldTitle)` tạo ra một đối tượng DVD mới với tiêu đề cũ. Việc này không ảnh hưởng đến đối tượng DVD ban đầu, vì sau đó biến tham chiếu `dvd` không còn liên kết với đối tượng gốc nữa.
- **A swap method that can correctly swap the two objects**

```

14
15 public static void swap (DigitalVideoDisc o1, DigitalVideoDisc o2)
16 {
17     String tmp = o1.getTitle();
18     o1.setTitle(o2.getTitle());
19     o2.setTitle(tmp);
20 }
21
terminated: test using parameter pass
Jungle dvd title: Cinderella
cinderella dvd title: Jungle
Jung le dvd title: Jungle

```

3. Classifier Member and Instance Member

```

2 public class DigitalVideoDisc {
3     private String title;
4     private String category;
5     private String director;
6     private int length;
7     private float cost;
8     private static int nbDigitalVideoDiscs = 0;
9     private int id;
10
11     public DigitalVideoDisc(String title) {
12         super();
13         this.title = title;
14         this.id = nbDigitalVideoDiscs++;
15     }
16
17     public DigitalVideoDisc(String title, String category, float cost) {
18         super();
19         this.title = title;
20         this.category = category;
21         this.cost = cost;
22         this.id = nbDigitalVideoDiscs++;
23     }
24
25     public DigitalVideoDisc(String title, String category, String director) {
26         super();
27         this.title = title;
28         this.category = category;
29         this.director = director;
30     }
31 }

```

- Hình trên minh họa việc tạo một thuộc tính lớp có tên là “nbDigitalVideoDiscs” và một thuộc tính instance gọi là “id” trong lớp `DigitalVideoDisc`. Mỗi khi một đối tượng của

lớp DigitalVideoDisc được khởi tạo, giá trị của nbDigitalVideoDiscs sẽ được cập nhật và ID sẽ được gán cho đối tượng đó.

4. Open the Cart class

- Create a new method to print the list of ordered items of a cart, the price of each item, and the total price. Format the outline as below:

```
06● public void printCART()
07 {
08     System.out.println("*****CART*****");
09     System.out.println("Ordered Items:");
10     System.out.println(toString());
11     System.out.println("Total cost: $" + totalCost());
12     System.out.println("*****");
13 }
14 }
15
```

```
74● public String toString()
75 {
76     return String.format(getId()+" - "+getTitle()+" - " + getCategory()+" - "+ getDirector()
77         + " - " + getLength()+" - " + getCost());
78 }
79 }
80
```

```
*****CART*****
Ordered Items:
3. The abc - Muc King - null - 0 - 14.2
4. The def - Lee King - null - 0 - 18.5
5. the xyz - Yasuo King - null - 0 - 20.4
0. The Lion King - Animation - Roger Allers - 87 - 19.95
1. Star Wars - Science Fiction - George Lucas - 87 - 24.95

Total cost: $98.0
*****
```

Suggestion: Write a `toString()` method for the `DigitalVideoDisc` class. What should be the return type of this method?

➔ Kiểu trả về của hàm `toString()` nên là chuỗi.

- Search methods and `isMatch` method

```
1 public void searchCart(int id)
2 {
3     boolean search = false;
4     for(int i=0;i<qtyOrdered;i++)
5     {
6         if(itemOrdered[i].isMatch(id-1))
7         {
8             search = true;
9             System.out.println("Found item with ID: "+id);
10            System.out.println(itemOrdered[i].toString());
11            break;
12        }
13    }
14    if(search==false) {
15        System.out.println("Not found!");
16    }
17 }
```

```
2 public void searchCart(String title)
3 {
4     boolean search = false;
5     for(int i=0;i<qtyOrdered;i++)
6     {
7         if(itemOrdered[i].isMatch(title))
8         {
9             search = true;
10            System.out.println("Found item with title included : "+title);
11            System.out.println(itemOrdered[i].toString());
12            break;
13        }
14    }
15    if(search==false)
16        System.out.println("Not found!");
17 }
18 }
19 }
20 }
```

```

    public boolean isMatch(int id)
    {
        return this.id == id;
    }
    public boolean isMatch(String title)
    {
        return this.title.toLowerCase().contains(title.toLowerCase());
    }
}

```

```

1
2 public class testCart {
3 public static void main(String[] args) {
4     Cart cart= new Cart();
5     DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
6         "Animation","Roger Allers",87,19.95f);
7     cart.addDigitalVideoDisc(dvd1);
8     DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
9         "Science Fiction","George Lucas",87,24.95f);
10    cart.addDigitalVideoDisc(dvd2);
11    DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
12        "Animation",18.99f);
13    cart.addDigitalVideoDisc(dvd3);
14
15    cart.printCART();
16    cart.searchCart("Aladin");
17    cart.searchCart(1);
18
19 }
20 }

```

Console × Problems Debug Shell

<terminated> testCart [Java Application] C:\Users\PC_KyAnh\p2\pool\plugins\org.eclipse.justj.openjdk

The Lion King Has been added!
 Star Wars Has been added!
 Aladin Has been added!

