Lab03 Report

Name: Lương Minh Hiếu   ID: 20230083
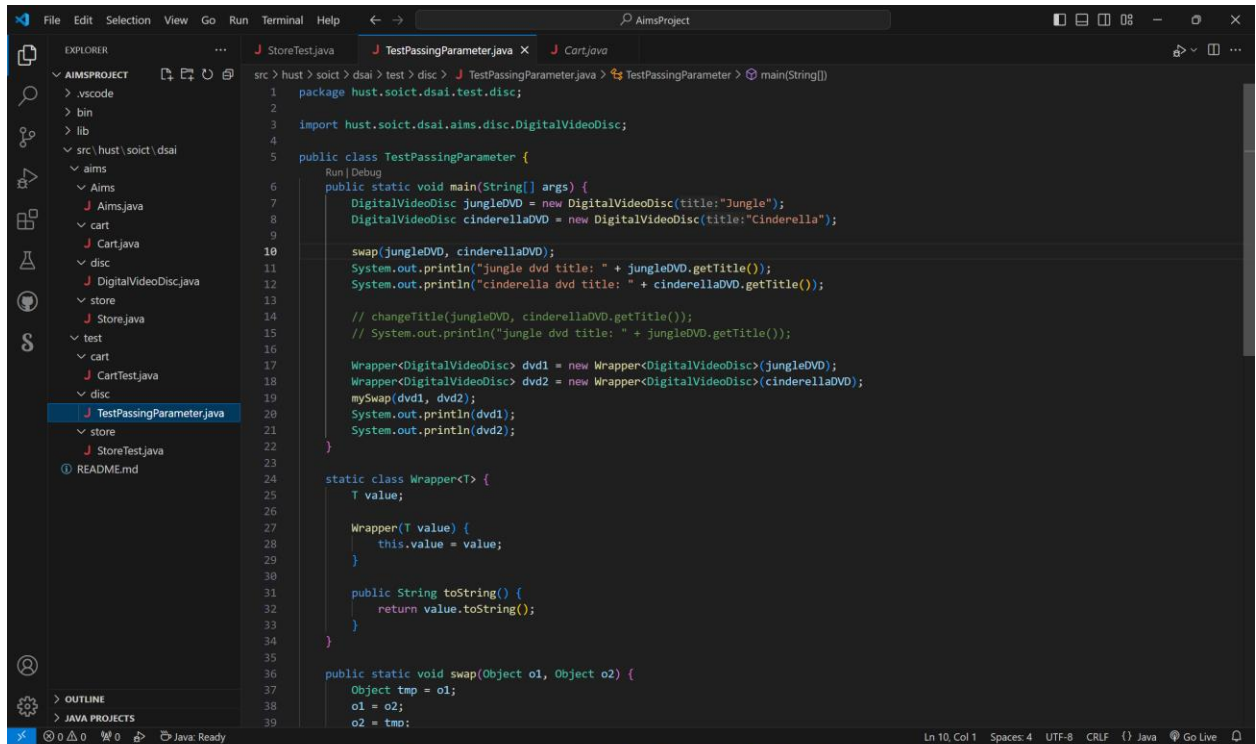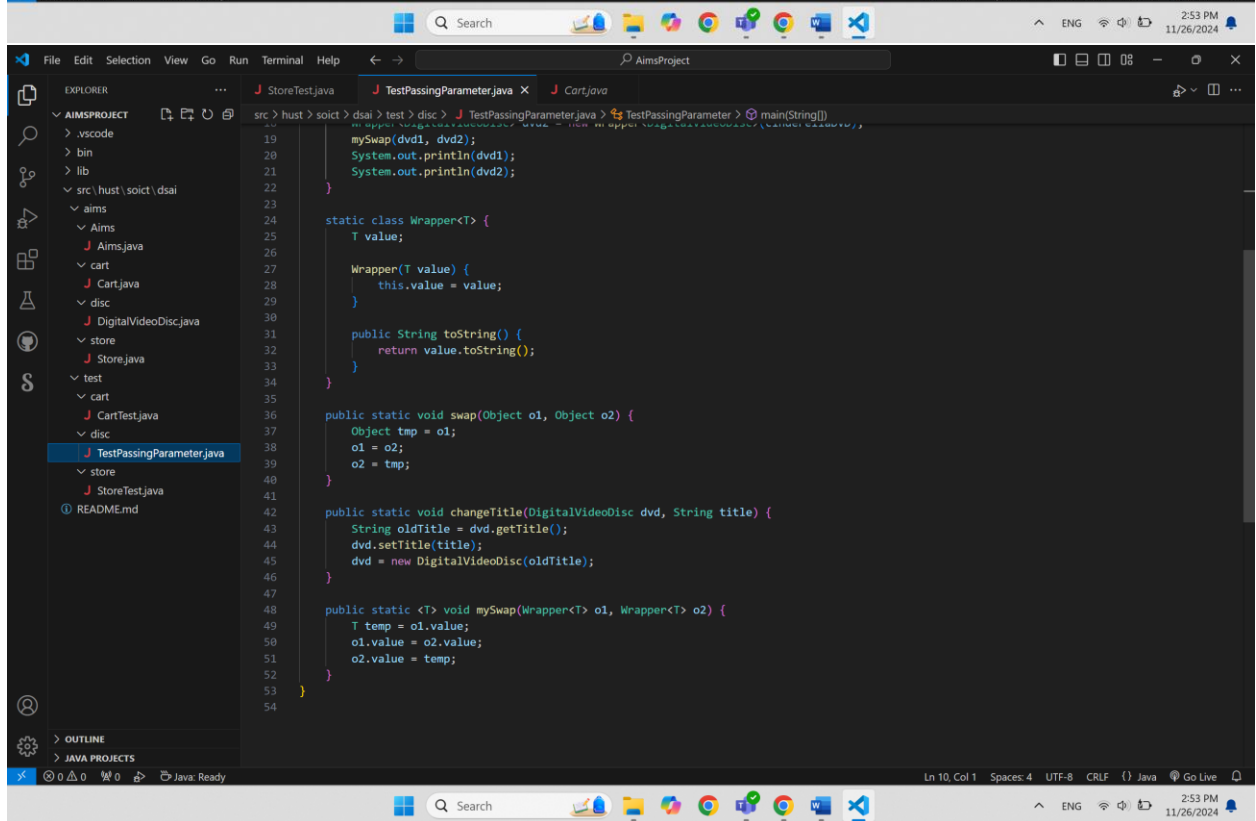
I.   Source code
1) Method overloading

