

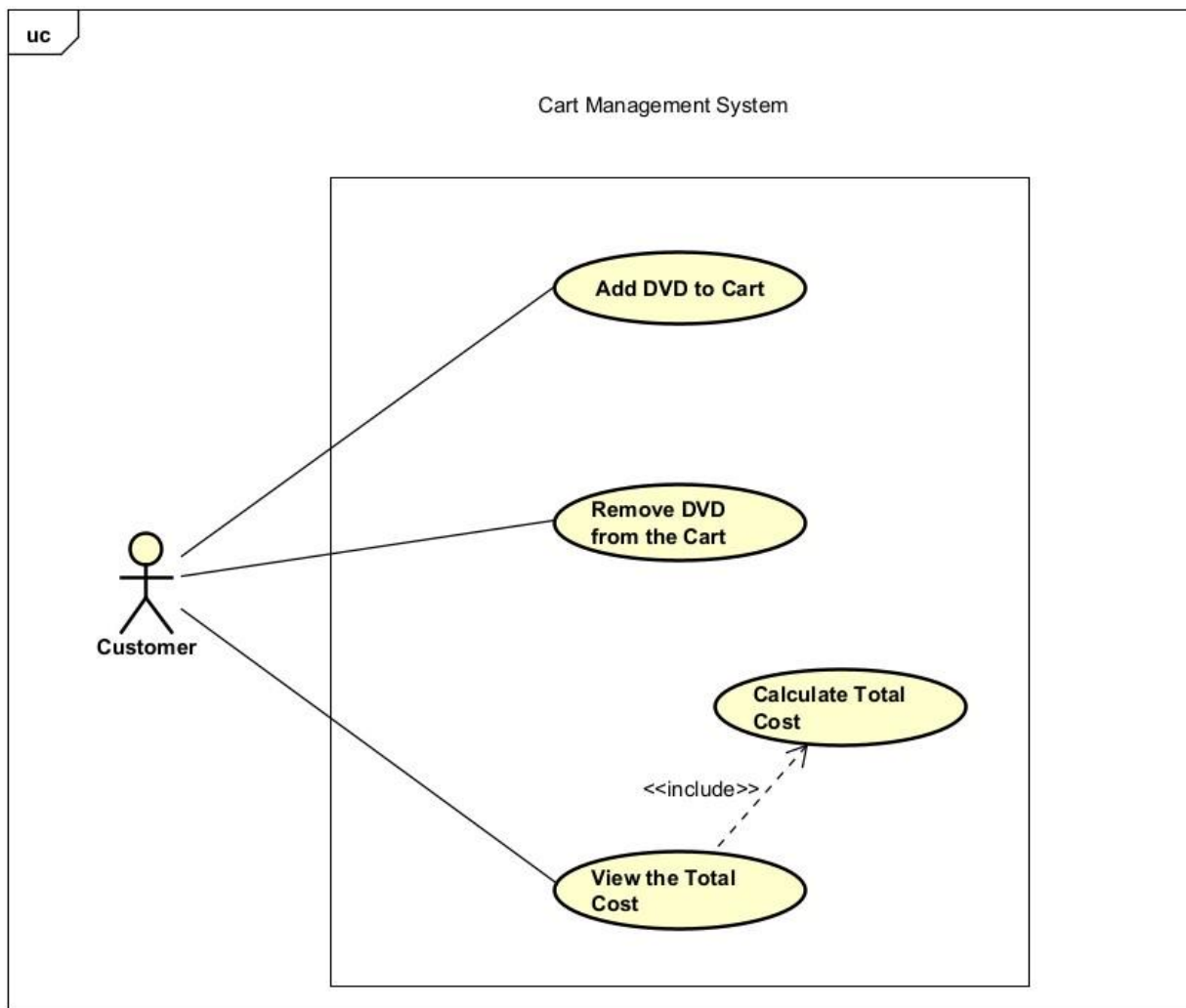
IT3100E Object-Oriented Programming Lab Report

