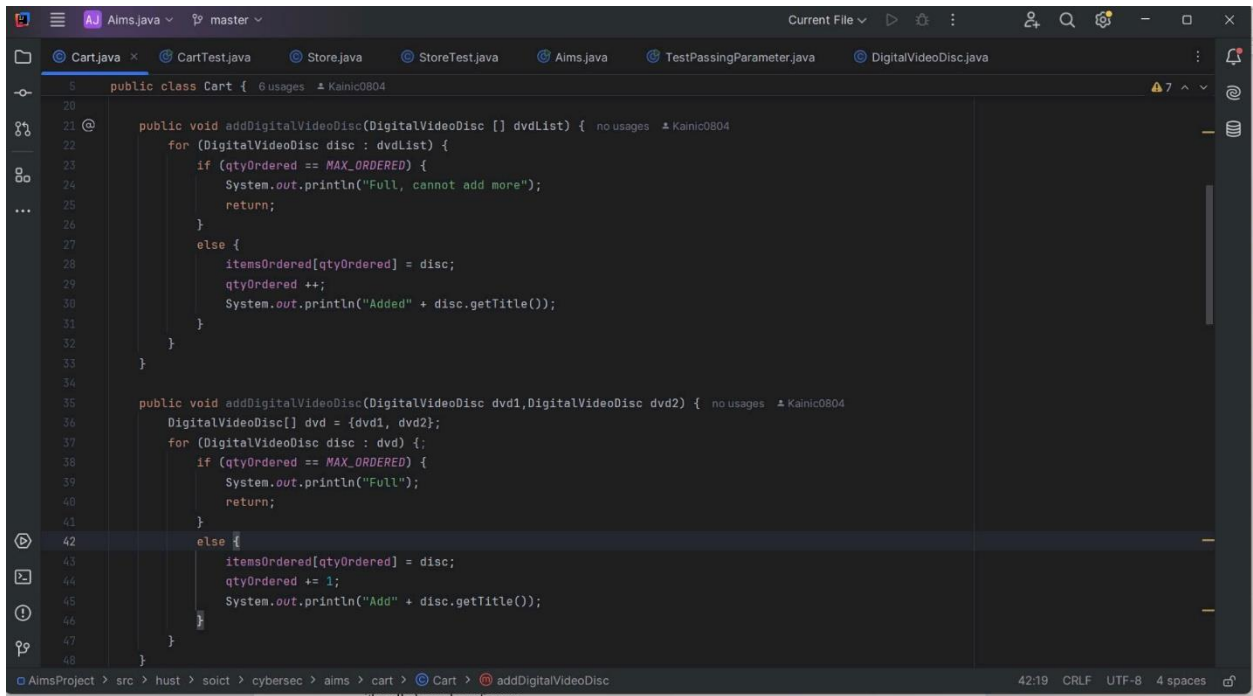


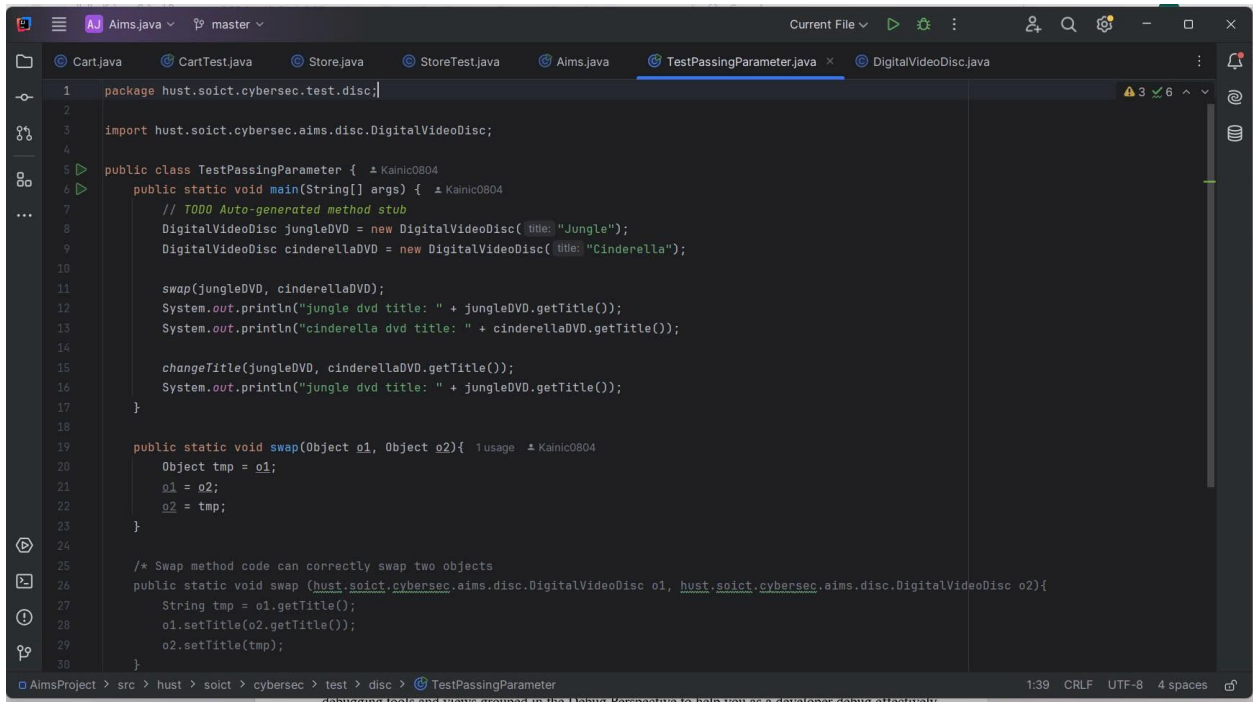
I. Screenshot of new written code

1. Method Overloading



```
20 public class Cart {
21     public void addDigitalVideoDisc(DigitalVideoDisc [] dvdList) {
22         for (DigitalVideoDisc disc : dvdList) {
23             if (qtyOrdered == MAX_ORDERED) {
24                 System.out.println("Full, cannot add more");
25                 return;
26             }
27             else {
28                 itemsOrdered[qtyOrdered] = disc;
29                 qtyOrdered++;
30                 System.out.println("Added" + disc.getTitle());
31             }
32         }
33     }
34
35     public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
36         DigitalVideoDisc[] dvd = {dvd1, dvd2};
37         for (DigitalVideoDisc disc : dvd) {
38             if (qtyOrdered == MAX_ORDERED) {
39                 System.out.println("Full");
40                 return;
41             }
42             else {
43                 itemsOrdered[qtyOrdered] = disc;
44                 qtyOrdered++;
45                 System.out.println("Add" + disc.getTitle());
46             }
47         }
48     }
49 }
```

2. Passing parameter



```
1 package hust.soict.cybersec.test.disc;
2
3 import hust.soict.cybersec.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter {
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
9         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
10
11         swap(jungleDVD, cinderellaDVD);
12         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
13         System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
14
15         changeTitle(jungleDVD, cinderellaDVD.getTitle());
16         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
17     }
18
19     public static void swap(Object o1, Object o2) {
20         Object tmp = o1;
21         o1 = o2;
22         o2 = tmp;
23     }
24
25     /* Swap method code can correctly swap two objects
26     public static void swap(hust.soict.cybersec.aims.disc.DigitalVideoDisc o1, hust.soict.cybersec.aims.disc.DigitalVideoDisc o2) {
27         String tmp = o1.getTitle();
28         o1.setTitle(o2.getTitle());
29         o2.setTitle(tmp);
30     }
31 }
```

```
5 public class TestPassingParameter { 1 Kainic0804
6     public static void main(String[] args) { 1 Kainic0804
14
15         changeTitle(jungleDVD, cinderellaDVD.getTitle());
16         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
17     }
18