2) Passing parameter


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

import hust.soict.dsai.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());

        Wrapper<DigitalVideoDisc> dvd1 = new Wrapper<DigitalVideoDisc>(jungleDVD);
        Wrapper<DigitalVideoDisc> dvd2 = new Wrapper<DigitalVideoDisc>(cinderellaDVD);
        mySwap(dvd1, dvd2);
        System.out.println(dvd1);
        System.out.println(dvd2);
    }

    static class Wrapper<T> {
        T value;

        Wrapper(T value) {
            this.value = value;
        }

        public String toString() {
            return value.toString();
        }
    }

    public static void swap(Object o1, Object o2) {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
```



```java
        mySwap(dvd1, dvd2);
        System.out.println(dvd1);
        System.out.println(dvd2);
    }

    static class Wrapper<T> {
        T value;

        Wrapper(T value) {
            this.value = value;
        }

        public String toString() {
            return value.toString();
        }
    }

    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);
    }

    public static <T> void mySwap(Wrapper<T> o1, Wrapper<T> o2) {
        T temp = o1.value;
        o1.value = o2.value;
        o2.value = temp;
    }
}
```

3) Use Debug Run

Screenshot 1 (top)

File Edit Selection View Go Run Terminal Help

StoreTest.java | TestPassingParameter.java

src > hust > soict > dsai > test > disc > TestPassingParameter.java > TestPassingParameter > swap(Object, Object)

