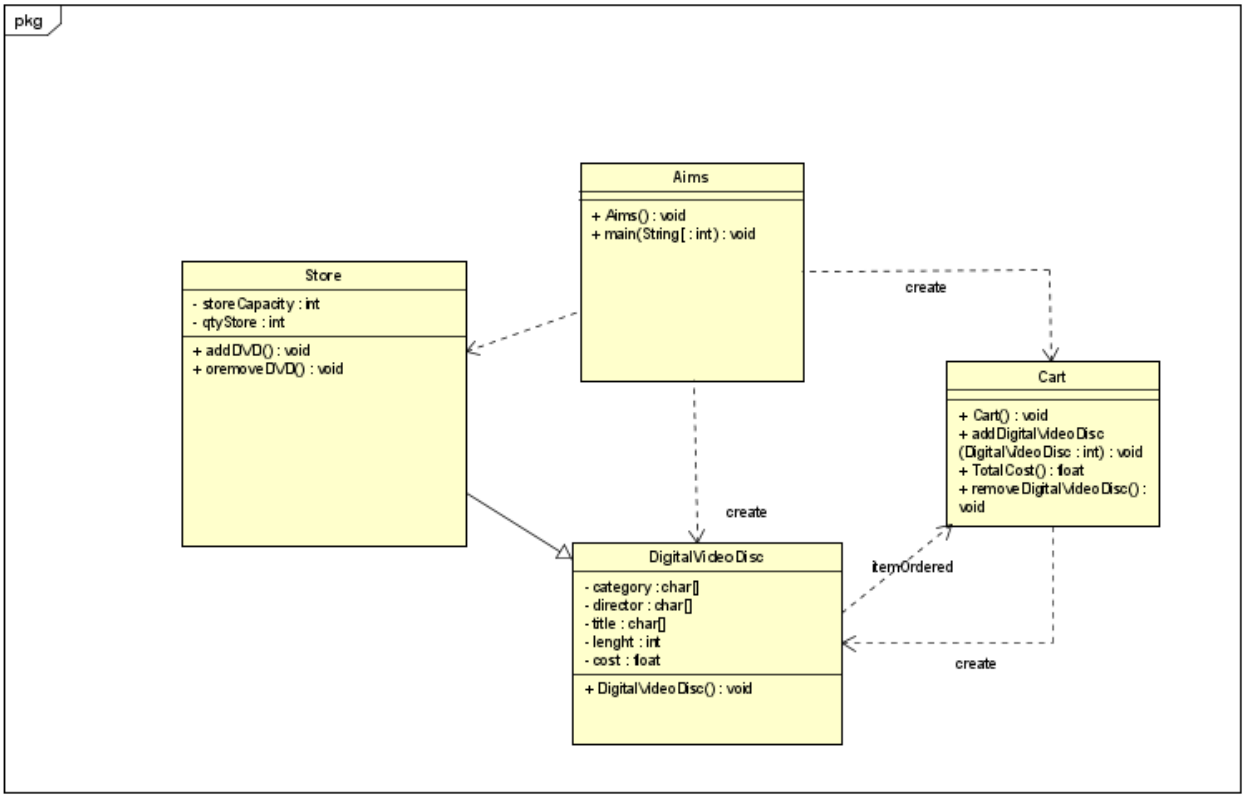