Name: Nguyễn Đức Anh

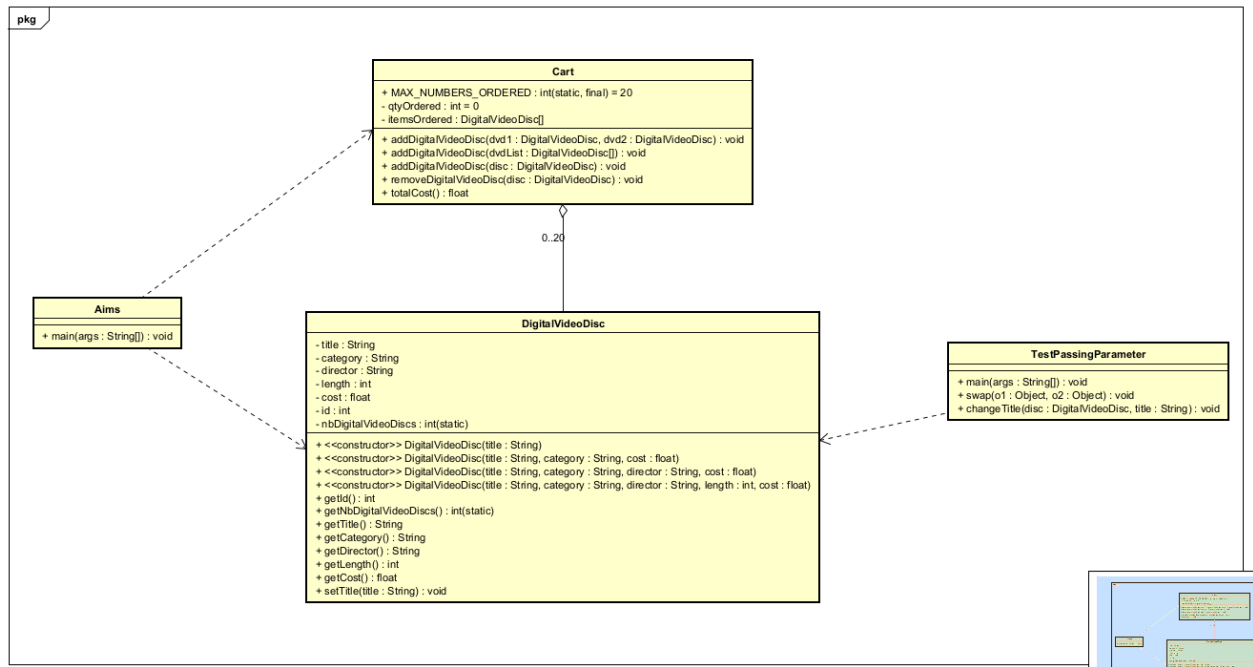
Student ID: 20235890

5, 6: UML UseCase Diagram and Class Diagram for use cases related to cart management.