```java
28              this.value = value;
29          }
30
31          public String toString() {
32              return value.toString();
33          }
34      }
35
36      public static void swap(Object o1, Object o2) { o1 = DigitalVideoDisc@10 "DVD - Jungle - null - null - 0: 0.0", o2 = DigitalVideoDisc@11 "DVD
37          Object tmp = o1; tmp = DigitalVideoDisc@10 "DVD - Jungle - null - null - 0: 0.0", o1 = DigitalVideoDisc@10 "DVD
38          o1 = o2; o1 = DigitalVideoDisc@10 "DVD - Jungle - null - null - 0: 0.0", o2 = DigitalVideoDisc@11 "DVD - Cinderella - null - null - 0: 0.
39          o2 = tmp;
40      }
41
42      public static void changeTitle(DigitalVideoDisc dvd, String title) {
43          String oldTitle = dvd.getTitle();
44          dvd.setTitle(title);
45          dvd = new DigitalVideoDisc(oldTitle);
46      }
```
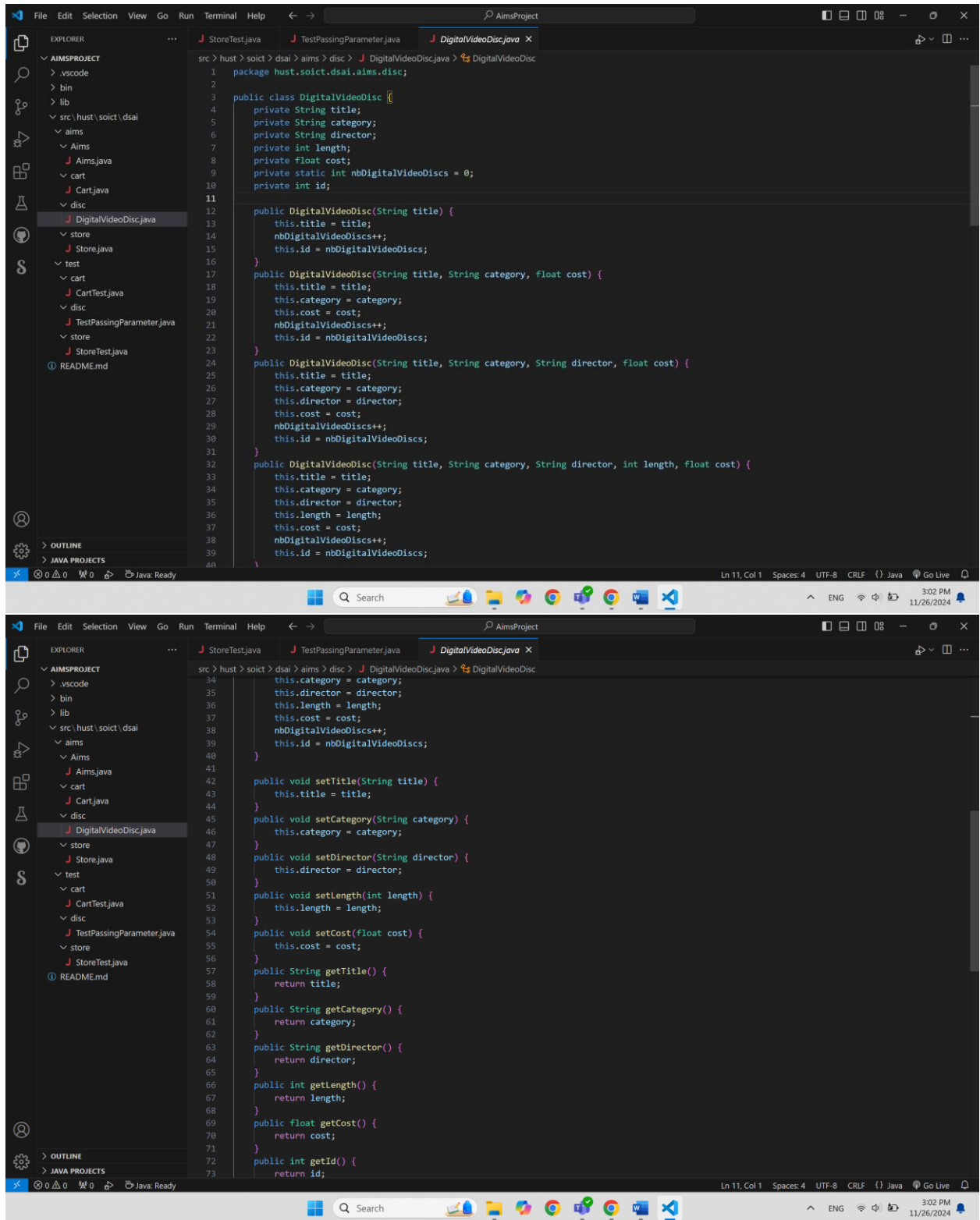
VARIABLES
- Local
  - o1 = DigitalVideoDisc@10 ...
    - category = null
    - cost = 0.000000
    - director = null
    - id = 1
    - length = 0
    - title = "Jungle"
  - o2 = DigitalVideoDisc@11 ...
    - category = null
    - cost = 0.000000
    - director = null
    - id = 2
    - length = 0
    - title = "Cinderella"
  - tmp = DigitalVideoDisc@1...
    - category = null
    - cost = 0.000000
    - director = null
    - id = 1
    - length = 0
    - title = "Jungle"

WATCH
CALL STACK
BREAKPOINTS
- Uncaught Exceptions
- Caught Exceptions
- TestPassingParameter.j... 10
- TestPassingParameter.j... 11
- TestPassingParameter.j... 12
- TestPassingParameter.j... 37
- TestPassingParameter.j... 38
- TestPassingParameter.j... 39

Ln 38, Col 1   Spaces: 4   UTF-8   CRLF   {} Java

Screenshot 2 (bottom)

File Edit Selection View Go Run Terminal Help

StoreTest.java | TestPassingParameter.java

src > hust > soict > dsai > test > disc > TestPassingParameter.java > TestPassingParameter > swap(Object, Object)