19     public static void swap(Object o1, Object o2){ 1 usage 1 Kainic0804
20         Object tmp = o1;
21         o1 = o2;
22         o2 = tmp;
23     }
24
25     /* Swap method code can correctly swap two objects
26     public static void swap (hust.soiict.cybersec.aims.disc.DigitalVideoDisc o1, hust.soiict.cybersec.aims.disc.DigitalVideoDisc o2){
27         String tmp = o1.getTitle();
28         o1.setTitle(o2.getTitle());
29         o2.setTitle(tmp);
30     }
31     */
32
33     @ public static void changeTitle(DigitalVideoDisc dvd, String title){ 1 usage 1 Kainic0804
34         String oldTitle = dvd.getTitle();
35         dvd.setTitle(title);
36         dvd = new DigitalVideoDisc(oldTitle);
37     }
38 }
39
```

3. Write a toString() method, searchByID, searchByTitle, printCart

```
public String toString() { 1 Kainic0804
    return "DVD - " + this.getTitle() + " - " + this.getCategory() + " - " + this.getDirector() + " - " + this.getLength() + ": " + this.getCost() + " $";
}
```

```
72
73     public void printCart() { 1 usage 1 Kainic0804
74         System.out.println("*****CART*****");
75         System.out.println("Ordered Item:");
76         for (int i = 0; i < qtyOrdered; i++) {
77             System.out.println((i+1) + itemsOrdered[i].toString());
78         }
79         System.out.println("Total Cost: $" + totalCost());
80         System.out.println("*****");
81     }
82 }
83
```

```

04 5 public class Cart { 6 usages 1 Kainic0804
04 83
04 84 public void searchById(int id) { 2 usages 1 Kainic0804
04 85     boolean found = false;
04 86     for (int i = 0; i < qtyOrdered; i++) {
04 87         if (itemsOrdered[i].getId() == id) {
04 88             found = true;
04 89             System.out.println("DVD found");
04 90             System.out.println(itemsOrdered[i].toString());
04 91             break;
04 92         }
04 93     }
04 94     if (found == false) {
04 95         System.out.println("DVD not found");
04 96     }
04 97 }
04 98
04 99 public void searchByTitle(String title) { 2 usages 1 Kainic0804
04 100     boolean found = false;
04 101     for (int i = 0; i < qtyOrdered; i++) {
04 102         if (itemsOrdered[i].getTitle().equals(title)) {
04 103             found = true;
04 104             System.out.println("DVD found");
04 105             System.out.println(itemsOrdered[i].toString());
04 106             break;
04 107         }
04 108     }
04 109     if (found == false) {
04 110         System.out.println("DVD not found");
04 111     }
04 112 }
04 113 }