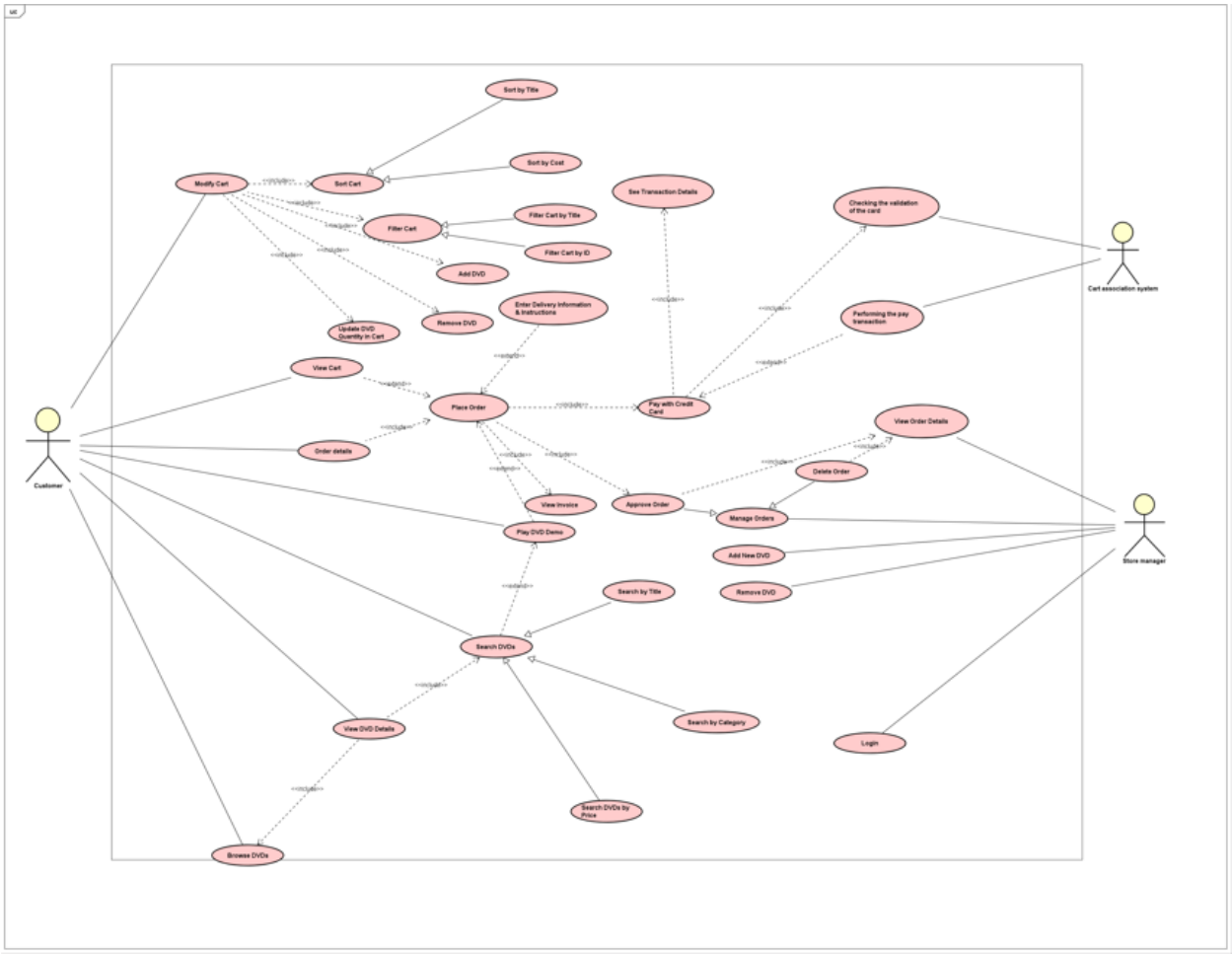
Name: Nguyen Duc Hoang