- UseCase Diagram:**

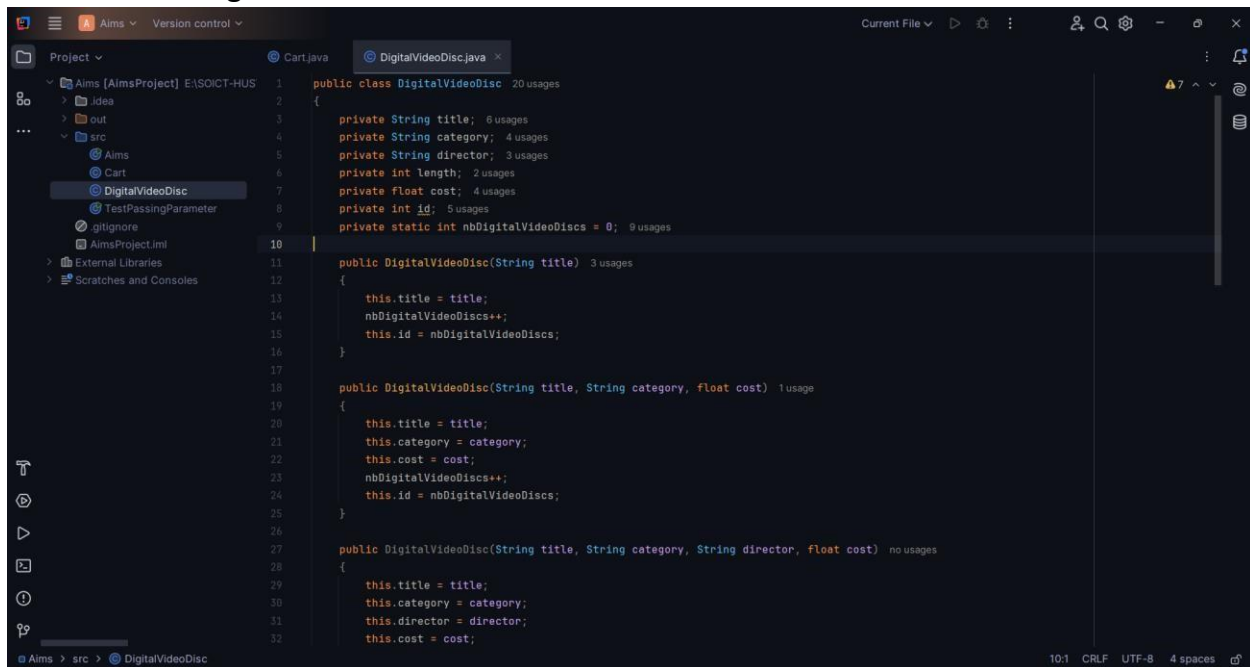


- **Class Diagram:**

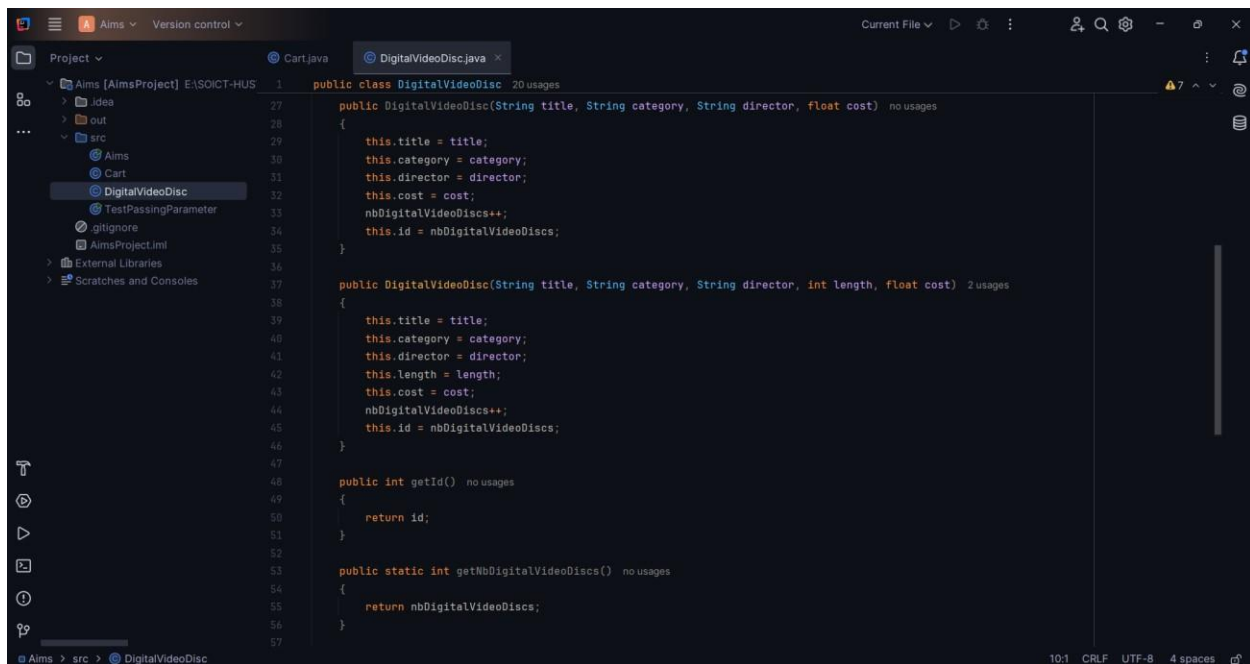


7, 8, 9, ..., 16: AimsProject (Class, Attributes, Accessors, Mutators, Constructor, Overloading, Passing parameter, Classifier Member and Instance Member)

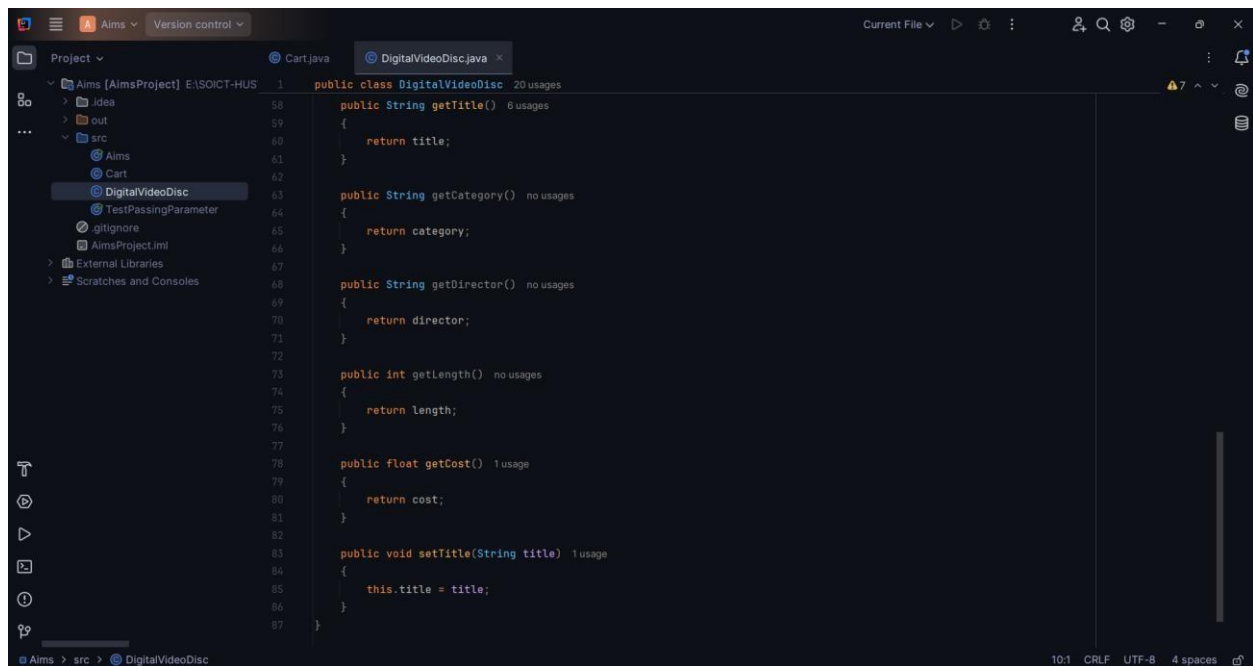
1. DigitalVideoDisc class:



```
1 public class DigitalVideoDisc 20 usages
2 {
3     private String title; 6 usages
4     private String category; 4 usages
5     private String director; 3 usages
6     private int length; 2 usages
7     private float cost; 4 usages
8     private int id; 5 usages
9     private static int nbDigitalVideoDiscs = 0; 0 usages
10
11     public DigitalVideoDisc(String title) 3 usages
12     {
13         this.title = title;
14         nbDigitalVideoDiscs++;
15         this.id = nbDigitalVideoDiscs;
16     }
17
18     public DigitalVideoDisc(String title, String category, float cost) 1 usage
19     {
20         this.title = title;
21         this.category = category;
22         this.cost = cost;
23         nbDigitalVideoDiscs++;
24         this.id = nbDigitalVideoDiscs;
25     }
26
27     public DigitalVideoDisc(String title, String category, String director, float cost) no usages
28     {
29         this.title = title;
30         this.category = category;
31         this.director = director;
32         this.cost = cost;
```