```java
28              this.value = value;
29          }
30
31          public String toString() {
32              return value.toString();
33          }
34      }
35
36      public static void swap(Object o1, Object o2) { o1 = DigitalVideoDisc@11 "DVD - Cinderella - null - null - 0: 0.0", o2 = DigitalVideoDisc@11
37          Object tmp = o1; tmp = DigitalVideoDisc@10 "DVD - Jungle - null - null - 0: 0.0", o1 = DigitalVideoDisc@11 "DVD - Cinderella - null - nul
38          o1 = o2; o1 = DigitalVideoDisc@11 "DVD - Cinderella - null - null - 0: 0.0", o2 = DigitalVideoDisc@11 "DVD - Cinderella - null - null - 0
39          o2 = tmp; o2 = DigitalVideoDisc@11 "DVD - Cinderella - null - null - 0: 0.0", tmp = DigitalVideoDisc@10 "DVD - Jungle - null - null - 0:
40      }
41
42      public static void changeTitle(DigitalVideoDisc dvd, String title) {
43          String oldTitle = dvd.getTitle();
44          dvd.setTitle(title);
45          dvd = new DigitalVideoDisc(oldTitle);
46      }
```

VARIABLES
- Local
  - o1 = DigitalVideoDisc@11 ...
    - category = null
    - cost = 0.000000
    - director = null
    - id = 2
    - length = 0
    - title = "Cinderella"
  - o2 = DigitalVideoDisc@11 ...
    - category = null
    - cost = 0.000000
    - director = null
    - id = 2
    - length = 0
    - title = "Cinderella"
  - tmp = DigitalVideoDisc@1...
    - category = null
    - cost = 0.000000
    - director = null
    - id = 1
    - length = 0
    - title = "Jungle"

WATCH
CALL STACK
BREAKPOINTS
- Uncaught Exceptions
- Caught Exceptions
- TestPassingParameter.j... 10
- TestPassingParameter.j... 11
- TestPassingParameter.j... 12
- TestPassingParameter.j... 37
- TestPassingParameter.j... 38
- TestPassingParameter.j... 39

Ln 39, Col 1   Spaces: 4   UTF-8   CRLF   {} Java
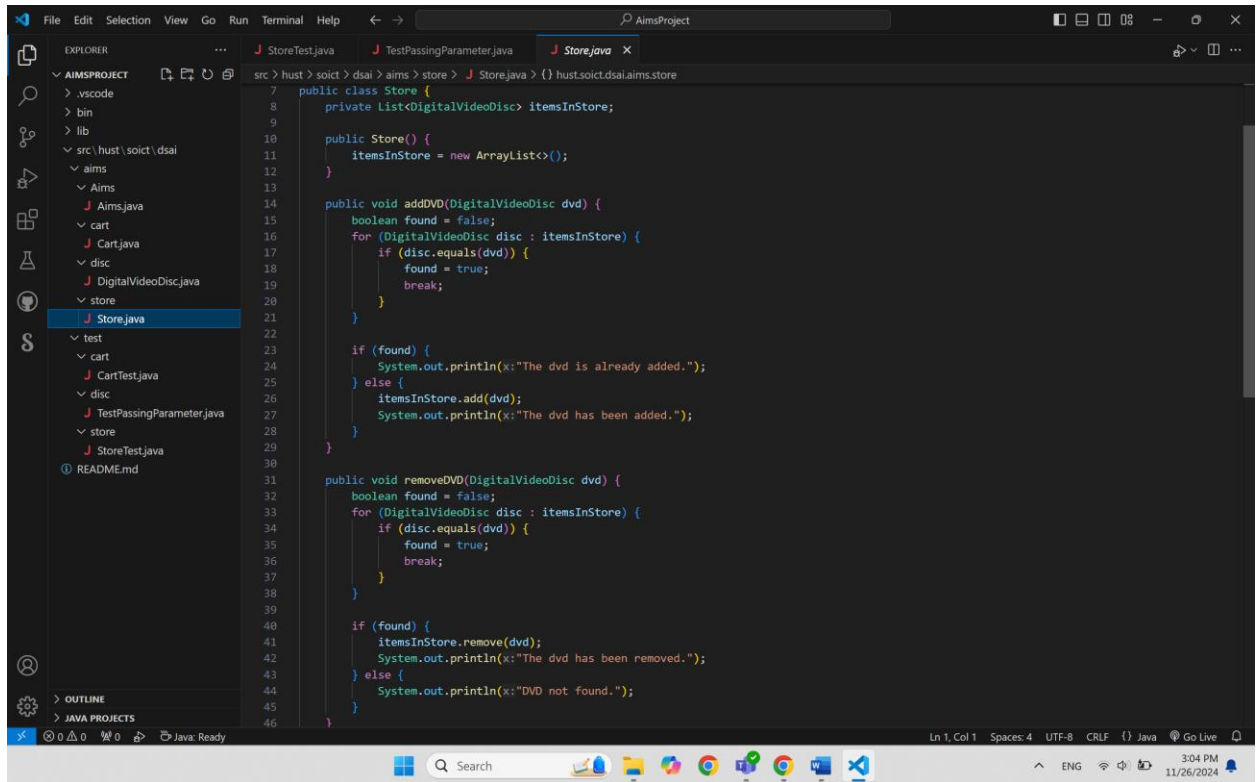
4) Classifier Member and Instance Member


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