ID: 20235938

1. Use-case diagram & class diagram:





2. Working with method overloading:

```java
public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList) {  no usages
    for (DigitalVideoDisc disc : dvdList) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered += 1;
            System.out.println("DVD list added");
        } else {
            System.out.println("Maximum capacity reached");
        }
    }
}

public void addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2) {  no usages
    if (qtyOrdered + 1 < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = dvd1;
        itemsOrdered[qtyOrdered + 1] = dvd2;
        qtyOrdered += 2;
        System.out.println("2 DVD added");
    } else {
        System.out.println("Maximum capacity reached");
    }
}
```

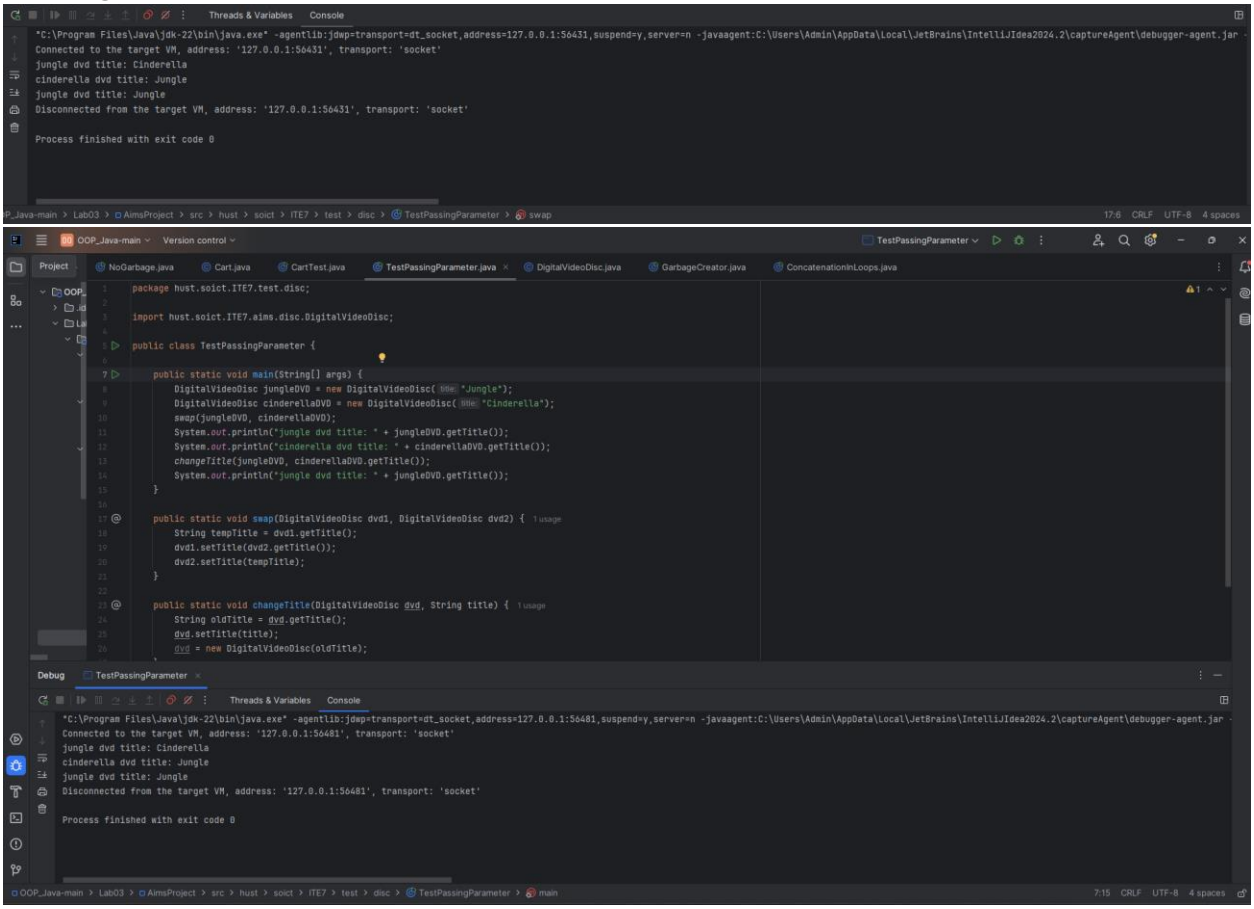
I choose array parameter because it is easy to add DVDs.

3. Passing parameter:

```java
package hust.soict.ITE7.test.disc;

import hust.soict.ITE7.aims.disc.DigitalVideoDisc;

public class TestPassingParameter {

    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(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {  1 usage
        String tempTitle = dvd1.getTitle();
        dvd1.setTitle(dvd2.getTitle());
        dvd2.setTitle(tempTitle);
    }

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

- Java is a **pass-by-value** language, meaning that when objects are passed to methods, the **value** of the reference (i.e., the memory address) is passed, not the actual object.
- Because it's just a method which is a way to copy, so the original object is not changed.
- The func changeTitle take all address and modify at that, so it will change the original object.

4. Debug:



5. Classifier and Instance:

```java
package hust.soict.ITE7.aims.disc;
public class DigitalVideoDisc {  44 usages
    private String title;  7 usages
    private String category;  4 usages
    private String director;  3 usages
    private int length;  2 usages
    private float cost;  4 usages
    private int id;  6 usages
    private static int nbDigitalVideoDiscs = 0;  8 usages
    public String getTitle() { return title; }
    public String getCategory() { return category; }
    public String getDirector() { return director; }
    public int getLength() { return length; }
    public float getCost() { return cost; }
    public int getId() { return id; }
    public DigitalVideoDisc(String title) {  3 usages
        super();
        this.title = title;
        this.id = nbDigitalVideoDiscs;
        nbDigitalVideoDiscs += 1;
    }
    public DigitalVideoDisc(String title, String category, float cost) {  3 usages
        super();
        this.title = title;
        this.category = category;
        this.cost = cost;
        this.id = nbDigitalVideoDiscs;
        nbDigitalVideoDiscs += 1;
    }
    public DigitalVideoDisc(String title, String category, String director, float cost) {  no usages
        super();
        this.title = title;
        this.category = category;
        this.director = director;
        this.cost = cost;
        this.id = nbDigitalVideoDiscs;
        nbDigitalVideoDiscs += 1;
    }
    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {  6 usages
        super();
```

6. Open the Cart class:
   - String.

