

Họ và tên : Lê Nguyễn Minh Đức

MSSV : 20225611

- Method overloading

```
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {  
    for (DigitalVideoDisc disc : dvdList) {  
        if (qtyOrdered < MAX_NUMBER_ORDERED) {  
            itemsOrdered[qtyOrdered] = disc;  
            qtyOrdered++;  
        } else {  
            System.out.println(x:"The cart is full, cannot add more discs.");  
            break;  
        }  
    }  
    System.out.println(x:"The list of discs has been added");  
}
```

```
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {  
    if (qtyOrdered < MAX_NUMBER_ORDERED) {  
        itemsOrdered[qtyOrdered] = dvd1;  
        qtyOrdered++;  
        if (qtyOrdered < MAX_NUMBER_ORDERED) {  
            itemsOrdered[qtyOrdered] = dvd2;  
            qtyOrdered++;  
        } else {  
            System.out.println(x:"The cart is full, cannot add second disc.");  
        }  
    } else {  
        System.out.println(x:"The cart is full, cannot add more discs.");  
    }  
    System.out.println(x:"Two discs have been added");  
}
```

```
public void addDigitalVideoDisc(DigitalVideoDisc... dvdList) {  
    for (DigitalVideoDisc disc : dvdList) {  
        if (qtyOrdered < MAX_NUMBER_ORDERED) {  
            itemsOrdered[qtyOrdered] = disc;  
            qtyOrdered++;  
            System.out.println("Added: " + disc.getTitle());  
        } else {  
            System.out.println("The cart is full. Cannot add: " + disc.getTitle());  
            break;  
        }  
    }  
}
```

- Passing Parameter

```

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

public class TestPassingParameter {
    Run | Debug
    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"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 tmpObject = o1;
        o1 = o2;
        o2 = tmpObject;
    }

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

```

- Classifier Member and Instance Member

```

1 package hust.soict.dsai.aims.disc;
2
3 public class DigitalVideoDisc {
4
5     private static int nbDigitalVideoDiscs = 0;
6
7     private int id;
8     private String title;
9     private String category;
10    private String director;
11    private int length;
12    private float cost;
13
14    public DigitalVideoDisc(String title) {
15        this.title = title;
16        this.id = ++nbDigitalVideoDiscs;
17    }
18
19    public DigitalVideoDisc(String title, String category, float cost) {
20        this.title = title;
21        this.category = category;
22        this.cost = cost;
23        this.id = ++nbDigitalVideoDiscs;
24    }
25
26    public DigitalVideoDisc(String title, String category, String director, float cost) {
27        this.title = title;
28        this.category = category;
29        this.director = director;
30        this.cost = cost;
31        this.id = ++nbDigitalVideoDiscs;
32    }
33
34 }

```

- Print and Search methods

```

public void printCartDetails() {
    System.out.println(x:"*****CART*****");
    System.out.println(x:"Ordered Items:");
    for (int i = 0; i < qtyOrdered; i++) {
        System.out.println((i + 1) + ". " + itemsOrdered[i].toString());
    }
    System.out.println("Total cost: " + totalCost() + " $");
    System.out.println(x:"*****");
}

public void searchById(int id) {
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].getId() == id) {
            System.out.println("Found DVD with ID " + id + ": " + itemsOrdered[i].toString());
            found = true;
            break;
        }
    }
    if (!found) {
        System.out.println("No DVD found with ID " + id);
    }
}

public void searchByTitle(String title) {
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].isMatch(title)) {
            System.out.println("Found DVD with title \"" + title + "\": " + itemsOrdered[i].toString());
            found = true;
        }
    }
    if (!found) {
        System.out.println("No DVD found with title \"" + title + "\"");
    }
}

public float totalCost() {
    float sum = 0.0f;
    for (int i = 0; i < qtyOrdered; i++) {
        sum += itemsOrdered[i].getCost();
    }
    return sum;
}
}

