

LAB 3

2. Working with method overloading:

2.1 Overloading by differing types of parameter:

```
93 // Method to add list new DVDs
94● public void addDigitalVideoDisc(DigitalVideoDisc []dvdlist) {
95     // If cart is full
96     if (qtyOrdered + dvdlist.length > 20) {
97         System.out.println("The cart is almost full");
98         return;
99     }
100
101     // Add to cart
102     System.arraycopy(dvdlist, 0, itemsOrdered, qtyOrdered, dvdlist.length);
103
104     // Increase the qtyOrdered
105     qtyOrdered += dvdlist.length;
106
107     // Notify
108     System.out.println("The list has been added");
109 }
```

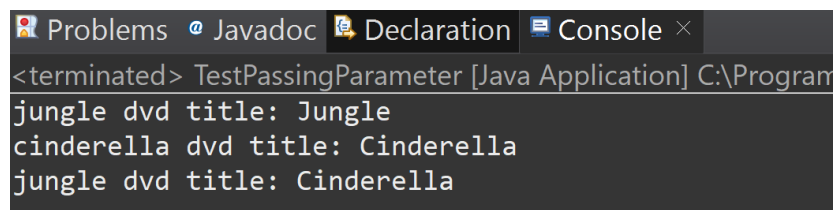
2.2. Overloading by differing the number of parameter:

```
111 // Method to add two new DVD
112● public void addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2) {
113     // If cart is full
114     if (qtyOrdered >= 19) {
115         System.out.println("The cart is almost full");
116         return;
117     }
118
119     // Increase the qtyOrdered
120     qtyOrdered += 2;
121
122     // Add to cart
123     itemsOrdered[qtyOrdered - 2] = dvd1;
124     itemsOrdered[qtyOrdered - 1] = dvd2;
125
126     // Notify
127     System.out.println("The disc has been added");
128 }
```

3. Passing parameter:

```
1 package Lab03.AimsProject;
2
3 public class TestPassingParameter {
4
5     static class DVDWrapper {
6         DigitalVideoDisc disc;
7
8         public DVDWrapper(DigitalVideoDisc disc) {
9             this.disc = disc;
10        }
11    }
12
13    public static void main(String[] args) {
14        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
15        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
16        DVDWrapper jungleDVDWrapper = new DVDWrapper(jungleDVD);
17        DVDWrapper cinderellaDVDWrapper = new DVDWrapper(cinderellaDVD);
18
19        swap(jungleDVDWrapper, cinderellaDVDWrapper);
20        System.out.println("jungle dvd title: " + jungleDVDWrapper.disc.getTitle());
21        System.out.println("cinderella dvd title: " + cinderellaDVDWrapper.disc.getTitle());
22
23        changeTitle(jungleDVD, cinderellaDVD.getTitle());
24        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
25    }
26
27
28    public static void swap(Object o1, Object o2) {
29        Object tmp = o1;
30        o1 = o2;
31        o2 = tmp;
32    }
33
34    public static void changeTitle(DigitalVideoDisc dvd, String title) {
35        String oldTitle = dvd.getTitle();
36        dvd.setTitle(title);
37        dvd = new DigitalVideoDisc(oldTitle);
38    }
39 }
```

→ Kết quả:



```
<terminated> TestPassingParameter [Java Application] C:\Program
jungle dvd title: Jungle
cinderella dvd title: Cinderella
jungle dvd title: Cinderella
```

Câu hỏi:

1. Is Java a Pass by Value or a Pass by Reference programming language?

→ Java là một ngôn ngữ lập trình Pass by Value.

→ Khi truyền tham số vào một phương thức, Java truyền bản sao của giá trị tham chiếu (không phải bản thân tham chiếu). Điều này có nghĩa là bạn không thể thay đổi tham chiếu gốc bên ngoài phương thức, nhưng bạn có thể thay đổi trạng thái của đối tượng mà tham chiếu đó trỏ tới.

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

→

- Khi bạn gọi swap(jungleDVD, cinderellaDVD), Java truyền **bản sao của giá trị tham chiếu** vào phương thức.

- Trong phương thức swap():

- dvd1 và dvd2 chỉ là **bản sao của tham chiếu gốc** (jungleDVD và cinderellaDVD).
- Khi hoán đổi dvd1 và dvd2, bạn chỉ thay đổi giá trị của các bản sao tham chiếu này bên trong phương thức, không làm thay đổi giá trị của các tham chiếu gốc bên ngoài.

- Kết quả: Các tham chiếu gốc jungleDVD và cinderellaDVD bên ngoài vẫn không thay đổi.

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

→

- Khi bạn gọi changeTitle(jungleDVD, cinderellaDVD.getTitle()), Java truyền **bản sao của giá trị tham chiếu** jungleDVD vào phương thức.

- Trong phương thức changeTitle():

- Tham số dvd trỏ đến cùng đối tượng trong bộ nhớ như jungleDVD.
- Khi bạn gọi dvd.setTitle(newTitle), bạn thực sự thay đổi thuộc tính title của đối tượng mà jungleDVD trỏ tới.

- Kết quả: Tiêu đề của đối tượng jungleDVD bị thay đổi.