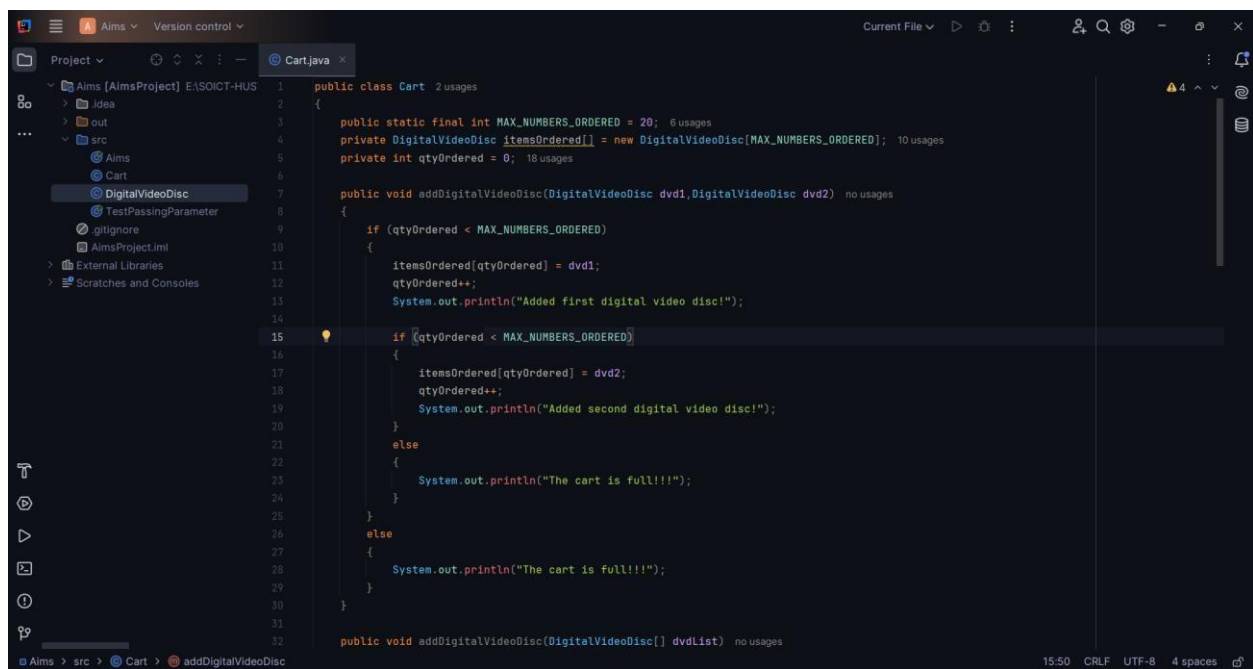


```
27     public DigitalVideoDisc(String title, String category, String director, float cost) no usages
28     {
29         this.title = title;
30         this.category = category;
31         this.director = director;
32         this.cost = cost;
33         nbDigitalVideoDiscs++;
34         this.id = nbDigitalVideoDiscs;
35     }
36
37     public DigitalVideoDisc(String title, String category, String director, int length, float cost) 2 usages
38     {
39         this.title = title;
40         this.category = category;
41         this.director = director;
42         this.length = length;
43         this.cost = cost;
44         nbDigitalVideoDiscs++;
45         this.id = nbDigitalVideoDiscs;
46     }
47
48     public int getId() no usages
49     {
50         return id;
51     }
52
53     public static int getNbDigitalVideoDiscs() no usages
54     {
55         return nbDigitalVideoDiscs;
56     }
57 }
```



```
1 public class DigitalVideoDisc 20 usages
58 public String getTitle() 6 usages
59 {
60     return title;
61 }
62
63 public String getCategory() no usages
64 {
65     return category;
66 }
67
68 public String getDirector() no usages
69 {
70     return director;
71 }
72
73 public int getLength() no usages
74 {
75     return length;
76 }
77
78 public float getCost() 1 usage
79 {
80     return cost;
81 }
82
83 public void setTitle(String title) 1 usage
84 {
85     this.title = title;
86 }
87 }
```