```

- CartTest

```

1 package hust.soict.dsai.test.cart;
2
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4 import hust.soict.dsai.aims.cart.Cart;
5
6 public class CartTest {
7     Run | Debug
8     public static void main(String[] args) {
9
10         Cart cart = new Cart();
11
12         DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:87, cost:19.95f);
13         cart.addDigitalVideoDisc(dvd1);
14
15         DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
16         cart.addDigitalVideoDisc(dvd2);
17
18         DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladdin", category:"Animation", cost:18.99f);
19         cart.addDigitalVideoDisc(dvd3);
20
21         cart.printCartDetails();
22
23         System.out.println(x:"\nSearch by ID:");
24         cart.searchById(id:1);
25         cart.searchById(id:4);
26
27         System.out.println(x:"\nSearch by title:");
28         cart.searchByTitle(title:"Star Wars");
29         cart.searchByTitle(title:"Frozen");
30     }
31 }

```

- Store Class

```

AIMSProject > src > hust > soict > dsai > aims > store > J Store.java > {} hust.soict.dsai.aims.store
1  package hust.soict.dsai.aims.store;
2  import hust.soict.dsai.aims.disc.DigitalVideoDisc;
3
4  public class Store {
5      private DigitalVideoDisc[] itemsInStore = new DigitalVideoDisc[50];
6      private int numberOfItems = 0;
7
8      public void addDVD(DigitalVideoDisc dvd) {
9          if (numberOfItems < itemsInStore.length) {
10             itemsInStore[numberOfItems] = dvd;
11             numberOfItems++;
12             System.out.println("DVD " + dvd.getTitle() + " has been added to the store.");
13         } else {
14             System.out.println(x:"The store is full, cannot add more DVDs.");
15         }
16     }
17
18     public void removeDVD(DigitalVideoDisc dvd) {
19         boolean found = false;
20         for (int i = 0; i < numberOfItems; i++) {
21             if (itemsInStore[i].equals(dvd)) {
22                 for (int j = i; j < numberOfItems - 1; j++) {
23                     itemsInStore[j] = itemsInStore[j + 1];
24                 }
25                 itemsInStore[numberOfItems - 1] = null;
26                 numberOfItems--;
27                 System.out.println("DVD " + dvd.getTitle() + " has been removed from the store.");
28                 found = true;
29                 break;
30             }
31         }
32         if (!found) {
33             System.out.println("DVD " + dvd.getTitle() + " not found in the store.");
34         }
35     }
36
37     public void printStoreDetails() {
38         System.out.println(x:"*****STORE*****");
39         if (numberOfItems == 0) {
40             System.out.println(x:"No DVDs available in the store.");
41         } else {
42             for (int i = 0; i < numberOfItems; i++) {
43                 DigitalVideoDisc dvd = itemsInStore[i];

```

- StoreTest

```

1 package hust.soict.dsai.test.store;
2
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4 import hust.soict.dsai.aims.store.Store;
5 public class StoreTest {
6     public static void main(String[] args) {
7         Store store = new Store();
8
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:87, cost:19.95f);
10        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
11        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladdin", category:"Animation", cost:18.99f);
12
13        store.addDVD(dvd1);
14        store.addDVD(dvd2);
15        store.addDVD(dvd3);
16
17        store.printStoreDetails();
18
19        store.removeDVD(dvd2);
20        store.removeDVD(dvd3);
21
22        store.printStoreDetails();
23    }
24 }
25

```

- Concatenation

```

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

```

- Garbage Creator

```

package hust.soict.dsai.garbage;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

public class GarbageCreator {
    Run | Debug
    public static void main(String[] args) {
        String filename = "D:/IT3103/Lab03/OtherProjects/hust/soict/dsai/garbage/text.txt";
        byte[] inputBytes = new byte[0];
        long startTime, endTime;

        try {
            // Read all bytes from the file
            inputBytes = Files.readAllBytes(Paths.get(filename));
            startTime = System.currentTimeMillis();

            // Concatenate strings using the + operator
            String outputString = "";
            for (byte b : inputBytes) {
                outputString += (char) b;
            }

            endTime = System.currentTimeMillis();
            System.out.println("Processing time with String concatenation (using +): " + (endTime - startTime) + " ms");

            startTime = System.currentTimeMillis();

            // Use StringBuffer to build the string
            StringBuffer sb = new StringBuffer();
            for (byte b : inputBytes) {
                sb.append((char) b);
            }
            outputString = sb.toString();

            endTime = System.currentTimeMillis();
            System.out.println("Processing time with StringBuffer: " + (endTime - startTime) + " ms");

        } catch (IOException e) {
            System.out.println("Error reading the file: " + e.getMessage());
        }
    }
}