4. Use debug run:

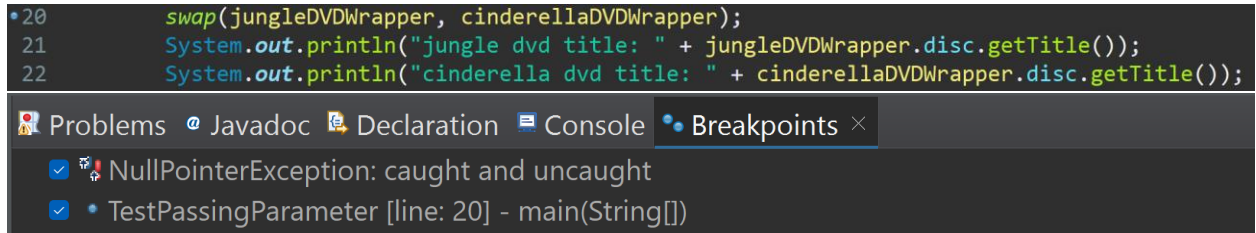
4.1. Debugging Java in Eclipse:

4.2. Example of debug run for the *swap* method of *TestPassingParameter*:

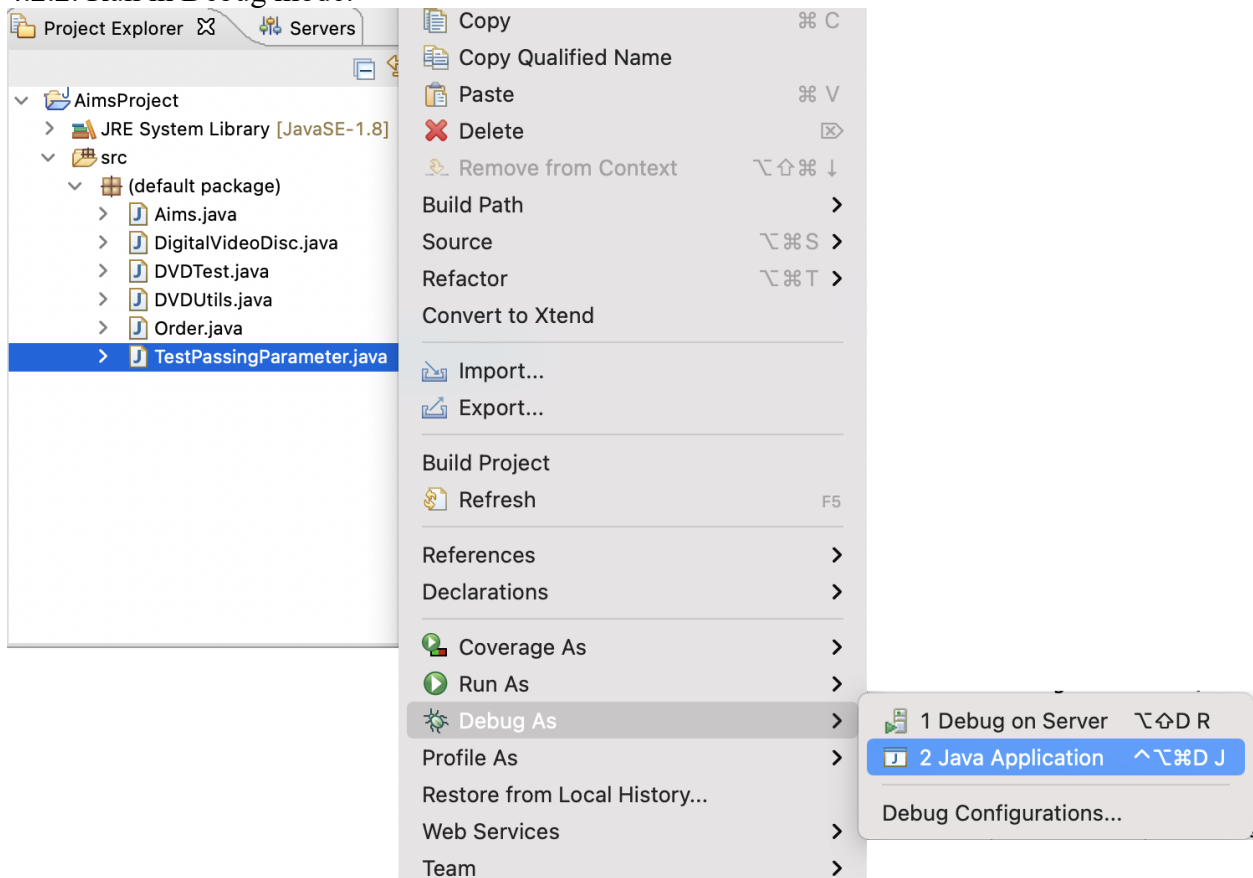
4.2.1. Setting, deleting & deactive breakpoint:

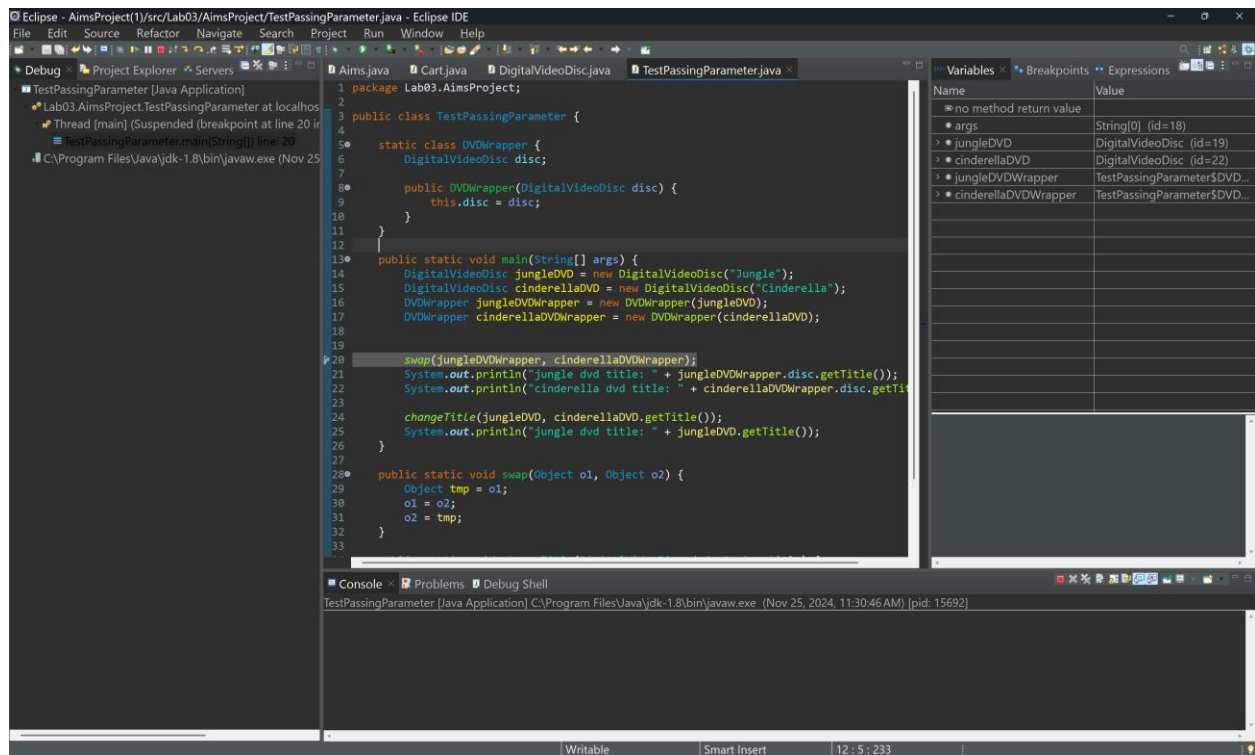
- Setting, deleting:

- Đặt con trỏ vào dòng muốn chọn.
- Ctrl + Shift + B / Double click bên lề trái



4.2.2. Run in Debug mode:

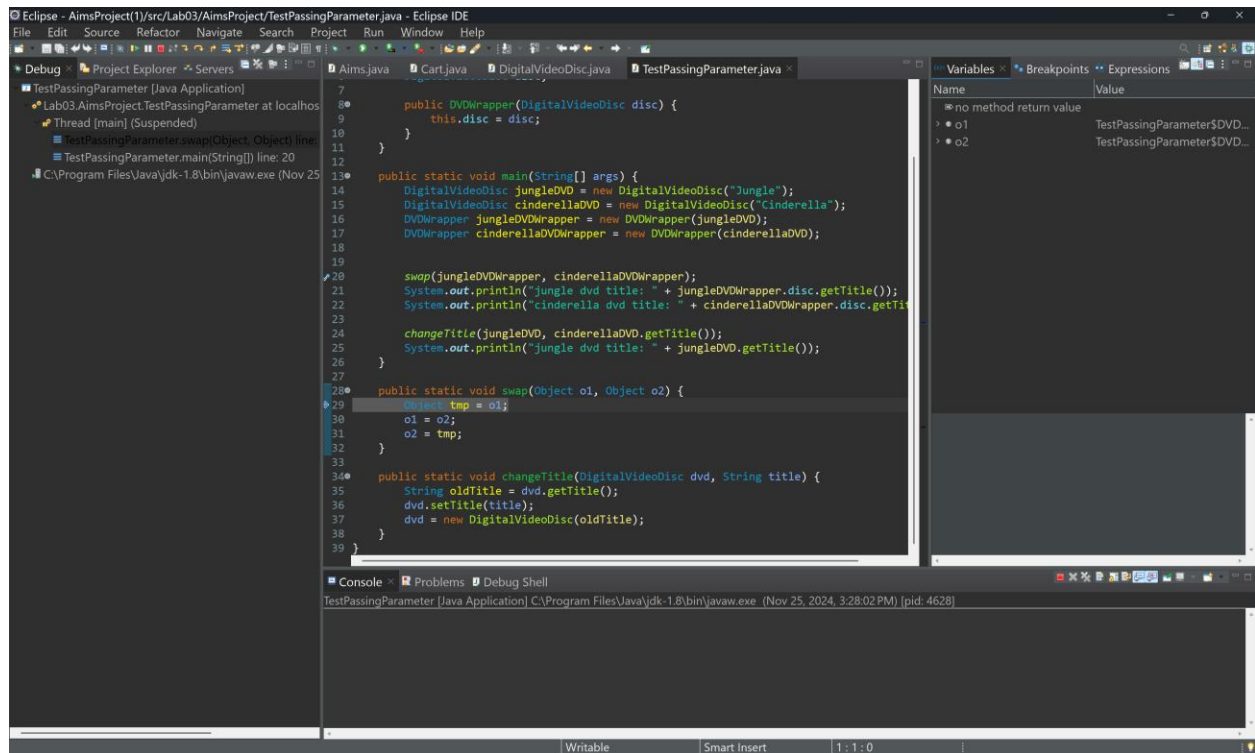




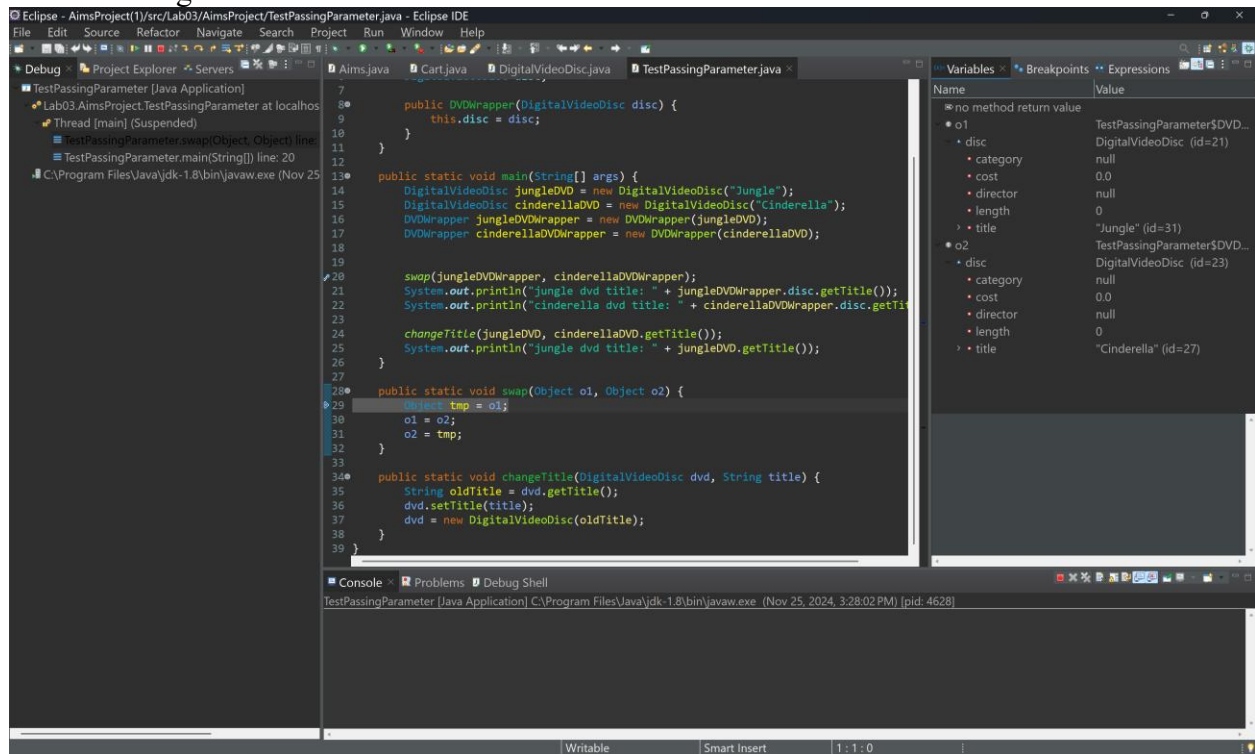
4.2.3. Step Into, Step Over, Step Return, Resume:

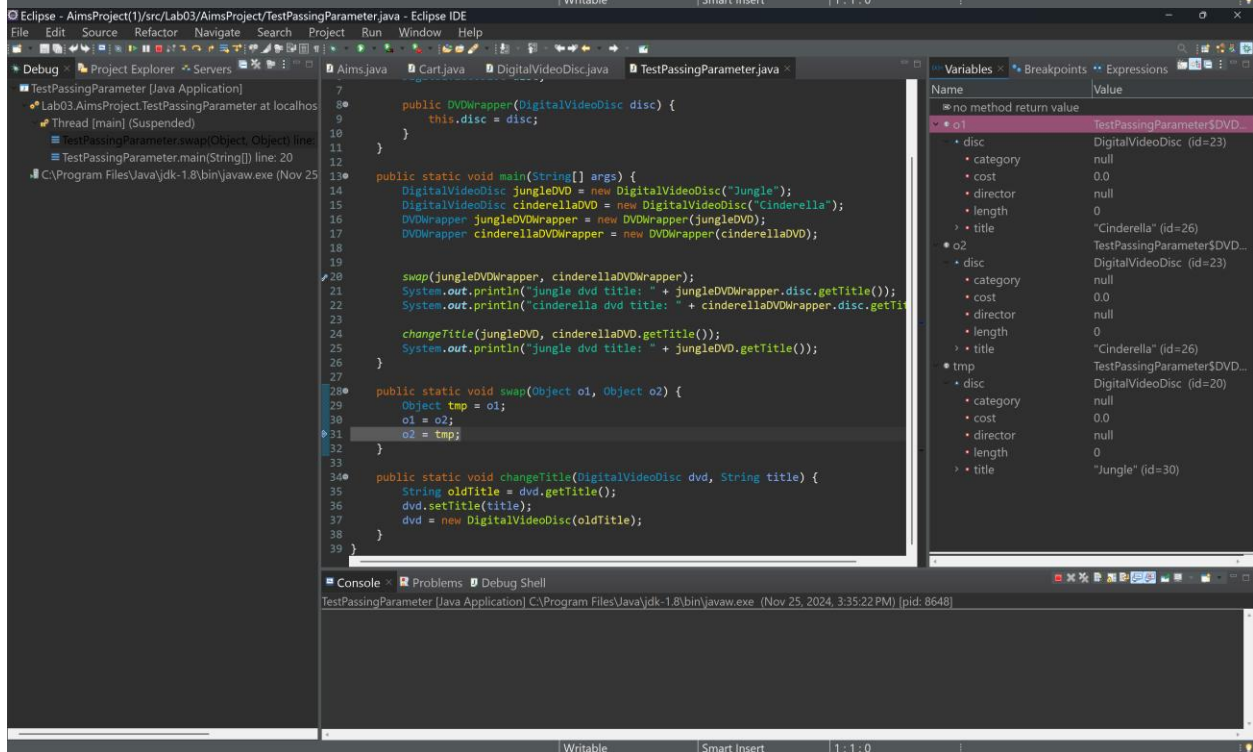
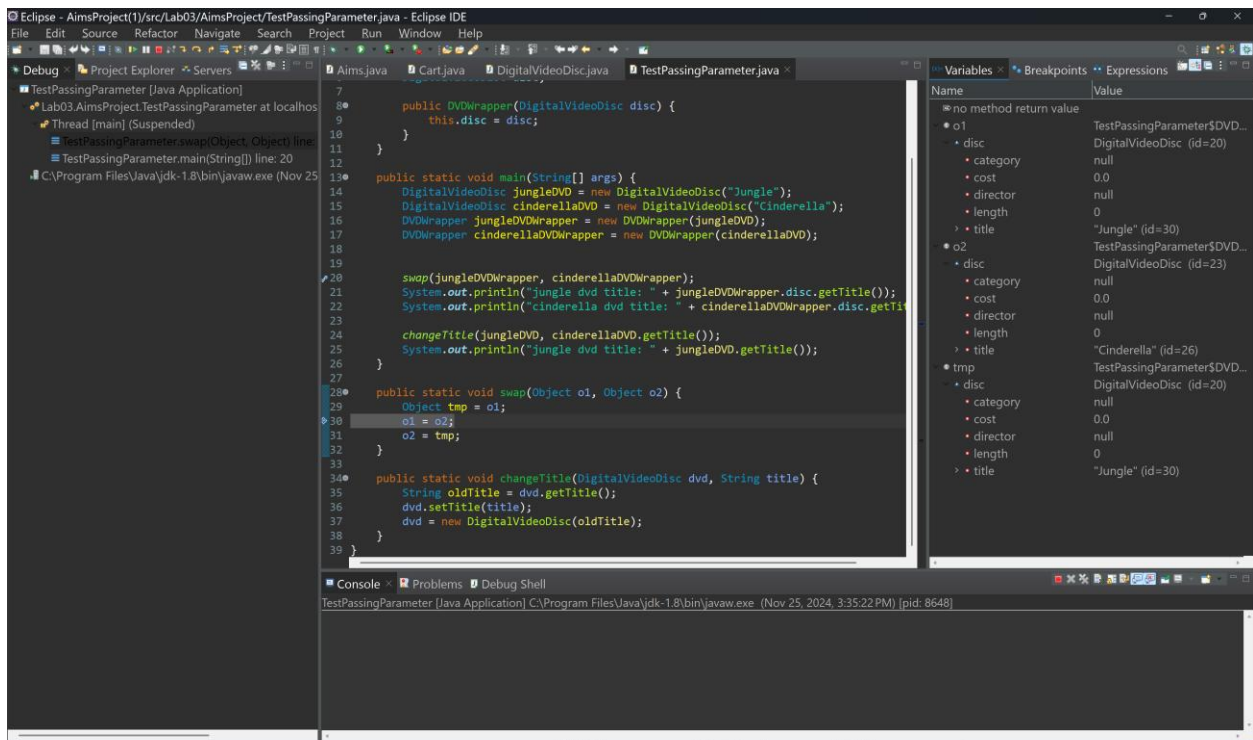