```java
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
        nbDigitalVideoDiscs++;
        this.id = nbDigitalVideoDiscs;
    }

    public void setTitle(String title) {
        this.title = title;
    }
    public void setCategory(String category) {
        this.category = category;
    }
    public void setDirector(String director) {
        this.director = director;
    }
    public void setLength(int length) {
        this.length = length;
    }
    public void setCost(float cost) {
        this.cost = cost;
    }
    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;
```

5) Cart Class
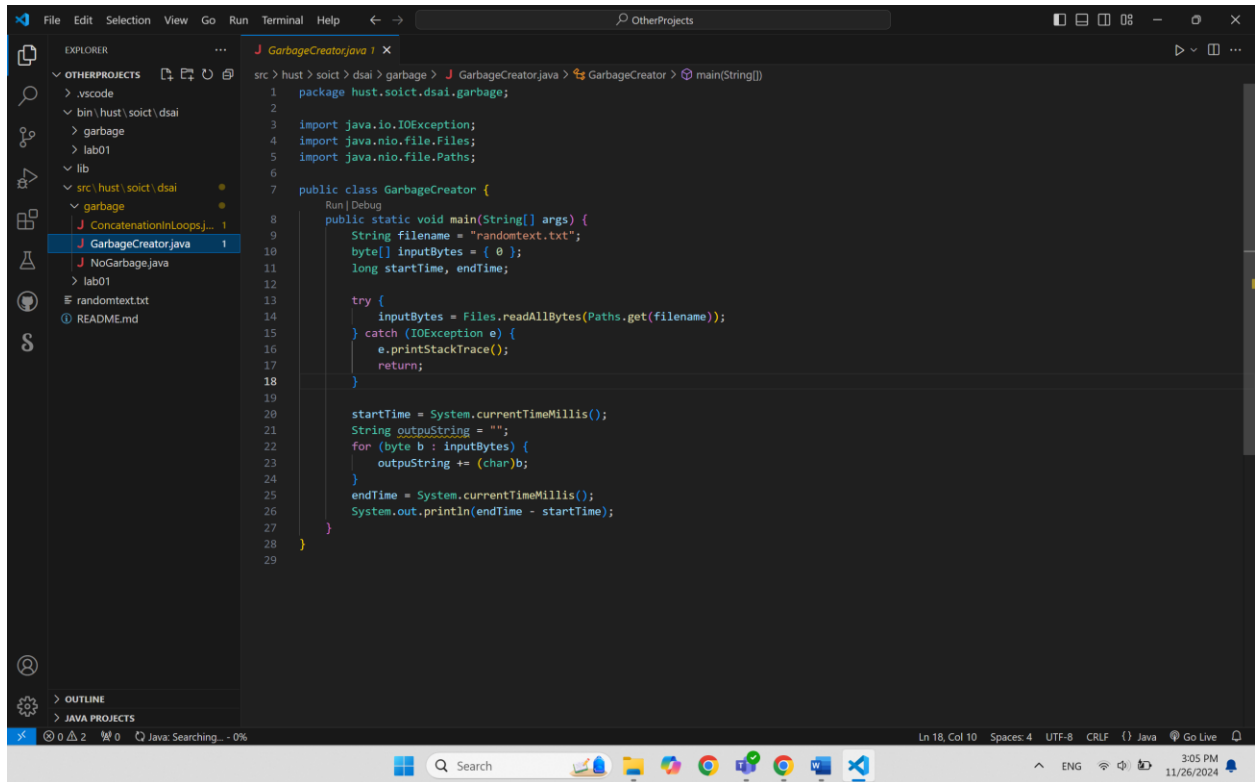

