

# Object-Oriented Programming

## **Lab 03: Basic Object-Oriented Techniques**

Họ và Tên: Nguyễn Bá Hoàng

MSSV: 20225844

Mã Lớp: 744520

## 1. Branch your repository

## 2. Working with method overloading

### 2.1. Overloading by differing types of parameter

```
28  public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList){ 1 usage  hello24680
29      int k = 0; //bien index phan tu cua dvdList de cho vao itemsOrdered
30      for(int i =0; i < 20; i++) {
31          if(k == dvdList.length){ //neu k == length cua dvdList thi dung chay for
32              break;
33          }
34          if (qtyOrdered != 20) {
35              if(itemsOrdered[i] == null) {
36                  itemsOrdered[i] = dvdList[k++]; //thuc hien xong roi thi k + 1
37                  System.out.println("The disc " + k + " has been added");
38                  qtyOrdered += 1;
39              }
40          } else {
41              System.out.println("The cart is almost full 2.1");
42              break;
43          }
44      }
45  }
```

Kết quả:

Test:

```
//test overLoad pass array dvd to addDigitalVideoDisc(DigitalVideoDisc[] dvdList)
DigitalVideoDisc arr_dvd[] = {new DigitalVideoDisc( title: "arr_dvd0"),new DigitalVideoDisc( title: "arr_dvd1"),new DigitalVideoDisc( title: "arr_dvd2")};
anOrder.addDigitalVideoDisc(arr_dvd);
```

Result:

```
The disc 1 has been added
The disc 2 has been added
The disc 3 has been added
```

-Method pass an arbitrary number of arguments for dvd:

```
50
51 //pass arbitrary number of arguments (pass bao nhieu tham so cung duoc)
52 public void addDigitalVideoDisc(DigitalVideoDisc... dvdList_arbitrary){ 2 usages  24680
53     int k = 0; //bien index phan tu cua dvdList de cho vao itemsOrdered
54     for(int i=0; i < 20; i++) {
55         if(k == dvdList_arbitrary.length){ //neu k == length cua dvdList thi dung chay for
56             break;
57         }
58         if (qtyOrdered != 20) {
59             if(itemsOrdered[i] == null) {
60                 itemsOrdered[i] = dvdList_arbitrary[k++]; //thuc hien xong roi thi k + 1
61                 System.out.println("The disc " + k + " has been added");
62                 qtyOrdered += 1;
63             }
64         } else {
65             System.out.println("The cart is almost full 2.2");
66             break;
67         }
68     }
69 }
```

Kết quả:

Test:

```
46 //test overload pass arbitrary argument to addDigitalVideoDisc(DigitalVideoDisc... dvdList_arbitrary)
47 DigitalVideoDisc arbitrary_dvd0 = new DigitalVideoDisc( title: "arbitrary_dvd0");
48 DigitalVideoDisc arbitrary_dvd1 = new DigitalVideoDisc( title: "arbitrary_dvd1");
49 DigitalVideoDisc arbitrary_dvd2 = new DigitalVideoDisc( title: "arbitrary_dvd2");
50 anOrder.addDigitalVideoDisc(arbitrary_dvd0,arbitrary_dvd1,arbitrary_dvd2);
```

Result:

```
The disc 1 has been added
The disc 2 has been added
The disc 3 has been added
```

**Câu hỏi:** Try to add a method **addDigitalVideoDisc** which allows to pass an arbitrary number of arguments for dvd. Compare to an array parameter. **What do you prefer in this case?**

**Trả lời:** + toán tử “...” Dùng để truyền số lượng tham số không xác định vào một phương thức. Trong khi đó , toán tử “[ ]” Dùng để lưu trữ một tập hợp các giá trị cố định với kích thước xác định.

+ toán tử “...” phù hợp với truyền số lượng tham số không biết trước còn “[ ]” lại phù hợp với truyền số lượng phần tử mà ta biết trước từ đầu.

- ⇒ Theo em, em cảm thấy thích dùng toán tử mang “[]” hơn vì theo thói quen nhưng không thể phủ nhận rằng toán tử “...” vẫn tiện hơn và dùng được trong nhiều trường hợp hơn.