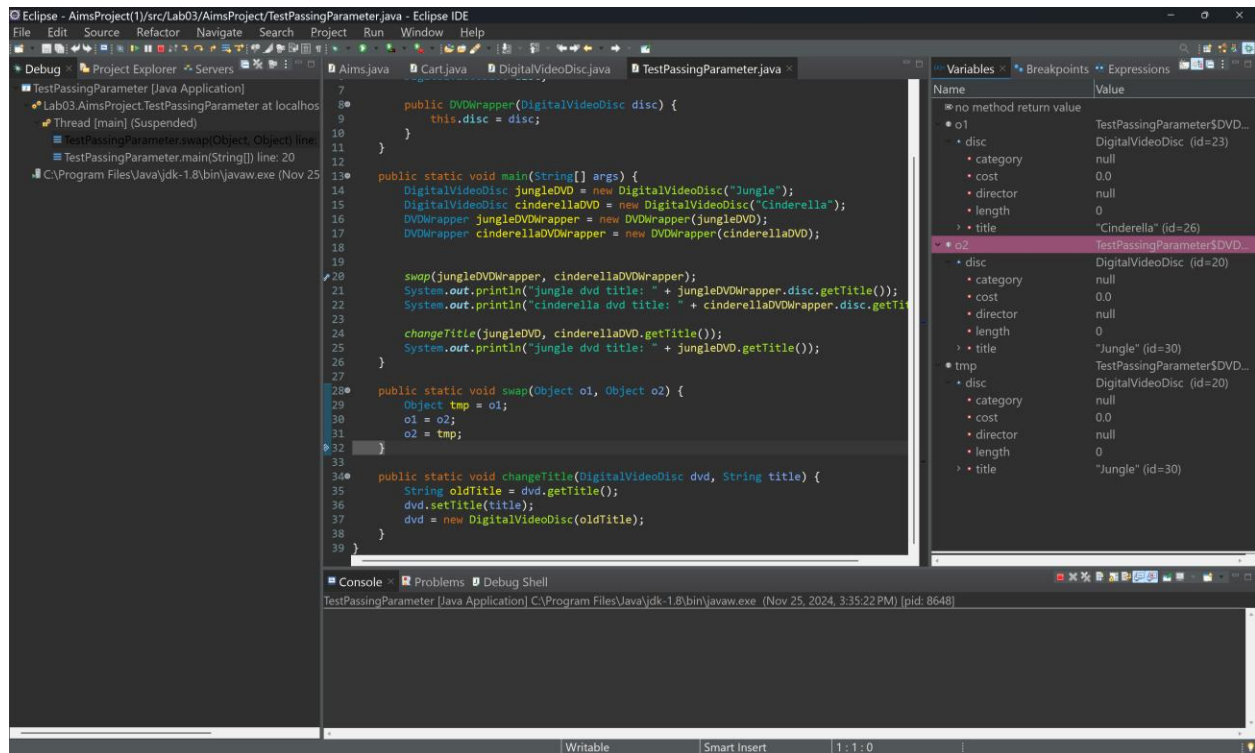
- Step Into (F5): trình gỡ lỗi bước vào hàm.
- Step Over (F6): chỉ di chuyển trình gỡ lỗi đến dòng tiếp theo trong cùng một hành động Java.
- Step Return (F7): hướng dẫn trình gỡ lỗi rời khỏi hàm → Lệnh ngược lại với Step Into.
- Resume (F8): hướng dẫn trình gỡ lỗi tiếp tục cho đến khi gặp một Breakpoint khác.