2. Cart class:



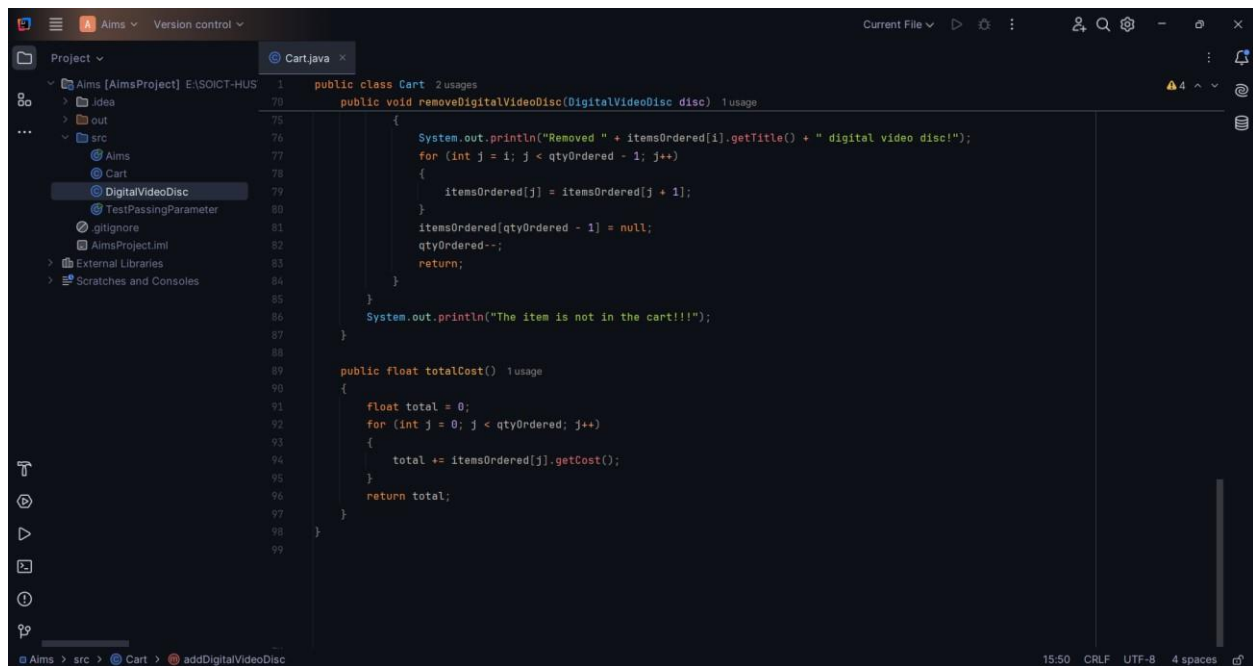
```
1 public class Cart 2 usages
2 {
3     public static final int MAX_NUMBERS_ORDERED = 20; 6 usages
4     private DigitalVideoDisc itemsOrdered[] = new DigitalVideoDisc[MAX_NUMBERS_ORDERED]; 10 usages
5     private int qtyOrdered = 0; 18 usages
6
7     public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) no usages
8     {
9         if (qtyOrdered < MAX_NUMBERS_ORDERED)
10         {
11             itemsOrdered[qtyOrdered] = dvd1;
12             qtyOrdered++;
13             System.out.println("Added first digital video disc!");
14
15             if (qtyOrdered < MAX_NUMBERS_ORDERED)
16             {
17                 itemsOrdered[qtyOrdered] = dvd2;
18                 qtyOrdered++;
19                 System.out.println("Added second digital video disc!");
20             }
21             else
22             {
23                 System.out.println("The cart is full!!!");
24             }
25         }
26         else
27         {
28             System.out.println("The cart is full!!!");
29         }
30     }
31
32     public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) no usages
33 }
```