```java
    public void print() {  1 usage
        System.out.println("**********CART**********");
        System.out.println("Ordered Items:");
        for (int i = 0; i < qtyOrdered; i++) {
            System.out.print((i + 1) + ". DVD - ");
            if (itemsOrdered[i].getTitle() != null) {
                System.out.print(itemsOrdered[i].getTitle() + " - ");
            }
            if (itemsOrdered[i].getCategory() != null) {
                System.out.print(itemsOrdered[i].getCategory() + " - ");
            }
            if (itemsOrdered[i].getDirector() != null) {
                System.out.print(itemsOrdered[i].getDirector() + " - ");
            }
            if (itemsOrdered[i].getLength() != 0) {
                System.out.print(itemsOrdered[i].getLength() + ": ");
            }
            if (itemsOrdered[i].getCost() != 0) {
                System.out.println(itemsOrdered[i].getCost() + " $");
            }
        }
        System.out.println("Total cost: " + this.totalCost() + " $");
        System.out.println("**************************************************");
    }
```

   -

-

```java
public void search(int id) {  1 usage
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].isMatch(id)) {
            System.out.print("Found: " + (i + 1) + ". DVD - ");
            if (itemsOrdered[i].getTitle() != null) {
                System.out.print(itemsOrdered[i].getTitle() + " - ");
            }
            if (itemsOrdered[i].getCategory() != null) {
                System.out.print(itemsOrdered[i].getCategory() + " - ");
            }
            if (itemsOrdered[i].getDirector() != null) {
                System.out.print(itemsOrdered[i].getDirector() + " - ");
            }
            if (itemsOrdered[i].getLength() != 0) {
                System.out.print(itemsOrdered[i].getLength() + ": ");
            }
            if (itemsOrdered[i].getCost() != 0) {
                System.out.println(itemsOrdered[i].getCost() + " $");
            }
            found = true;
        }
    }
    if (found == false) {
        System.out.println("No match is found");
    }
}
```

-

```java
public void search(String title) {  1 usage
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {

        if (itemsOrdered[i].isMatch(title)) {
            System.out.print("Found: " + (i + 1) + ". DVD - ");
            if (itemsOrdered[i].getTitle() != null) {
                System.out.print(itemsOrdered[i].getTitle() + " - ");
            }
            if (itemsOrdered[i].getCategory() != null) {
                System.out.print(itemsOrdered[i].getCategory() + " - ");
            }
            if (itemsOrdered[i].getDirector() != null) {
                System.out.print(itemsOrdered[i].getDirector() + " - ");
            }
            if (itemsOrdered[i].getLength() != 0) {
                System.out.print(itemsOrdered[i].getLength() + ": ");
            }
            if (itemsOrdered[i].getCost() != 0) {
                System.out.println(itemsOrdered[i].getCost() + " $");
            }
            found = true;
        }
    }
    if (found == false) {
        System.out.println("No match is found");
    }
}
}
```

-

```
DVD added
DVD added
DVD added
**********CART**********
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95 $
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95 $
3. DVD - Aladdin - Animation - 18.99 $
Total cost: 63.89 $
**************************************************
Found: 3. DVD - Aladdin - Animation - 18.99 $
Found: 2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95 $

Process finished with exit code 0
```

P_Java-main › Lab03 › AimsProject › src › hust › soict › ITE7 › test › cart › CartTest 22:2 CRLF UTF-8 4 spaces

7. Impliment Store class:

```java
package hust.soict.ITE7.test.store;
import hust.soict.ITE7.aims.disc.DigitalVideoDisc;
import hust.soict.ITE7.aims.store.Store;

public class StoreTest {

    public static void main(String[] args) {
        Store store = new Store();
        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: "Aladdin",
                category: "Animation", cost: 18.99f);
        store.addDVD(dvd3);
        store.removeDVD(dvd2);
    }
}
```

8. String, Strting Builder & StringBuffer:

```java
package hust.soict.ITE7.garbage;

import java.util.Random;

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);
        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);
    }
}
```

```java
package hust.soict.ITE7.garbage;

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

public class GarbageCreator {
    public static void main(String[] args) throws IOException {
        String filepath = "C:/Users/Vo/Desktop/large_input.txt";
        byte[] inputBytes = {0};
        long startTime, endTime;
        inputBytes = Files.readAllBytes(Paths.get(filepath));
        startTime = System.currentTimeMillis();
        String outputString = "";
        for (byte b : inputBytes) {
            outputString += (char)b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

```java
package hust.soict.ITE7.garbage;

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

public class NoGarbage {
    public static void main(String[] args) throws IOException {
        String filepath = "C:/Users/Vo/Desktop/large_input.txt";
        byte[] inputBytes = {0};
        long startTime, endTime;
        inputBytes = Files.readAllBytes(Paths.get(filepath));
        startTime = System.currentTimeMillis();
        StringBuilder outputString = new StringBuilder();
        for (byte b : inputBytes) {
            outputString.append(b);
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```