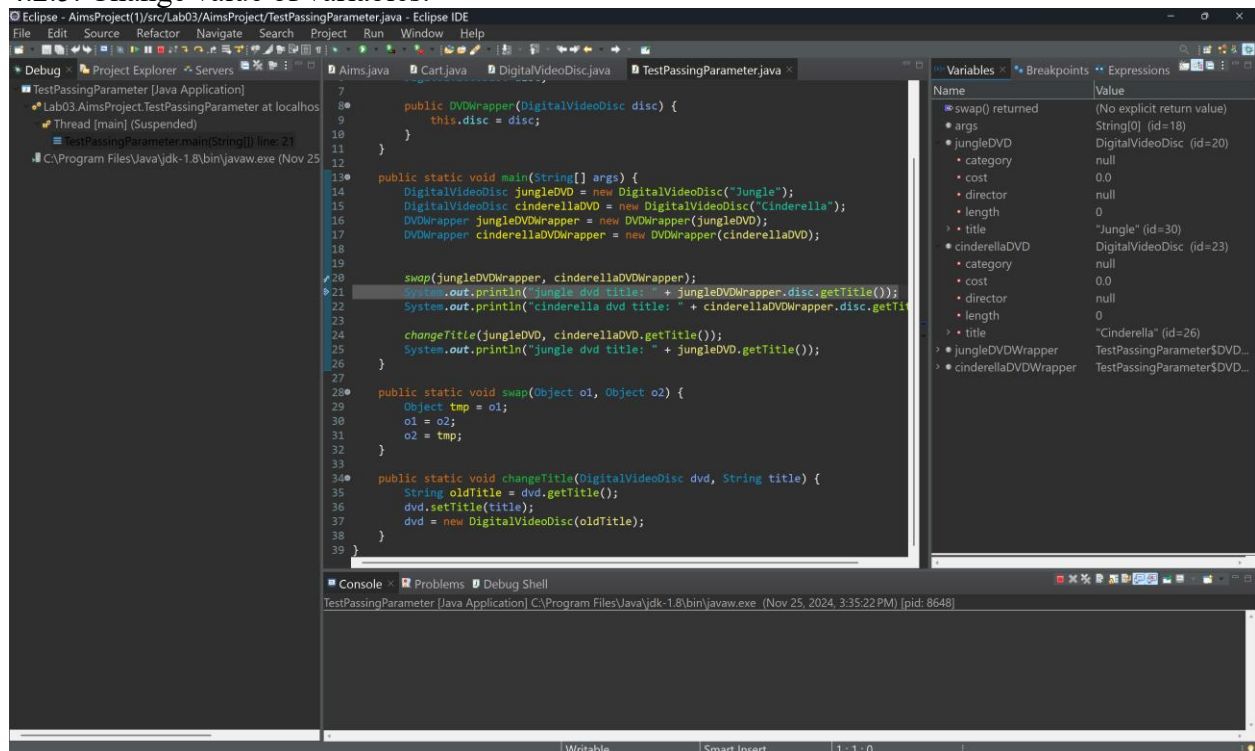
4.2.4. Investigate value of variables:







4.2.5. Change value of variables:



Name	Value	Name	Value
swap() returned	(No explicit return value)	swap() returned	(No explicit return value)
args	String[0] (id=18)	args	String[0] (id=18)
jungleDVD	DigitalVideoDisc (id=20)	jungleDVD	DigitalVideoDisc (id=20)
category	null	category	null
cost	0.0	cost	0.0
director	null	director	null
length	0	length	0
title	"Jungle" (id=30)	title	"abcd1234" (id=36)
cinderellaDVD	DigitalVideoDisc (id=23)	cinderellaDVD	DigitalVideoDisc (id=23)
jungleDVDWrapper	TestPassingParameter\$DVD...	jungleDVDWrapper	TestPassingParameter\$DVD...
cinderellaDVDWrapper	TestPassingParameter\$DVD...	cinderellaDVDWrapper	TestPassingParameter\$DVD...

→ Output:

```

Console × Problems Debug Shell
TestPassingParameter [Java Application] C:\Prog
jungle dvd title: abcd1234

Console × Problems Debug Shell
<terminated> TestPassingParameter [Java Applic
jungle dvd title: abcd1234
cinderella dvd title: Cinderella
jungle dvd title: Cinderella

