

Report lab03

1, New written codes&Answer the question

```
public int addDigitalVideoDisc(DigitalVideoDisc... dvdArray) {
    int addCount = 0;
    for (DigitalVideoDisc disc : dvdArray) {
        if (qtyOrdered == MAX_NUMBERS_ORDERED) {
            System.out.println("The cart is almost full. Can't add more discs");
            break;
        } else {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD " + '\'' + disc.getTitle() + '\'' + " has been added!");
            addCount++;
        }
    }
    return addCount;
}

//Ham them 2 dia DVD
public int addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    if (qtyOrdered + 1 >= MAX_NUMBERS_ORDERED) {
        System.out.println("The cart is almost full. Can't add more discs");
        return 0;
    } else {
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered++;
        System.out.println("The DVD " + '\'' + dvd1.getTitle() + '\'' + " has been added!");

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

        return 2;
    }
}
```

addDigitalVideoDisc method overloading

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

```
package hust.soict.dsai.test.disc;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;;

public class TestPassingParameter {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("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(Object o1, Object o2) {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

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

TestPassingParameter class

- After the call of `swap(jungleDVD, cinderellaDVD)`, the title value is passed inside the method as a new Object not have any related to the jungle and Cinderella in the main method. So that the value is not changed
- After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())`, the address of the jungleDVD is passed to the `changeTitle` as a `DigitalVideoDisc` object type
→ changing the value inside that address means changing the value in `jungleDVD` as well.

→ By default, Java is always a pass-to-value programming language.

```
package hust.soict.dsai.aims.disc;

public class DigitalVideoDisc {
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;
    private static int nbDigitalVideoDiscs = 0;
    private int id;

    public DigitalVideoDisc(String title) {
        super();
        this.title = title;
        this.id = ++nbDigitalVideoDiscs;
    }

    public DigitalVideoDisc(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }

    public DigitalVideoDisc(String title, String category, String director, float cost) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }

    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 = ++nbDigitalVideoDiscs;
    }
}
```

Adding id attribute and nbDigitalVideoDiscs updated.

```
public void print() {
    StringBuilder output = new StringBuilder("*****CART***** \nOrdered items: \n");
    for (int i = 0; i < qtyOrdered; i++) {
        output.append(i+1 + ".[" + itemsOrdered[i].getTitle() + "] - [" + itemsOrdered[i].getCategory() + "] - ["
            + itemsOrdered[i].getDirector() + "] - [" + itemsOrdered[i].getLength() + "]: "
            + itemsOrdered[i].getCost() + " $\n");
    }
    output.append("total: ").append(totalCost()).append(" $\n");
    output.append("*****\n");
    System.out.println(output);
}
```

Print list of ordered item in the cart

```

1 package hust.soict.dsai.test.cart;
2 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
3
4
5
6 public class CartTest{
7     public static void main(String[] args) {
8         Cart cart = new Cart();
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King","Animation",
10             "Roger Allers",87,19.95f);
11         cart.addDigitalVideoDisc(dvd1);
12         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars","Science Fiction",
13             "Geogre Lucas",87,24.95f);
14         cart.addDigitalVideoDisc(dvd2);
15         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin","Animation",18.99f);
16         cart.addDigitalVideoDisc(dvd3);
17
18         cart.print();
19
20         //Test search by ID method
21         cart.searchById(3);
22         cart.searchById(4);
23
24         //Test search by Title method
25         cart.searchByTitle("The Lion King");
26         cart.searchByTitle("Alan Walker");
27     }
28 }

```

CartTest method

```

The DVD "The Lion King" have been added!
The DVD "Star wars" have been added!
The DVD "Aladin" have been 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: 63.89 $
*****

Result: [Aladin] - [Animation] - [null] - [0]: 18.99 $

No match found !
Result: [The Lion King] - [Animation] - [Roger Allers] - [87]: 19.95 $

No match found !

```

Test result

```

1 package hust.soict.dsai.aims.store;
2 import java.util.LinkedList;
3
4
5
6 public class Store {
7     private LinkedList<DigitalVideoDisc> itemsInStore = new LinkedList<DigitalVideoDisc>();
8
9
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( disc.getTitle() + " 've been deleted from the store !");
23        } else {
24            System.out.println("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( disc.getTitle() + " 've been added to the store !");
32        } else {
33            System.out.println( disc.getTitle() + " 'already exists in the store !");
34        }
35    }
36
37
38    public String toString() {
39        StringBuilder string = new StringBuilder("*****STORE*****\nitems in the store: \n");
40        if (itemsInStore.isEmpty()) string.append("There is no dvd in the store !\n");
41        else {
42            for (DigitalVideoDisc dvd : itemsInStore) {
43                string.append(dvd.getTitle() + " - " + dvd.getCost() + " $\n");
44            }
45        }
46        string.append("*****");
47        return string.toString();
48    }
49 }

```

Store Class

```

1 package hust.soict.dsai.test.store;
2 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
3
4 public class StoreTest {
5     public static void main(String[] args) {
6         Store store = new Store();
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         store.addDVD(dvd1);
13         store.addDVD(dvd2);
14         store.addDVD(dvd3);
15
16         System.out.println(store.toString());
17
18         store.removeDVD(dvd1);
19         store.removeDVD(dvd2);
20         store.removeDVD(dvd3);
21
22         System.out.println(store.toString());
23     }
24 }
25

```

StoreTest Class

```

<terminated> StoreTest [Java Application] C:\Program Fi
The Lion King 've been added to the store !
Star wars 've been added to the store !
Aladin 've been added to the store !
*****STORE*****
items in the store:
The Lion King - 19.95 $
Star wars - 24.95 $
Aladin - 18.99 $
*****
The Lion King 've been deleted from the store !
Star wars 've been deleted from the store !
Aladin 've been deleted from the store !
*****STORE*****
items in the store:
There is no dvd in the store !
*****

```

Test on Store class result

```

1 package hust.soict.dsai.garbage;
2 import java.io.IOException;
3
4
5
6 public class GarbageCollector {
7
8     public static void main(String[] args) throws IOException {
9         String filename = "text.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

```

GarbageCollector Class

```

package hust.soict.dsai.garbage;
import java.util.Random;

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

        r = new Random(123);
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 65536; i++) {
            sb.append(r.nextInt(2));
        }
        s = sb.toString();
        System.out.println(System.currentTimeMillis() - start); // This prints 5.
    }
}

```

ConcatenationInLoop class

2, Updated Use-case diagram and Class Diagram