This screenshot shows the IDE interface with the `Cart.java` file open. The left sidebar displays the project structure, including the `src` directory and the `DigitalVideoDisc` class. The main editor area shows the `Cart` class with two methods: `addDigitalVideoDisc(DigitalVideoDisc[] dvdList)` and `addDigitalVideoDisc(DigitalVideoDisc disc)`. The first method checks if the `dvdList` is null and returns if it is. It then iterates over the `dvdList` and adds each disc to the `itemsOrdered` array if it is not null and the `qtyOrdered` is less than `MAX_NUMBERS_ORDERED`. The second method adds a single disc to the `itemsOrdered` array if the `qtyOrdered` is less than `MAX_NUMBERS_ORDERED`.

```
1 public class Cart 2 usages
32 public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) no usages
33 {
34     if (dvdList == null)
35     {
36         System.out.println("DVD list is empty!");
37         return;
38     }
39
40     for (DigitalVideoDisc disc : dvdList)
41     {
42         if (disc != null && qtyOrdered < MAX_NUMBERS_ORDERED)
43         {
44             itemsOrdered[qtyOrdered] = disc;
45             qtyOrdered++;
46             System.out.println("Added a digital video disc!");
47         }
48         else if (qtyOrdered == MAX_NUMBERS_ORDERED)
49         {
50             System.out.println("The cart is full!!!");
51             return;
52         }
53     }
54 }
55
56 public void addDigitalVideoDisc(DigitalVideoDisc disc) 3 usages
57 {
58     if (qtyOrdered < MAX_NUMBERS_ORDERED)
59     {
60         itemsOrdered[qtyOrdered] = disc;
61         qtyOrdered++;
62         System.out.println("Added a digital video disc!");
63     }
```

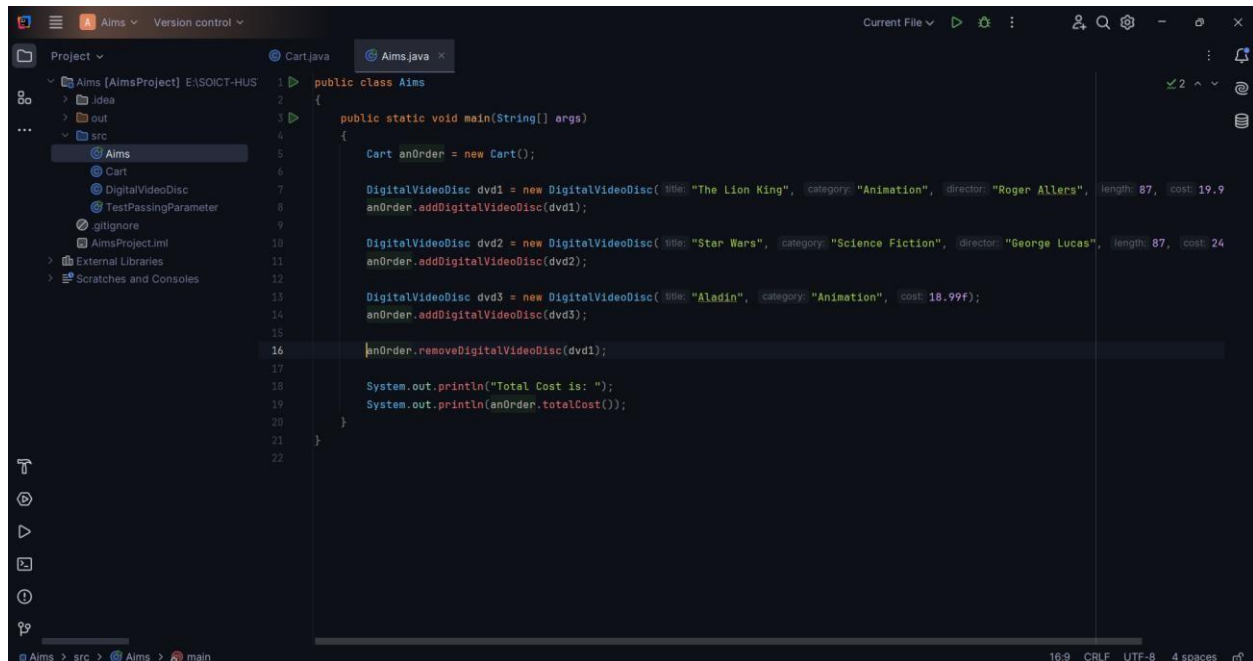
This screenshot shows the IDE interface with the `Cart.java` file open. The left sidebar displays the project structure, including the `src` directory and the `DigitalVideoDisc` class. The main editor area shows the `Cart` class with two methods: `addDigitalVideoDisc(DigitalVideoDisc disc)` and `removeDigitalVideoDisc(DigitalVideoDisc disc)`. The first method adds a single disc to the `itemsOrdered` array if the `qtyOrdered` is less than `MAX_NUMBERS_ORDERED`. The second method removes a disc from the `itemsOrdered` array if it is present. It iterates over the `itemsOrdered` array and finds the index of the disc to be removed. It then shifts all elements from that index to the right by one position and sets the last element to null. The `qtyOrdered` is decremented and the method returns.

```
56 public void addDigitalVideoDisc(DigitalVideoDisc disc) 3 usages
57 {
58     if (qtyOrdered < MAX_NUMBERS_ORDERED)
59     {
60         itemsOrdered[qtyOrdered] = disc;
61         qtyOrdered++;
62         System.out.println("Added a digital video disc!");
63     }
64     else
65     {
66         System.out.println("The cart is full!!!");
67     }
68 }
69
70 public void removeDigitalVideoDisc(DigitalVideoDisc disc) 1 usage
71 {
72     for (int i = 0; i < qtyOrdered; i++)
73     {
74         if (itemsOrdered[i] == disc)
75         {
76             System.out.println("Removed " + itemsOrdered[i].getTitle() + " digital video disc!");
77             for (int j = i; j < qtyOrdered - 1; j++)
78             {
79                 itemsOrdered[j] = itemsOrdered[j + 1];
80             }
81             itemsOrdered[qtyOrdered - 1] = null;
82             qtyOrdered--;
83             return;
84         }
85     }
86     System.out.println("The item is not in the cart!!!");
87 }
```