```

5. Classifier Member and Instance Member:

- DigitalVideoDisc:

```
1 package Lab03.AimsProject;
2
3 public class DigitalVideoDisc {
4     private int id;
5     private String title;
6     private String category;
7     private String director;
8     private int length;
9     private float cost;
10
11     private static int nbDigitalVideoDiscs = 0;
12
13     // Constructor
14     public DigitalVideoDisc(String title, String category, float cost) {
15         this.id = ++nbDigitalVideoDiscs;
16         this.title = title;
17         this.category = category;
18         this.cost = cost;
19     }
20
21     public DigitalVideoDisc(String title, String category, String director, float cost) {
22         this.id = ++nbDigitalVideoDiscs;
23         this.title = title;
24         this.category = category;
25         this.director = director;
26         this.cost = cost;
27     }
28
29     public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
30         this.id = ++nbDigitalVideoDiscs;
31         this.title = title;
32         this.category = category;
33         this.director = director;
34         this.length = length;
35         this.cost = cost;
36     }
37
38     public DigitalVideoDisc(String title) {
39         this.title = title;
40         this.id = ++nbDigitalVideoDiscs;
41     }
42
43     // Method to print a DVD
44     public void printDVD() {
45         System.out.println(id + ". DVD - "
46             + title + " - "
47             + category + " - "
48             + director + " - "
49             + length + ": "
50             + cost + "$");
51     }
52
53     // Method to finds out if the corresponding disk is a match given the title.
54     public boolean isMatch(String title) {
55         return title.equals(this.title);
56     }
57 }
```

```

58 // Getter
59 public String getTitle() {
60     return title;
61 }
62 public String getCategory() {
63     return category;
64 }
65 public String getDirector() {
66     return director;
67 }
68 public int getLength() {
69     return length;
70 }
71 public float getCost() {
72     return cost;
73 }
74 public int getId() {
75     return id;
76 }
77 public static int getNbDigitalVideoDiscs() {
78     return nbDigitalVideoDiscs;
79 }
81 // Setter
82 public void setTitle(String title) {
83     this.title = title;
84 }
85 public void setCategory(String category) {
86     this.category = category;
87 }
88 public void setDirector(String director) {
89     this.director = director;
90 }
91 public void setLength(int length) {
92     this.length = length;
93 }
94 public void setCost(float cost) {
95     this.cost = cost;
96 }
97 public void setId(int id) {
98     this.id = id;
99 }
100 public static void setNbDigitalVideoDiscs(int nbDigitalVideoDiscs) {
101     DigitalVideoDisc.nbDigitalVideoDiscs = nbDigitalVideoDiscs;
102 }
103 }

```

- Aims:

```

1 package Lab03.AimsProject;
2
3 public class Aims {
4     public static void main(String[] args) {
5
6         // Create a new cart
7         Cart anOrder = new Cart();
8
9         // Create new DVD objects and add them to the cart
10        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
11        anOrder.addDigitalVideoDisc(dvd1);
12
13        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
14        anOrder.addDigitalVideoDisc(dvd2);
15
16        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
17        anOrder.addDigitalVideoDisc(dvd3);
18
19        System.out.println(anOrder.toString());
20
21        // Print number of DVD in cart
22        System.out.print("Number of DVD is: ");
23        System.out.println(DigitalVideoDisc.getNbDigitalVideoDiscs());
24
25        // Print id of dvd3
26        System.out.print("Id of dvd3 is: ");
27        System.out.println(dvd3.getId());
28
29        // Print total cost
30        System.out.print("Total Cost is: " + anOrder.totalCost());
31    }
32 }

```

→ Kết quả:

```

The disc has been added
The disc has been added
The disc has been added
Cart Details:
DVD ID: 1, Title: The Lion King, Cost: 19.95
DVD ID: 2, Title: Star Wars, Cost: 24.95
DVD ID: 3, Title: Aladin, Cost: 18.99

Number of DVD is: 3
Id of dvd3 is: 3
Total Cost is: 63.89

```

6. Open the Cart class:

- Method in Cart class:

```
130 // Method to print the list of ordered items of a cart, the price of each item, and the total price
131● public void printCart() {
132     System.out.println("*****CART*****");
133     System.out.println("Ordered Items:");
134     for (DigitalVideoDisc dvd : itemsOrdered) {
135         if (dvd != null)
136             dvd.printDVD();
137     }
138     System.out.println("Total cost: " + totalCost());
139     System.out.println("*****");
140 }
141
142 // Method to search for DVDs in the cart by ID and display the search results.
143● public void searchByID(int id) {
144     boolean found = false;
145     for (DigitalVideoDisc dvd: itemsOrdered) {
146         if (dvd != null && dvd.getId() == id) {
147             found = true;
148             dvd.printDVD();
149         }
150     }
151     if (!found) {
152         System.out.println("Not found!");
153     }
154 }
155
156 // Method to search for DVDs in the cart by title and print the results.
157● public void searchByTitle(String title) {
158     boolean found = false;
159     for (DigitalVideoDisc dvd: itemsOrdered) {
160         if (dvd != null && dvd.isMatch(title)) {
161             found = true;
162             dvd.printDVD();
163         }
164     }
165     if (!found) {
166         System.out.println("Not found!");
167     }
168 }
169
170● public String toString() {
171     StringBuilder details = new StringBuilder("Cart Details:\n");
172
173     for (int i = 0; i < getQtyOrdered(); i++) {
174         DigitalVideoDisc disc = itemsOrdered[i];
175         if (disc != null) { // Kiểm tra null trước khi sử dụng
176             details.append("DVD ID: ").append(disc.getId())
177                 .append(", Title: ").append(disc.getTitle())
178                 .append(", Cost: ").append(disc.getCost()).append("\n");
179         }
180     }
181
182     return details.toString();
183 }
```

- Method in DigitalVideoDisc class:


```

44 // Method to print a dvd
45 public void printDVD() {
46     System.out.println(id + ". DVD - "
47         + title + " - "
48         + category + " - "
49         + director + " - "
50         + length + ": "
51         + cost + "$");
52 }
53
54 // Method to finds out if the corresponding disk is a match given the title.
55 public boolean isMatch(String title) {
56     return title.equals(this.title);
57 }