```

4. CartTest class

```

1 package hust.soict.cybersec.test.cart;
2
3 import hust.soict.cybersec.aims.cart.Cart;
4 import hust.soict.cybersec.aims.disc.DigitalVideoDisc;
5
6 public class CartTest { 1 Kainic0804
7     public static void main(String[] args) { 1 Kainic0804
8         Cart cart = new Cart();
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King",
10             category: "Animation", director: "Roger Aller", length: 87, cost: 19.95f);
11         cart.addDigitalVideoDisc(dvd1);
12
13         DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars",
14             category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
15         cart.addDigitalVideoDisc(dvd2);
16
17         DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin",
18             category: "Animation", cost: 18.99f);
19         cart.addDigitalVideoDisc(dvd3);
20
21         cart.printCart();
22         cart.searchById(3);
23         cart.searchById(2);
24         cart.searchByTitle("Star Wars");
25         cart.searchByTitle("Blackhat");
26     }
27 }
28

```

5. Store class

```

package hust.soict.cybersec.aims.store;

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

public class Store {
    private DigitalVideoDisc[] itemInStore;
    private int itemCount;

    public Store() {
        itemInStore = new DigitalVideoDisc[1000];
        itemCount = 0;
    }

    public void addDVD(DigitalVideoDisc dvd) {
        if (itemCount < itemInStore.length) {
            itemInStore[itemCount] = dvd;
            itemCount++;
            System.out.println(dvd + " added to the store");
        } else {
            System.out.println("The storage is full");
        }
    }

    public void removeDVD(DigitalVideoDisc dvd) {
        for (int i = 0; i < itemCount; i++) {
            if (itemInStore[i] == dvd) {
                for (int j = i + 1; j < itemCount - 1; j++) {
                    itemInStore[j] = itemInStore[j + 1];
                }
                itemInStore[itemCount - 1] = null;
                itemCount--;
                System.out.println(dvd + " removed from the store");
                return;
            }
        }
        System.out.println("Cannot find the item");
    }
}

```

6. StoreTest class

```

package hust.soict.cybersec.test.store;

import hust.soict.cybersec.aims.disc.DigitalVideoDisc;
import hust.soict.cybersec.aims.store.Store;

public class StoreTest {
    public static void main(String[] args) {
        Store store = new Store();
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(
            "The Lion King",
            "category: Animation",
            "director: Roger Allen",
            "length: 87",
            "cost: 19.95f");
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(
            "Star Wars",
            "category: Science Fiction",
            "director: George Lucas",
            "length: 87",
            "cost: 24.95f");
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(
            "Aladin",
            "category: Animation",
            "cost: 18.99f");

        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);

        store.removeDVD(dvd1);
        store.removeDVD(dvd3);
    }
}