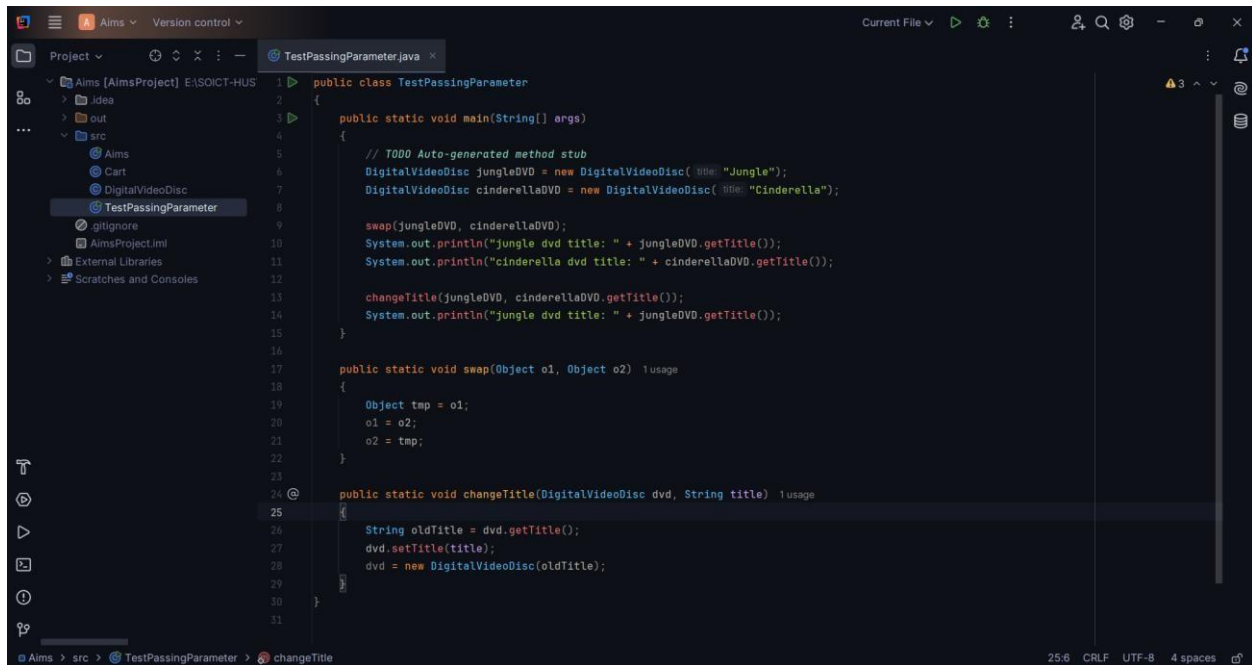
```
1 public class Cart {
2     public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
3         for (int j = itemsOrdered.length - 1; j >= 0; j--) {
4             if (itemsOrdered[j].getTitle().equals(disc.getTitle())) {
5                 itemsOrdered[j] = itemsOrdered[j + 1];
6             }
7         }
8         itemsOrdered[itemsOrdered.length - 1] = null;
9         return;
10    }
11    System.out.println("The item is not in the cart!!!");
12
13    public float totalCost() {
14        float total = 0;
15        for (int j = 0; j < itemsOrdered.length; j++) {
16            total += itemsOrdered[j].getCost();
17        }
18        return total;
19    }
20 }
```

3. Aims class:



```
1 public class Aims {
2     public static void main(String[] args) {
3         Cart anOrder = new Cart();
4
5         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.9);
6         anOrder.addDigitalVideoDisc(dvd1);
7
8         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24);
9         anOrder.addDigitalVideoDisc(dvd2);
10
11         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladdin", "Animation", 87, 18.99f);
12         anOrder.addDigitalVideoDisc(dvd3);
13
14         anOrder.removeDigitalVideoDisc(dvd1);
15
16         System.out.println("Total Cost is: ");
17         System.out.println(anOrder.totalCost());
18     }
19 }
```