```

- CartTest:

```

1 package Lab03.AimsProject;
2
3 public class CartTest {
4
5     public static void main(String[] args) {
6         // Create a new cart
7         Cart cart = new Cart();
8
9         // Create new DVD objects and add them to the cart
10        DigitalVideoDisc dvd1 = new DigitalVideoDisc ("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
11        cart.addDigitalVideoDisc(dvd1);
12
13        DigitalVideoDisc dvd2 = new DigitalVideoDisc ("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
14        cart.addDigitalVideoDisc(dvd2);
15
16        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
17        cart.addDigitalVideoDisc(dvd3);
18
19        // Test the print method
20        cart.printCart();
21        // To-do: Test the search methods here
22        cart.searchByID(2);
23        cart.searchByID(3);
24        cart.searchByTitle("Star Wars");
25        cart.searchByTitle("Aladin");
26    }
27 }

```

→ Kết quả:

```

The disc has been added
The disc has been added
The disc has been 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 - Aladin - Animation - null - 0: 18.99$
Total cost: 63.89
*****
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$
3. DVD - Aladin - Animation - null - 0: 18.99$
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$
3. DVD - Aladin - Animation - null - 0: 18.99$

```

7. Implement the Store class:

- Store:

```
1 package Lab03.AimsProject;
2
3 public class Store {
4     public static final int MAX_ITEMS_IN_STORE = 100;
5     private DigitalVideoDisc[] itemsInStore = new DigitalVideoDisc[MAX_ITEMS_IN_STORE];
6     private int qtyInStore;
7
8     // Constructor
9     public Store() {
10         this.qtyInStore = 0;
11     }
12
13     // Add DVD to store
14     public void addDVD(DigitalVideoDisc disc) {
15         if (qtyInStore >= MAX_ITEMS_IN_STORE) {
16             System.out.println("The store is full, cannot add more DVDs.");
17             return;
18         }
19         itemsInStore[qtyInStore] = disc;
20         qtyInStore++;
21         System.out.println("The DVD has been added to the store: " + disc.getTitle());
22     }
23
24     // Remove DVD from store
25     public void removeDVD(DigitalVideoDisc disc) {
26         boolean found = false;
27         for (int i = 0; i < qtyInStore; i++) {
28             if (itemsInStore[i] != null && itemsInStore[i].equals(disc)) {
29                 found = true;
30                 // Move the following elements up
31                 for (int j = i; j < qtyInStore - 1; j++) {
32                     itemsInStore[j] = itemsInStore[j + 1];
33                 }
34                 itemsInStore[qtyInStore - 1] = null; // Delete last element
35                 qtyInStore--;
36                 System.out.println("The DVD has been removed from the store: " + disc.getTitle());
37                 break;
38             }
39         }
40         if (!found) {
41             System.out.println("The DVD was not found in the store.");
42         }
43     }
44
45     // Print a list of DVDs currently available in the store
46     public void printStore() {
47         System.out.println("*****STORE*****");
48         System.out.println("Items in Store:");
49         for (int i = 0; i < qtyInStore; i++) {
50             if (itemsInStore[i] != null) {
51                 itemsInStore[i].printDVD();
52             }
53         }
54         System.out.println("*****");
55     }
56 }
```

- StoreTest:

```

1 package Lab03.AimsProject;
2
3 public class StoreTest {
4     public static void main(String[] args) {
5         Store store = new Store();
6
7         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
8         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 124, 24.95f);
9         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladdin", "Animation", 18.99f);
10
11         store.addDVD(dvd1);
12         store.addDVD(dvd2);
13         store.addDVD(dvd3);
14
15         store.printStore();
16
17         store.removeDVD(dvd2);
18
19         store.printStore();
20
21         DigitalVideoDisc dvd4 = new DigitalVideoDisc("Frozen");
22         store.removeDVD(dvd4);
23     }
24 }

```

→ Kết quả:

```

The DVD has been added to the store: The Lion King
The DVD has been added to the store: Star Wars
The DVD has been added to the store: Aladdin
*****STORE*****
Items in Store:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
2. DVD - Star Wars - Science Fiction - George Lucas - 124: 24.95$
3. DVD - Aladdin - Animation - null - 0: 18.99$
*****
The DVD has been removed from the store: Star Wars
*****STORE*****
Items in Store:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
3. DVD - Aladdin - Animation - null - 0: 18.99$
*****
The DVD was not found in the store.

```

9. String, StringBuilder and StringBuffer:

```
1 package hust.soict.ite6.garbage;|
2 import java.util.*;
3
4 public class ConcatenationInLoops {
5
6     public static void main(String[] args) {
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 prints roughly 4500.
13        r = new Random (123);
14        start = System.currentTimeMillis();
15        StringBuilder sb = new StringBuilder();
16        for (int i = 0; i < 65536; i++)
17            sb.append(r.nextInt(2));
18        s = sb.toString();
19        System.out.println(System.currentTimeMillis() - start); // This prints 5.
20    }
21 }
```

```
1 package hust.soict.ite6.garbage;
2
3 import java.io.IOException;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 public class GarbageCreator {
8     public static void main(String[] args) throws IOException {
9         String filename = "D:\\2024.1\\00PLab-IT3103\\Lab_02\\answer.pdf";
10        byte[] inputBytes = { 0 };
11        long startTime, endTime;
12
13        inputBytes = Files.readAllBytes(Paths.get(filename));
14        startTime = System.currentTimeMillis();
15        String outputString = "";
16        for (byte b : inputBytes) {
17            outputString += (char) b;
18        }
19        endTime = System.currentTimeMillis();
20        System.out.println(endTime - startTime);
21    }
22 }
```