```

7. Classifier Member and Instance Member

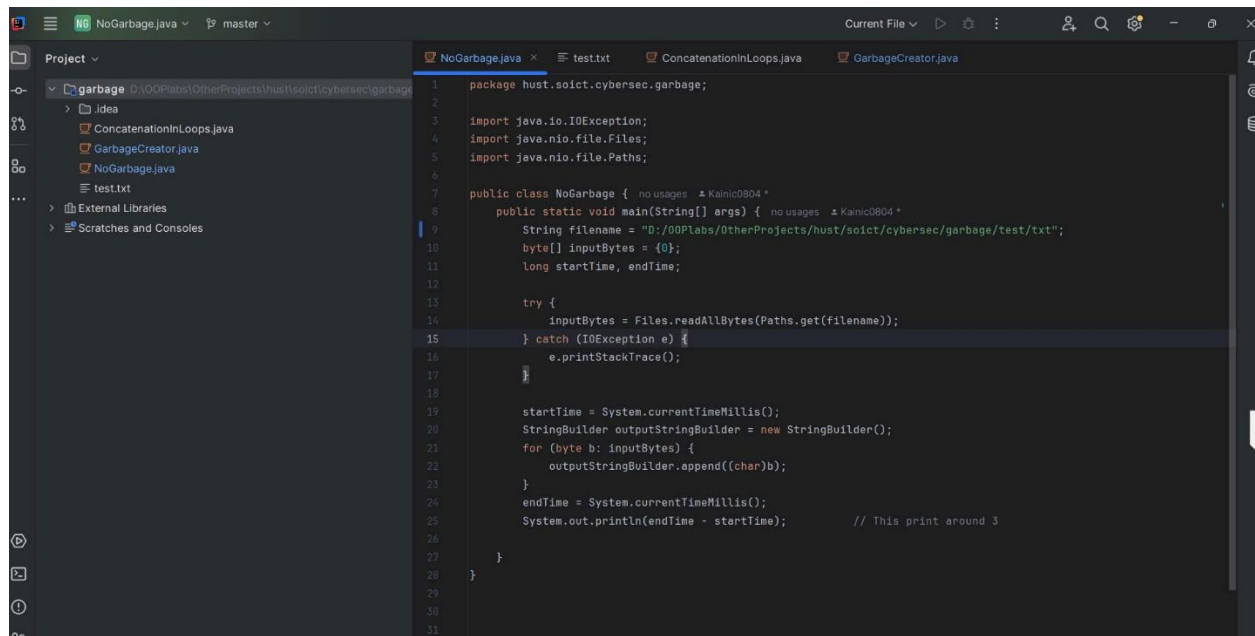
```
NoGarbage.java ConcatenationInLoops.java GarbageCreator.java DigitalVideoDisc.java x
25 public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
26     super();
27     this.title = title;
28     this.category = category;
29     this.director = director;
30     this.length = length;
31     this.cost = cost;
32     updateID();
33 }
34
35 > public String getTitle() { return title; }
38 > public String getCategory() { return category; }
41 > public String getDirector() { return director; }
44 > public int getLength() { return length; }
47 > public float getCost() { return cost; }
50 public int getId() { return id; }
51
52 public String toString() {
53     return "DVD - " + this.getTitle() + " - " + this.getCategory() + " - " + this.getDirector() + " - " + this.getLength() + ": " + this.getCost() + " $";
54 }
55
56 > public void updateID() { this.id = ++DigitalVideoDisc.nbDigitalVideoDiscs; }
59
60
61 > public void setTitle(String title) { this.title = title; }
64
65 }
66
67
68
69
70
```

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

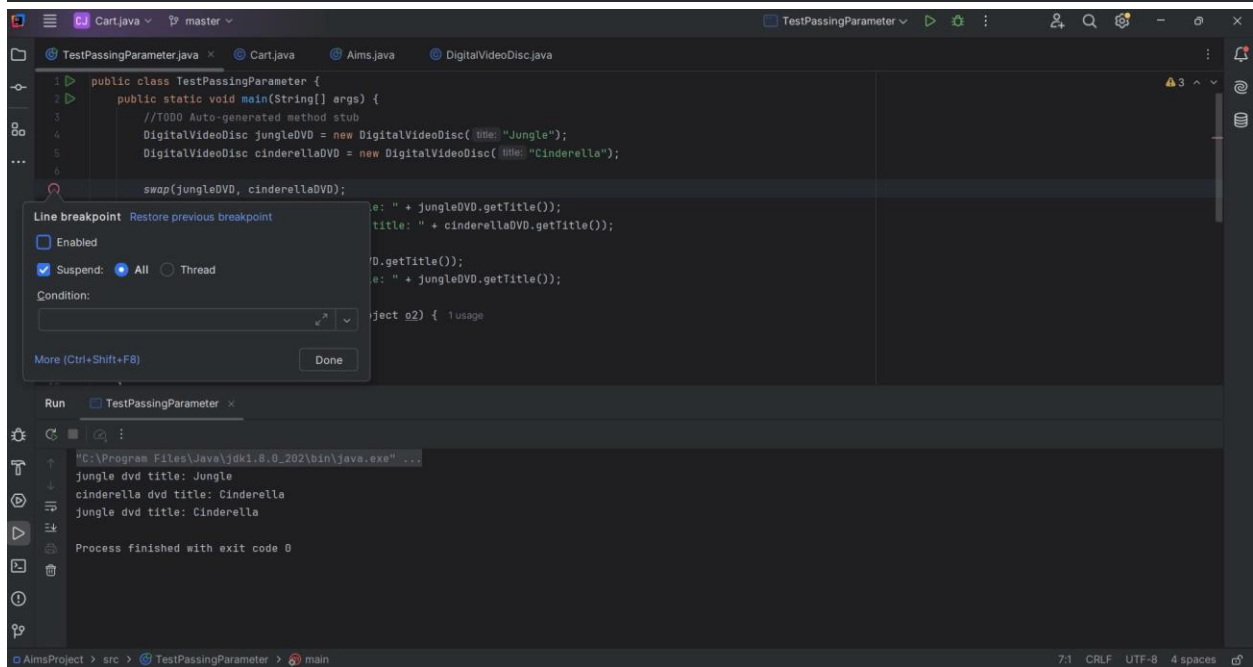
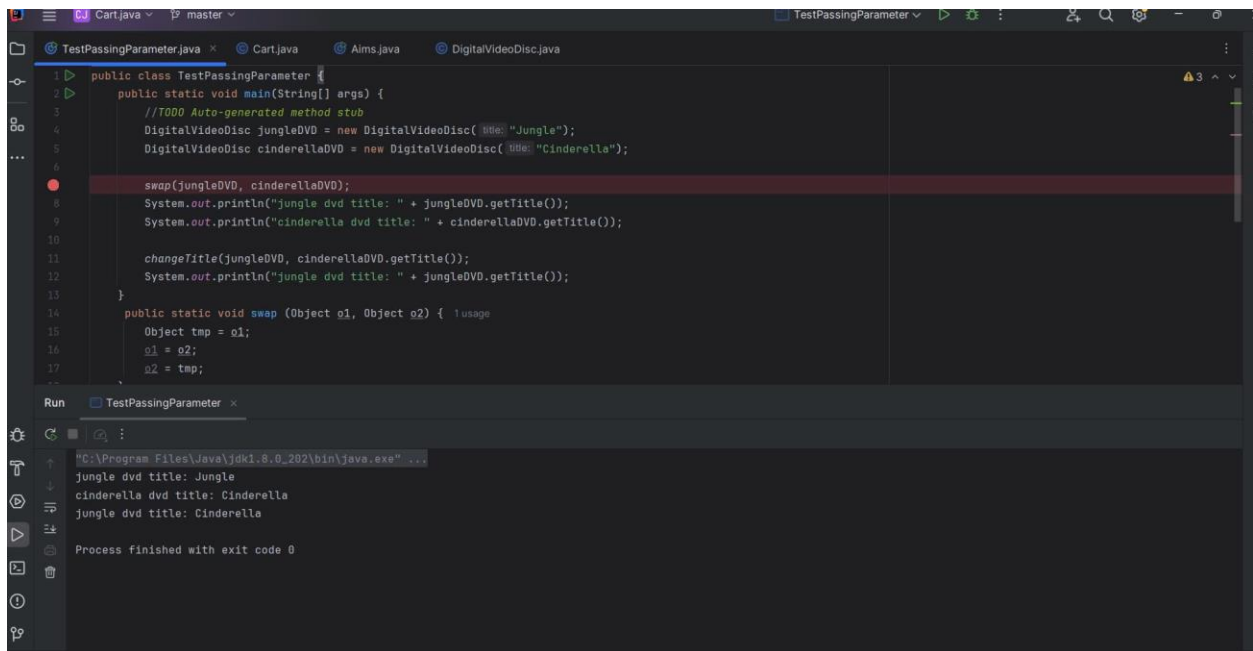
8. String, StringBuilder and StringBuffer