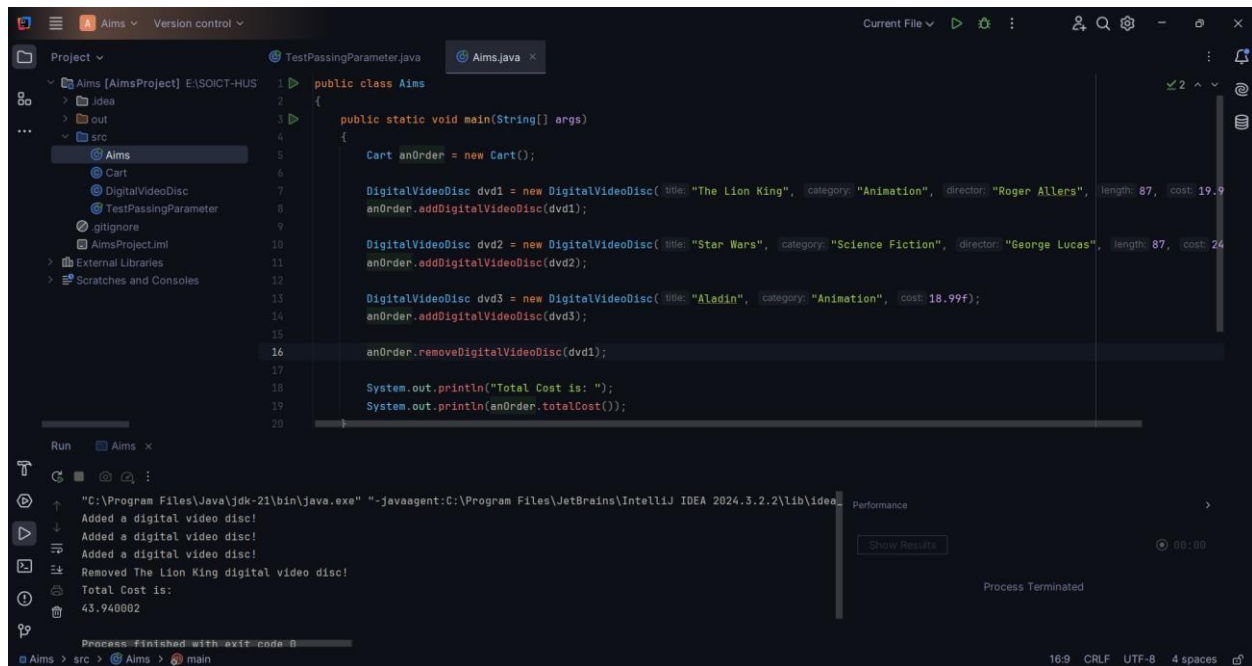
4. TestPassingParameter class:



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

Result:

- Running Aims:



```
1 public class Aims
2 {
3     public static void main(String[] args)
4     {
5         Cart anOrder = new Cart();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc( "The Lion King", category: "Animation", director: "Roger Allers", length: 87, cost: 19.9);
8         anOrder.addDigitalVideoDisc(dvd1);
9
10        DigitalVideoDisc dvd2 = new DigitalVideoDisc( "Star Wars", category: "Science Fiction", director: "George Lucas", length: 87, cost: 24);
11        anOrder.addDigitalVideoDisc(dvd2);
12
13        DigitalVideoDisc dvd3 = new DigitalVideoDisc( "Aladin", category: "Animation", cost: 18.99f);
14        anOrder.addDigitalVideoDisc(dvd3);
15
16        anOrder.removeDigitalVideoDisc(dvd1);
17
18        System.out.println("Total Cost is: ");
19        System.out.println(anOrder.totalCost());
20    }
21 }
```

Run Aims x

↑
↓
Added a digital video disc!
Added a digital video disc!
Added a digital video disc!
Removed The Lion King digital video disc!
Total Cost is:
43.940002

Process finished with **exit** code 0

Performance
Show Results
00:00
Process Terminated

- **Running TestPassingParameter:**

The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon). The left sidebar contains the Project view, showing the file structure of the 'Aims' project, with 'TestPassingParameter.java' selected. The main editor window shows the source code of 'TestPassingParameter.java'. The code includes a 'main' method that creates 'DigitalVideoDisc' objects for 'Jungle' and 'Cinderella', prints their titles, and calls a 'changeTitle' method. It also includes a 'swap' method. The bottom panel shows the 'Run' output, indicating that the process finished successfully with exit code 0. The output text shows the titles being printed: 'jungle dvd title: Jungle', 'cinderella dvd title: Cinderella', and 'jungle dvd title: Cinderella'. The status bar at the bottom indicates the file encoding is UTF-8 and the line ending is CRLF.

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

Run TestPassingParameter x

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.3.2\lib\idea_... Performance

jungle dvd title: Jungle
cinderella dvd title: Cinderella
jungle dvd title: Cinderella

Show Results 00:00

Process finished with exit code 0 Process Terminated

Aims > src > TestPassingParameter > changeTitle 25.6 CRLF UTF-8 4 spaces