*****CART*****

Ordered Items:

1. The Lion King - Animation - Roger Allers - 87 - 19.95
2. Star Wars - Science Fiction - George Lucas - 87 - 24.95
3. Aladin - Animation - null - 0 - 18.99

Total cost: \$63.89

Found item with title included : Aladin

3. Aladin - Animation - null - 0 - 18.99

Found item with ID: 1

1. The Lion King - Animation - Roger Allers - 87 - 19.95

5. Implement the Store class

```

2 public class Store {
3     private DigitalVideoDisc itemsInStore[];
4     private int numberOfItems;
5     public Store(int fullsize)
6     {
7         itemsInStore = new DigitalVideoDisc[fullsize];
8         numberOfItems=0;
9     }
10    public void addDVD(DigitalVideoDisc dvd)
11    {
12        if(numberOfItems<itemsInStore.length) {
13            itemsInStore[numberOfItems] = dvd;
14            numberOfItems++;
15            System.out.println("ADDED: "+dvd.getTitle());
16        }
17        else {
18            System.out.println("The DVDs Store is full!");
19        }
20    }
21    public void removeDVD(DigitalVideoDisc dvd)
22    {
23        boolean check = false;
24        for(int i=0;i<numberOfItems;i++)
25        {
26            if(itemsInStore[i].equals(dvd))
27            {
28                check = true;
29                for(int j=i;j<numberOfItems-1;j++)
30                {
31                    itemsInStore[j]=itemsInStore[j+1];
32                }
33                numberOfItems--;
34                System.out.println("REMOVE: "+dvd.getTitle());
35                break;
36            }
37        }
38        if(check == false)
39        {
40            System.out.println("DVD not found: "+dvd.getTitle());
41        }
42    }
43    }
44    public void displayStore()
45    {
46        System.out.println("*****Store*****");
47        for(int i=0;i<numberOfItems;i++)
48        {
49            System.out.println(itemsInStore[i].toString());
50        }
51        System.out.println("*****");
52    }
53    }
54    }

```



```
2 public class StoreTest {
3 public static void main(String[] args) {
4     Store store = new Store(5);
5     DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
6         "Animation", "Roger Allers", 87, 19.95f);
7     DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
8         "Science Fiction", "George Lucas", 87, 24.95f);
9     DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
10         "Animation", 18.99f);
11     store.addDVD(dvd1);
12     store.addDVD(dvd2);
13     store.addDVD(dvd3);
14     store.displayStore();
15     store.removeDVD(dvd2);
16     store.displayStore();
17 }
18 }
19
```

Console ×

<terminated> StoreTest [Java Application] C:\Users\PC_KyAnh\p2\pool\plugins\org.eclipse.justj.c

ADDED: The Lion King

ADDED: Star Wars

ADDED: Aladin

*****Store*****

1. The Lion King - Animation - Roger Allers - 87 - 19.95

2. Star Wars - Science Fiction - George Lucas - 87 - 24.95

3. Aladin - Animation - null - 0 - 18.99

REMOVE: Star Wars

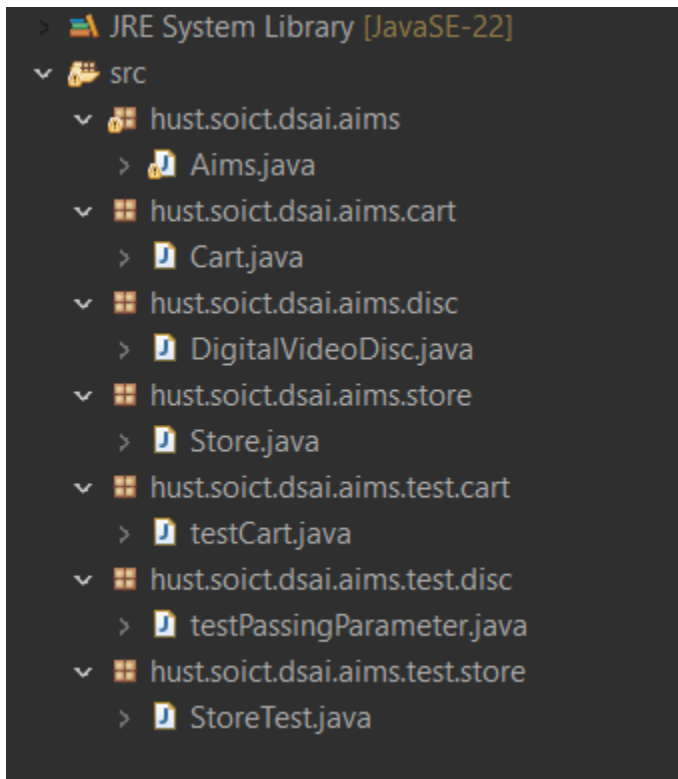
*****Store*****

1. The Lion King - Animation - Roger Allers - 87 - 19.95

3. Aladin - Animation - null - 0 - 18.99

6. Re-organize your projects

Rename project, use packages and re-organize hands-on labs and exercises.



6.1 *String, StringBuilder and StringBuffer*

```
1 import java.util.Random;
2 public class ConcatenationInLoops {
3     public static void main(String[] args) {
4         Random r = new Random(seed:123);
5         long start = System.currentTimeMillis();
6         String s="";
7         for(int i=0;i< 65536;i++)
8         {
9             s+=r.nextInt(bound:2);
10        }
11        System.out.println(System.currentTimeMillis()-start);
12
13        r = new Random(seed:123);
14        start = System.currentTimeMillis();
15        StringBuilder sb = new StringBuilder();
16        for(int i = 0;i<65536;i++)
17        {
18            sb.append(r.nextInt(bound:2));
19        }
20
21        s=sb.toString();
22        System.out.println(System.currentTimeMillis()-start);
23    }
24 }
25
```