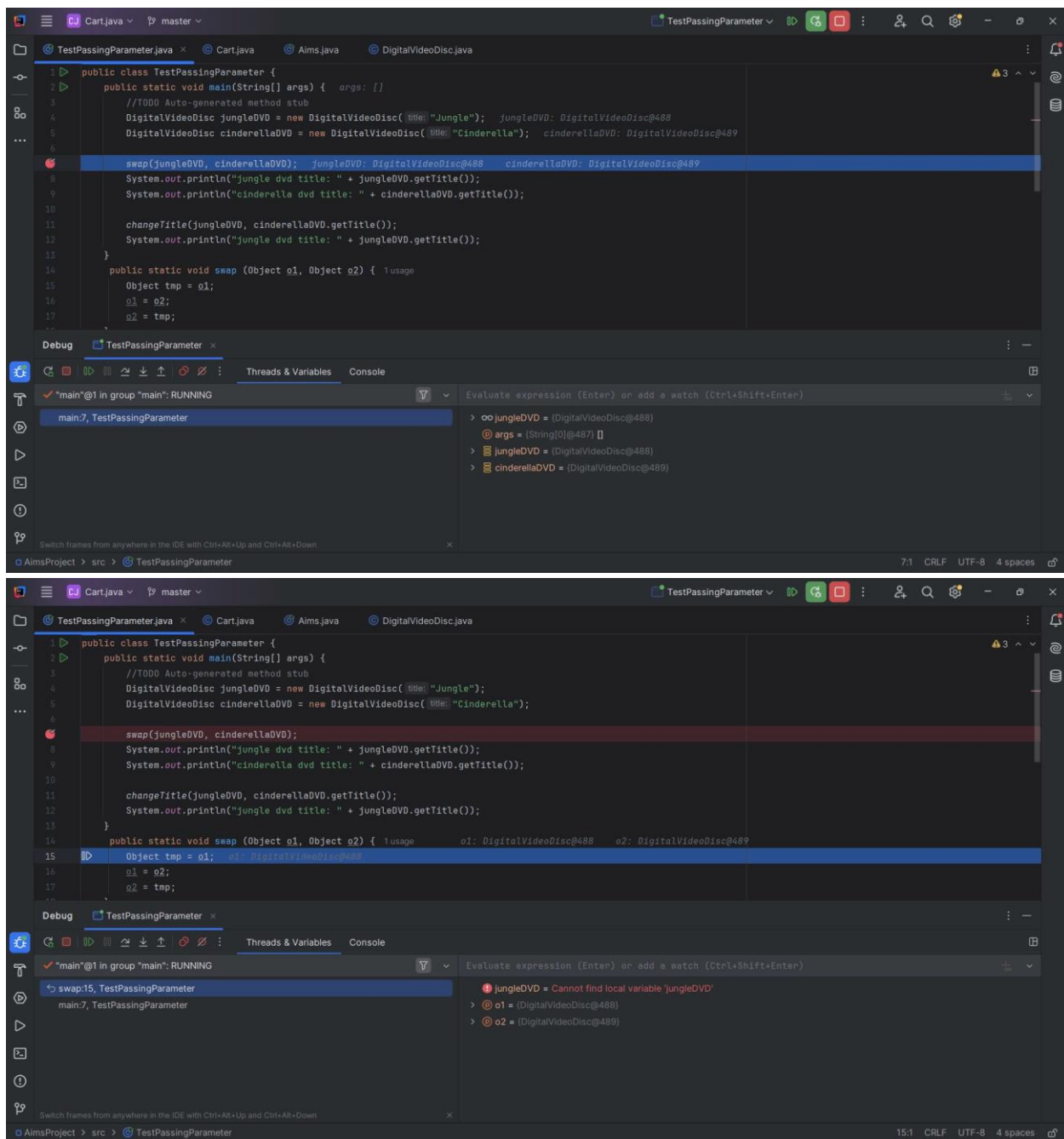
```
NoGarbage.java ConcatenationInLoops.java GarbageCreator.java ×
1 package hust.soict.cybersec.garbage;
2
3 import java.io.IOException;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 public class GarbageCreator { no usages 1 Kainic0804 *
8     public static void main(String[] args) { no usages 1 Kainic0804 *
9         String filename = "D:/00PLabs/OtherProjects/hust/soict/cybersec/garbage/test/txt";
10        byte[] inputBytes = {0};
11        long startTime, endTime;
12
13        try {
14            inputBytes = Files.readAllBytes(Paths.get(filename));
15        } catch (IOException e) {
16            e.printStackTrace();
17        }
18
19        startTime = System.currentTimeMillis();
20        String outputString = "";
21        for (byte b: inputBytes) {
22            outputString += (char)b;
23        }
24        endTime = System.currentTimeMillis();
25        System.out.println(endTime - startTime); // This print around 2046
26    }
27 }
28
```

```
NoGarbage.java DigitalVideoDisc.java ConcatenationInLoops.java × GarbageCreator.java
1 package hust.soict.cybersec.garbage;
2
3 import java.util.Random;
4
5 public class ConcatenationInLoops { no usages 1 Kainic0804
6     public static void main(String[] args) { no usages 1 Kainic0804
7         Random r = new Random(123);
8         long start = System.currentTimeMillis();
9         String s = "";
10        for (int i = 0; i < 65536; i++)
11            s += r.nextInt(2);
12        System.out.println(System.currentTimeMillis() - start); // This print around 400
13
14        r = new Random(123);
15        start = System.currentTimeMillis();
16        StringBuilder sb = new StringBuilder();
17        for (int i = 0; i < 65536; i++)
18            sb.append(r.nextInt(2));
19        s = sb.toString();
20        System.out.println(System.currentTimeMillis() - start); // This print around 2
21    }
22 }
23
```