```java
public void displayCart() {
    System.out.println(x:"**********************CART*********************");
    System.out.println(x:"Ordered Items:");
    for (int i = 0; i < qtyOrdered; i++) {
        System.out.println(String.valueOf(i+1) + ". DVD - " + itemsOrdered[i].toString());
    }
    System.out.println("Total cost: " + totalCost());
    System.out.println(x:"*********************************************");
}

public void searchDVD(int id) {
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].getId() == id) {
            found = true;
            System.out.println(x:"Disc is found. Displaying information:");
            System.out.println(itemsOrdered[i].toString());
            break;
        }
    }

    if (!found) {
        System.out.println(x:"Disc not found.");
    }
}

public void searchDVD(String title) {
    boolean found = false;
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].isMatch(title)) {
            found = true;
            System.out.println(x:"Disc is found. Displaying information:");
            System.out.println(itemsOrdered[i].toString());
            break;
        }
    }

    if (!found) {
        System.out.println(x:"Disc not found.");
    }
```

6) Store Class

```java
public class Store {
    private List<DigitalVideoDisc> itemsInStore;

    public Store() {
        itemsInStore = new ArrayList<>();
    }

    public void addDVD(DigitalVideoDisc dvd) {
        boolean found = false;
        for (DigitalVideoDisc disc : itemsInStore) {
            if (disc.equals(dvd)) {
                found = true;
                break;
            }
        }

        if (found) {
            System.out.println(x:"The dvd is already added.");
        } else {
            itemsInStore.add(dvd);
            System.out.println(x:"The dvd has been added.");
        }
    }

    public void removeDVD(DigitalVideoDisc dvd) {
        boolean found = false;
        for (DigitalVideoDisc disc : itemsInStore) {
            if (disc.equals(dvd)) {
                found = true;
                break;
            }
        }

        if (found) {
            itemsInStore.remove(dvd);
            System.out.println(x:"The dvd has been removed.");
        } else {
            System.out.println(x:"DVD not found.");
        }
    }
}
```

7) String, StringBuilder and StringBuffer
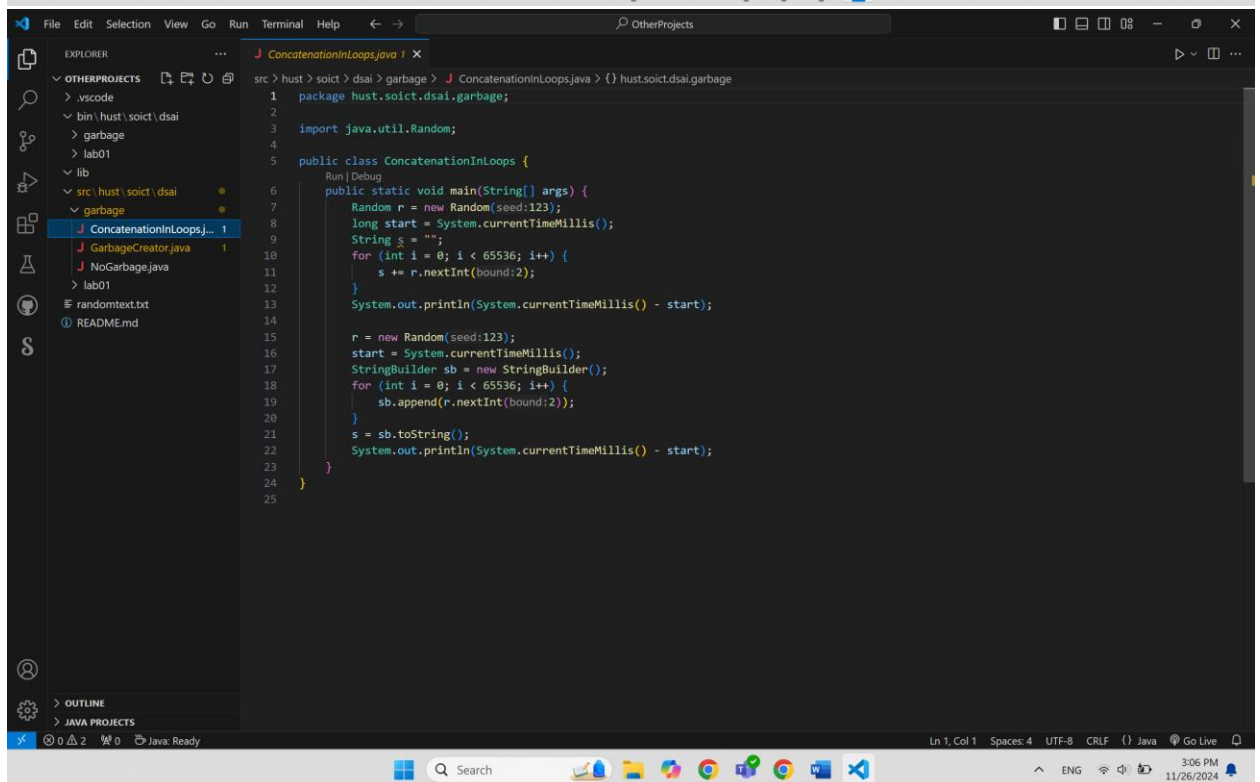
```java
package hust.soict.dsai.garbage;