## 2.2. Overloading by differing the number of parameters

```
72
73     public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2){ 1 usage  hello24680
74         int k = 1; //danh dau dvd nao se dc add => k = 1 la dvd 1, k = 2 la dvd 2
75         for(int i =0; i < 20; i++) {
76             if (qtyOrdered != 20) {
77                 if(itemsOrdered[i] == null) {
78                     if(k == 1) {
79                         itemsOrdered[i] = dvd1;
80                         System.out.println("The disc 1 has been added");
81                         qtyOrdered += 1;
82                         k++;
83                     } else {
84                         itemsOrdered[i] = dvd2;
85                         System.out.println("The disc 2 has been added");
86                         qtyOrdered += 1;
87                         break;
88                     }
89                 }
90             } else {
91                 System.out.println("The cart is almost full 3");
92                 break;
93             }
94         }
95     }
```

**Kết quả:**

**Test:**

```
53     //test overload pass 2 dvd to addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2)
54     DigitalVideoDisc two_dvd0 = new DigitalVideoDisc( title: "two_dvd0");
55     DigitalVideoDisc two_dvd1 = new DigitalVideoDisc( title: "two_dvd1");
56     anOrder.addDigitalVideoDisc(two_dvd0,two_dvd1);
57
```

**Result:**

```
The disc 1 has been added
The disc 2 has been added
```

### 3. Passing parameter

#### Câu hỏi 1: Is JAVA a Pass by Value or a Pass by Reference programming language?

**Trả lời:** Java truyền tham trị và khi làm việc với đối tượng, tham trị này lại là tham trị của tham chiếu (con trỏ).

#### TestPassingParameter:

```
1 package hust.soict.hedspi.test.disc;
2
3 import hust.soict.hedspi.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter {
6     public static void main(String[] args) {
7         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
8         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
9
10        swap(jungleDVD, cinderellaDVD);
11        System.out.println("Jungle dvd title: " + jungleDVD.getTitle());
12        System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle());
13
14        changeTitle(jungleDVD, cinderellaDVD.getTitle());
15        System.out.println("Jungle dvd title: " + jungleDVD.getTitle());
16    }
17
18
19    public static void swap(Object d1, Object d2) {
20        Object temp = d1;
21        d1 = d2;
22        d2 = temp;
23    }
24
25
26
27    public static void changeTitle(DigitalVideoDisc dvd, String newTitle) {
28        String oldTitle = dvd.getTitle();
29        dvd.setTitle(newTitle);
30        dvd = new DigitalVideoDisc(oldTitle);
31    }
32}
```

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

**Trả lời:** Khi ta gọi swap(jungleDVD, cinderellaDVD), giá trị của các tham chiếu (jungleDVD và cinderellaDVD) được sao chép và truyền vào tham số d1 và d2 trong phương thức swap. Thay đổi các đối tượng trong phương thức chỉ có ảnh hưởng đến bản sao tham chiếu chứ không thay đổi đến đối tượng gốc

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

**Trả lời:** khi truyền một biến có kiểu object vào một hàm thì lúc đó ta chỉ truyền giá trị tham chiếu của biến đó chứ không phải đối tượng gốc được truyền vào. Vì vậy, khi gán object này cho object khác trong hàm thì object gốc sẽ không đổi. Chỉ khi sử dụng các method (Setter) của chính các object được tham chiếu này thì dữ liệu của object gốc mới được thay đổi.

**Swap() method mới thay đổi được object:**

```
34 @ - public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) { 1 usage hello24680
35     String tmp1 = dvd1.getTitle();
36     String tmp2 = dvd2.getTitle();
37     dvd1.setTitle(tmp2);
38     dvd2.setTitle(tmp1);
39 }
```

**Kết quả:**

```
↓
Jungle dvd title: Cinderella
Cinderella dvd title: Jungle
Jungle dvd title: Jungle
Process finished with exit code 0
```

## 4. Use debug run

### 4.1. Debugging Java in Eclipse

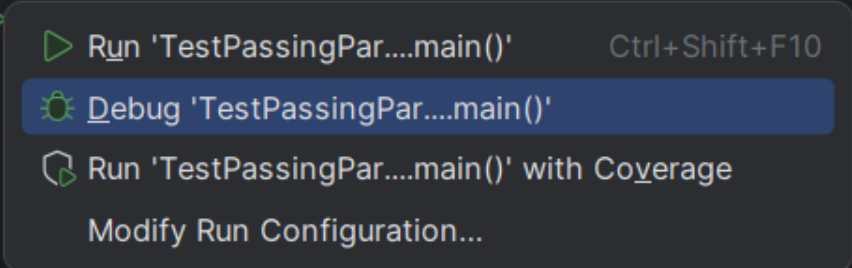
### 4.2. Example of debug run for the swap method of TestPassingParameter