II. Screenshot of code debugging and the results





TestPassingParameter.java | Cart.java | Aims.java | DigitalVideoDisc.java

```
1 public class TestPassingParameter {
2     public static void main(String[] args) {
3         //TODO Auto-generated method stub
4         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
5         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
6
7         swap(jungleDVD, cinderellaDVD);
8         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
9         System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
10
11         changeTitle(jungleDVD, cinderellaDVD.getTitle());
12         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
13     }
14     public static void swap (Object o1, Object o2) { 1 usage      o1: DigitalVideoDisc@488  o2: DigitalVideoDisc@489
15         Object tmp = o1; tmp: DigitalVideoDisc@488
16         o1 = o2; o1: DigitalVideoDisc@488  o2: DigitalVideoDisc@489
17         o2 = tmp;
18     }
```

Debug | TestPassingParameter

Threads & Variables | Console

✓ "main"@1 in group "main": RUNNING

swap:16, TestPassingParameter
main:7, TestPassingParameter

! jungleDVD = Cannot find local variable 'jungleDVD'

- > o1 = (DigitalVideoDisc@488)
- > o2 = (DigitalVideoDisc@489)
- > tmp = (DigitalVideoDisc@488)

Switch frames from anywhere in the IDE with Ctrl+Alt+Up and Ctrl+Alt+Down

AlmsProject > src > TestPassingParameter 16:1 CRLF UTF-8 4 spaces

TestPassingParameter.java | Cart.java | Aims.java | DigitalVideoDisc.java

```
1 public class TestPassingParameter {
2     public static void main(String[] args) {
3
4     }
5
6     public static void swap (Object o1, Object o2) { 1 usage      o1: DigitalVideoDisc@489  o2: DigitalVideoDisc@488
7         Object tmp = o1; tmp: DigitalVideoDisc@488
8         o1 = o2; o1: DigitalVideoDisc@489
9         o2 = tmp; o2: DigitalVideoDisc@489 tmp: DigitalVideoDisc@488
10    }
11    /* Swap method code can correctly swap two objects
12    public static void swap (DigitalVideoDisc o1, DigitalVideoDisc o2){
13        String tmp = o1.getTitle();
14        o1.setTitle(o2.getTitle());
15        o2.setTitle(tmp);
16    }
17    */
18    public static void changeTitle (DigitalVideoDisc dvd, String title){ 1 usage
19        String oldTitle = dvd.getTitle();
20        dvd <=+title(+title>
21    }
22 }
```