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

public class GarbageCreator {
    public static void main(String[] args) {
        String filename = "randomtext.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            e.printStackTrace();
            return;
        }

        startTime = System.currentTimeMillis();
        String outpuString = "";
        for (byte b : inputBytes) {
            outpuString += (char)b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

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

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

public class NoGarbage {
    public static void main(String[] args) {
        String filename = "randomtext.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            e.printStackTrace();
            return;
        }

        startTime = System.currentTimeMillis();
        StringBuilder outpuStringbuilder = new StringBuilder();
        for (byte b : inputBytes) {
            outpuStringbuilder.append((char)b);
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

```java
package hust.soict.dsai.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);
    }
}
```
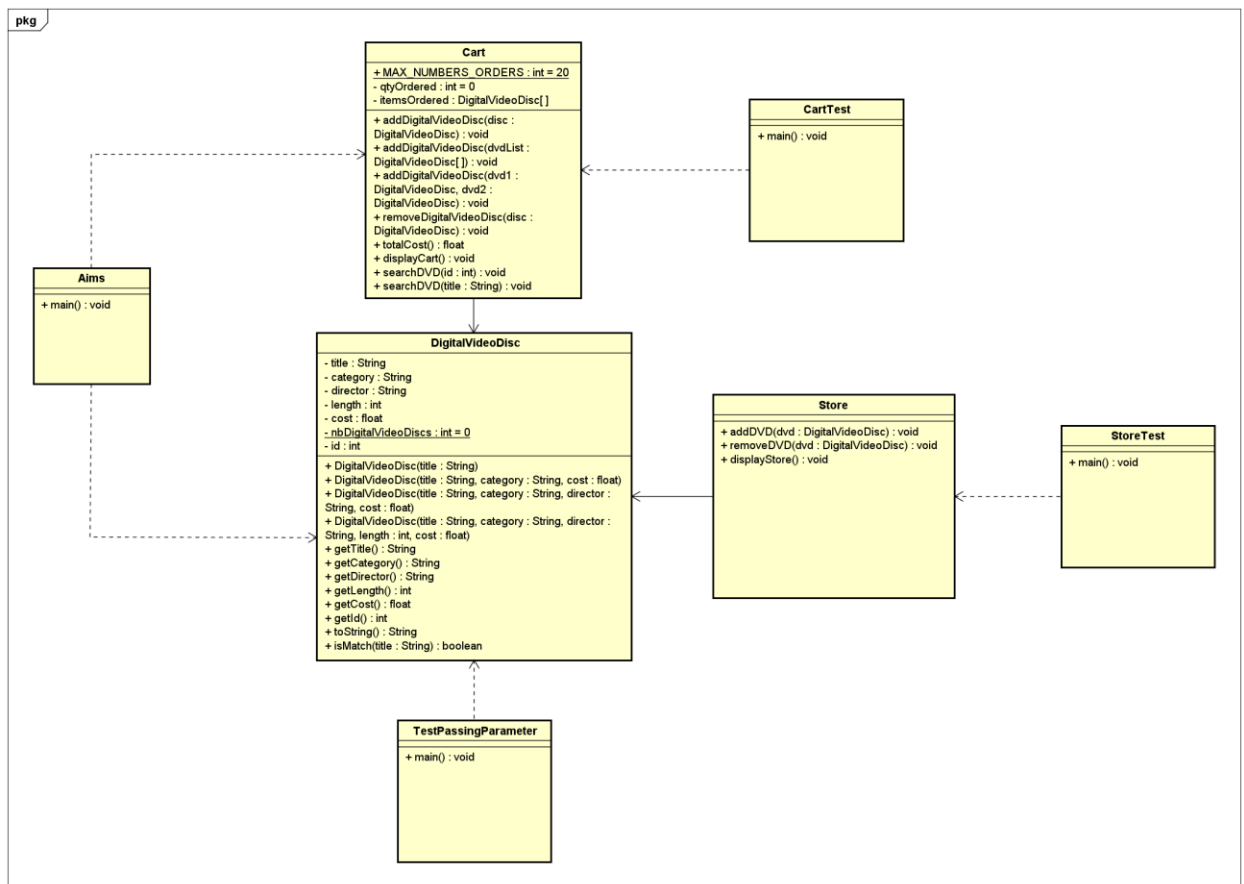
II. Updated diagram