```

- NoGarbage

```

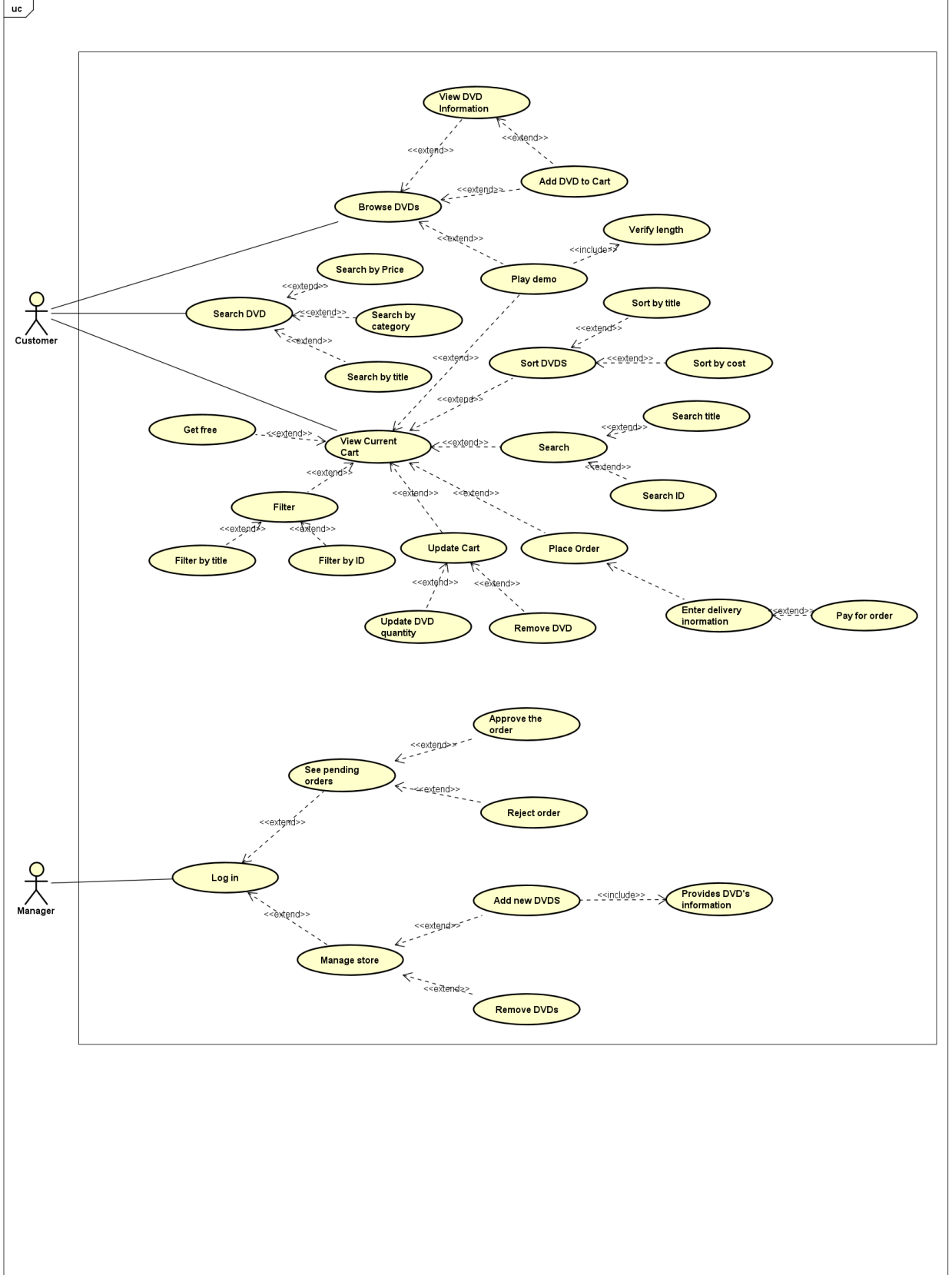
package hust.soict.dsai.garbage;

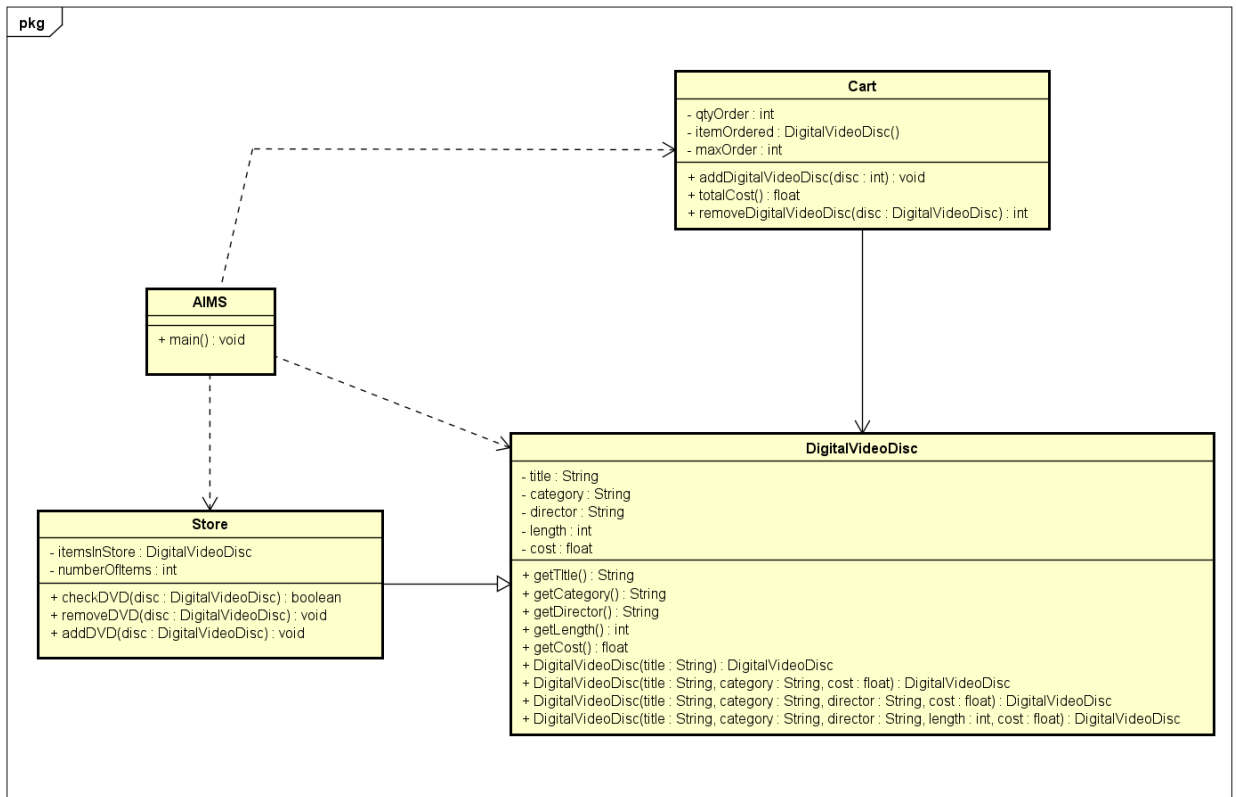
import java.io.*;

public class NoGarbage {
    Run | Debug
    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer();
        try {
            FileInputStream fileInputStream = new FileInputStream(name:"D:/IT3103/Lab03/OtherProjects/hust/soict/dsai/garbage/text.txt");
            int content;
            while ((content = fileInputStream.read()) != -1) {
                sb.append((char) content);
            }
            fileInputStream.close();
            String result = sb.toString();
        } catch (IOException e) {
            System.out.println("Error while reading the file: " + e.getMessage());
        }
    }
}

```

- Update use case and class diagrams





Answer question

- Page 6 : JAVA is a Pass by Value programming language
- Page 8 :

Titles unchanged after swap: Java passes references by value, so the method swaps only local copies, leaving the original references outside unchanged.

Title changed after changeTitle: The method modifies the object's state directly via setTitle, affecting the actual object referenced by jungleDVD.