Debug | TestPassingParameter

Threads & Variables | Console

✓ "main"@1 in group "main": RUNNING

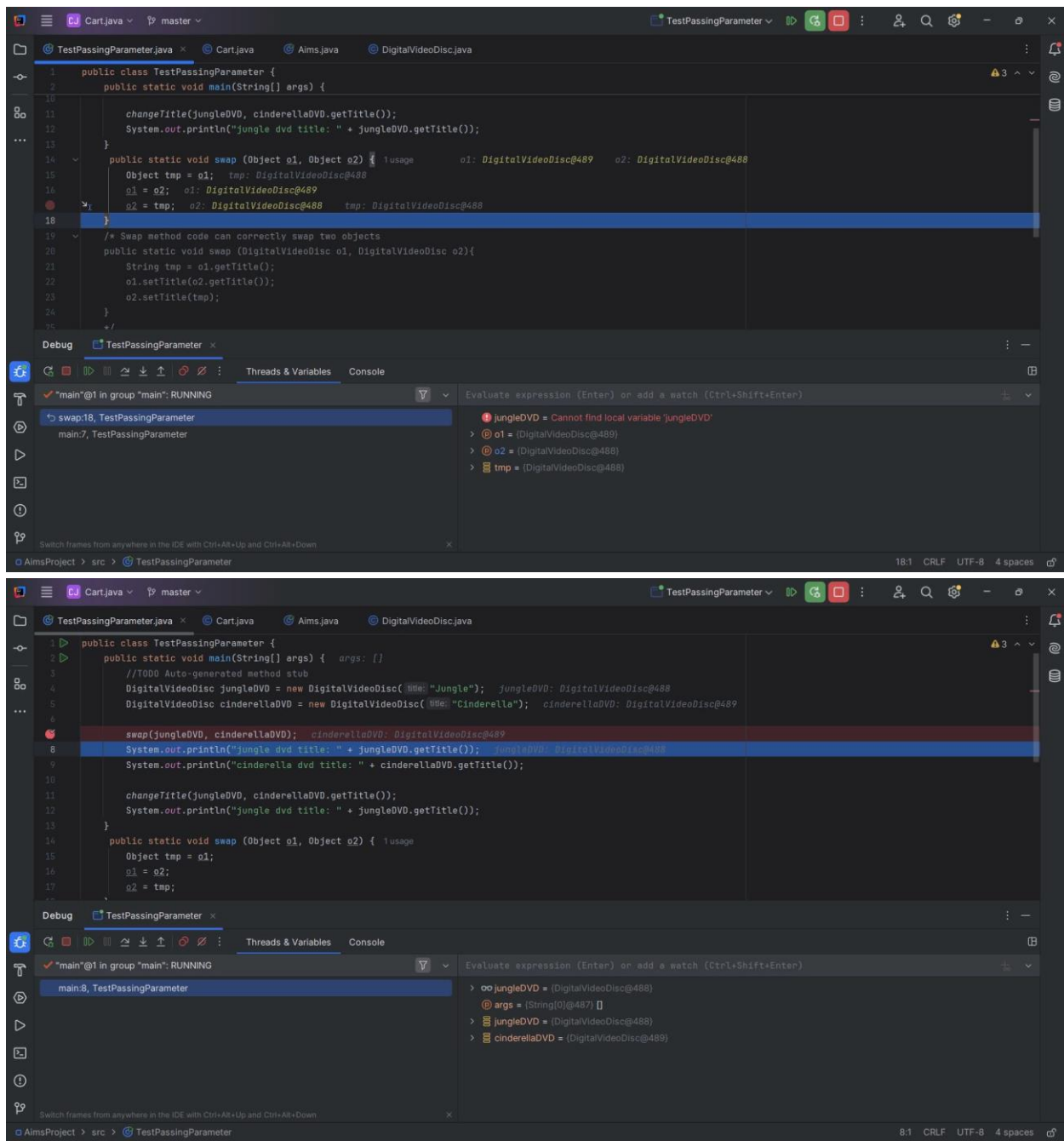
swap:17, TestPassingParameter
main:7, TestPassingParameter