1) Use Case
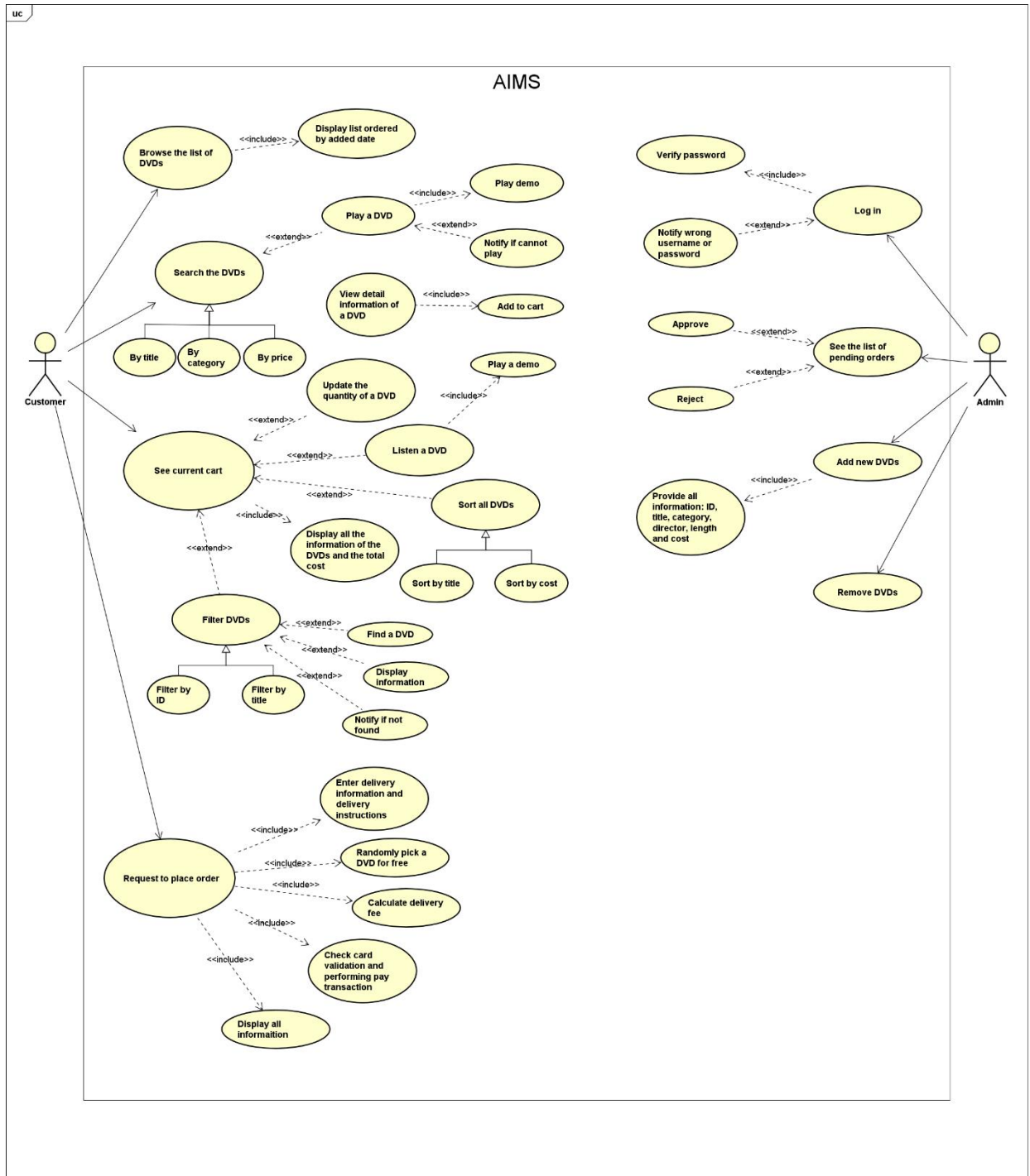
2) Class diagram

III.   Answer questions

1) After the call of **swap(jungleDVD, cinderellaDVD)** why does the title of these two objects still remain?
2) After the call of **changeTitle(jungleDVD, cinderellaDVD.getTitle())** why is the title of the JungleDVD changed?
3) ***Is JAVA a Pass by Value or a Pass by Reference programming language?***

Java is a pass by value programming language, so after the call of **swap(jungleDVD, cinderellaDVD),** this method only swaps the copies of the references, not the actual references, hence these two objects still remain. In the second method, the dvd parameter is a reference to the same DigitalVideoDisc object as jungle DVD, hence when you use method setTitle, it change the title of that actual object.

Conclusion, Java is a pass by value programming language.