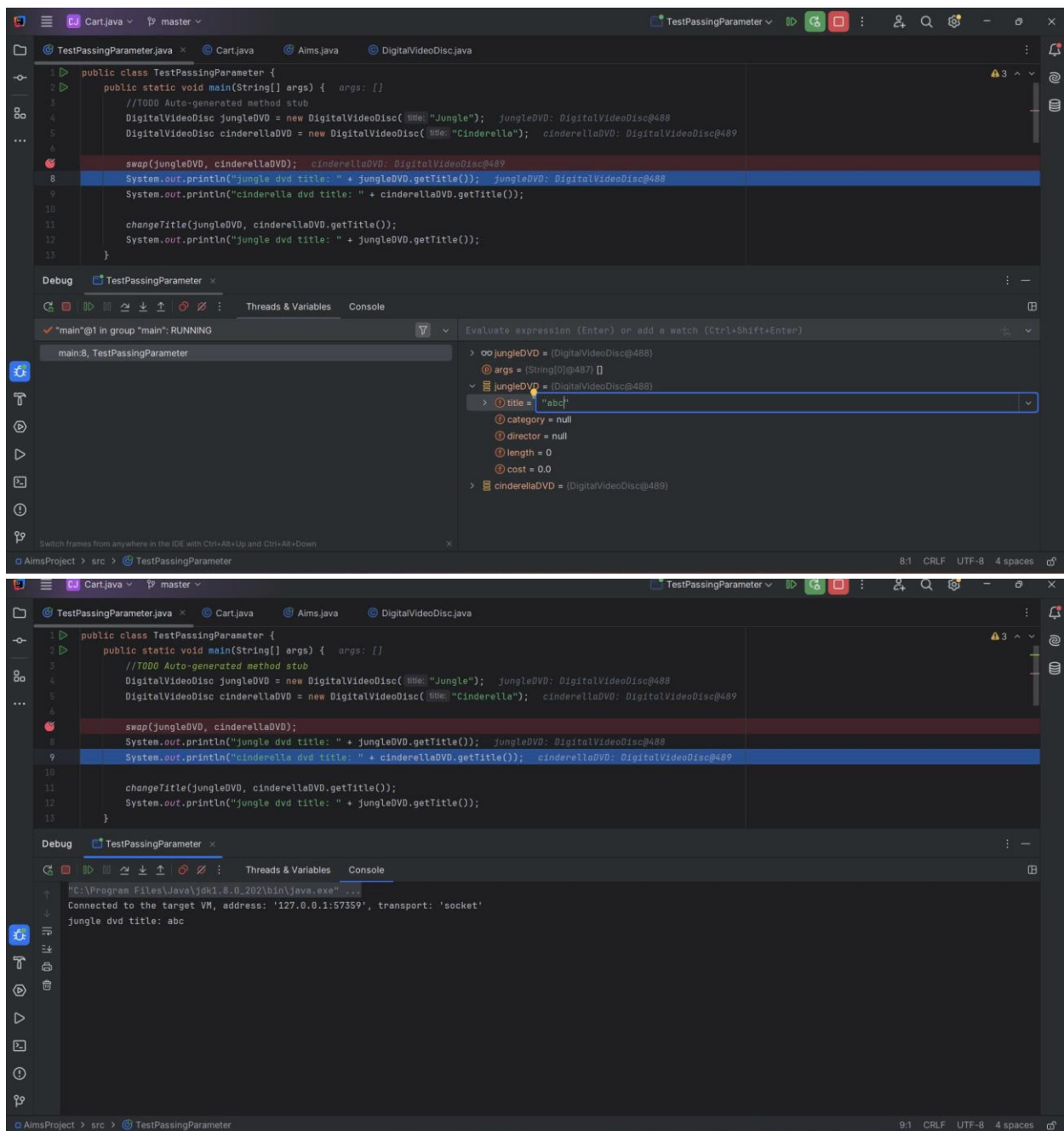
! jungleDVD = Cannot find local variable 'jungleDVD'

- > o1 = (DigitalVideoDisc@489)
- > o2 = (DigitalVideoDisc@488)
- > tmp = (DigitalVideoDisc@488)

Switch frames from anywhere in the IDE with Ctrl+Alt+Up and Ctrl+Alt+Down

AlmsProject > src > TestPassingParameter 17:1 CRLF UTF-8 4 spaces





III. Use-case diagram and class diagram

IV. Answer the questions

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

Java is pass by Value. We can see after call of `swap(jungleDVD, cinderellaDVD)`, the value of their references are copied into the parameters `o1` and `o2` respectively. Inside the `swap` method, the reference `o1` and `o2` are swapped. However, this only affects the local variables `o1` and `o2` within the `swap` method. The original references `jungleDVD` and `cinderellaDVD` in the main method are not modified, and that's why the titles remain unchanged in the main method after calling `swap(jungleDVD, cinderellaDVD)`.

- ***After the call of `swap(jungleDVD, cinderellaDVD)` why does the title of these two objects still remain?***

The value of their references are copied into the parameters `o1` and `o2` respectively. Inside the `swap` method, the reference `o1` and `o2` are swapped but outside the original references `jungleDVD` and `cinderellaDVD` in the main method not swapped at all, so title of these two objects still remain.

- ***After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())` why is the title of the `JungleDVD` changed?***

After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())`, the title of the `JungleDVD` changed. The reason is that Java object variables are simply references that point to real objects in the memory heap. Therefore, even though Java passes parameters to methods by value, if the variable points to an object reference, the real object will also be changed.