#### 4.2.1. Setting, deleting & deactivate breakpoints:

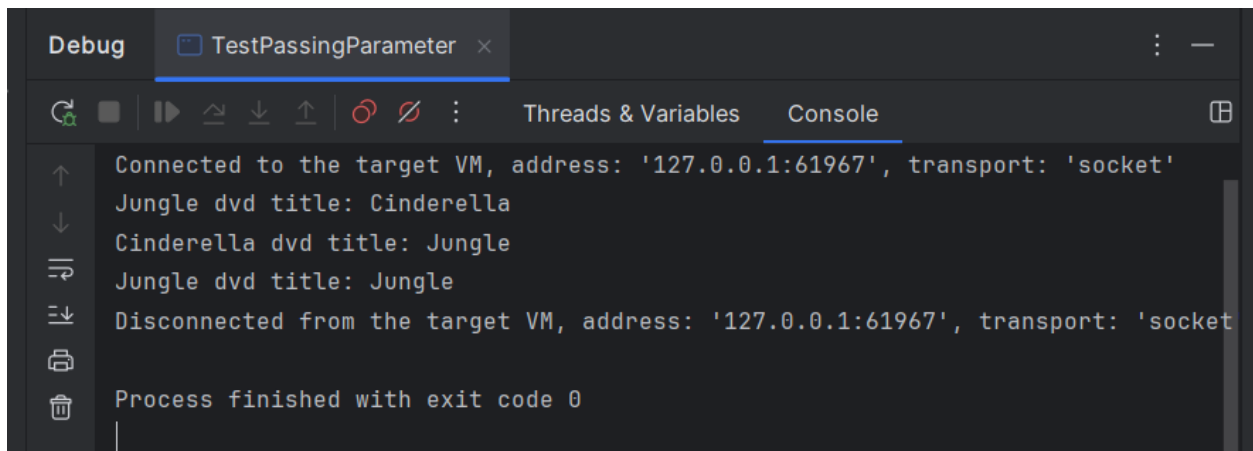
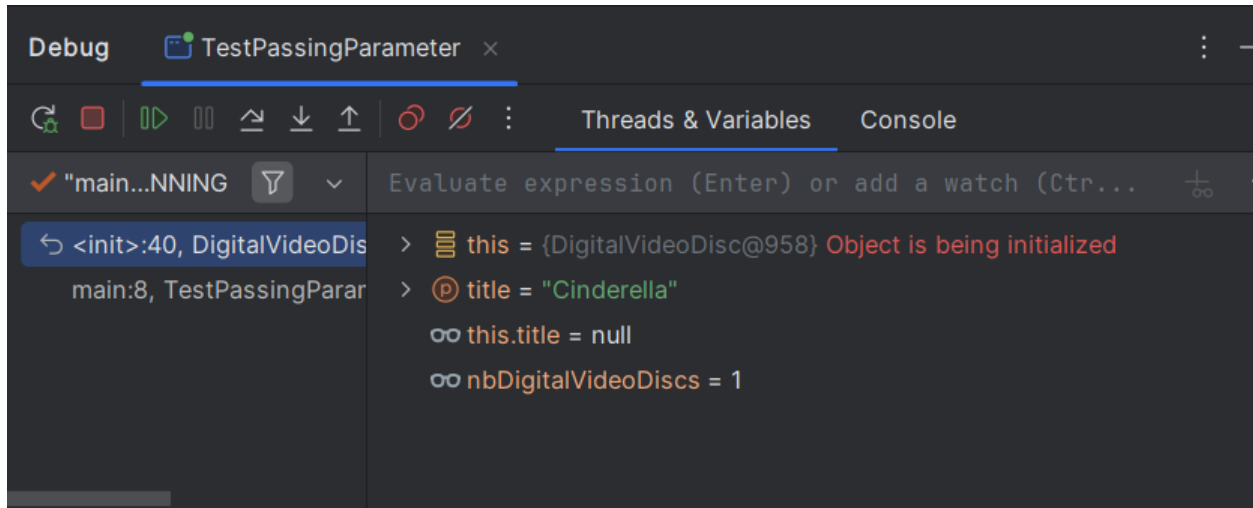
```
1 package hust.soict.hedspi.test.disc;
2
3 import hust.soict.hedspi.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter {
6     public static void main(String[] args) {
7         DigitalVideoDisc jungleDVD = new DigitalVideoDisc( title: "Jungle");
8         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc( title: "Cinderella");
9
10        swap(jungleDVD, cinderellaDVD);
11        System.out.println("Jungle dvd title: " + jungleDVD.getTitle());
12        System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle());
13
14        changeTitle(jungleDVD, cinderellaDVD.getTitle());
15        System.out.println("Jungle dvd title: " + jungleDVD.getTitle());
16    }
17
18    /*
19     public static void swap(Object d1, Object d2) {
20        Object temp = d1;
21        d1 = d2;
22        d2 = temp;
23    }
24
```

#### 4.2.2. Run in Debug mode:

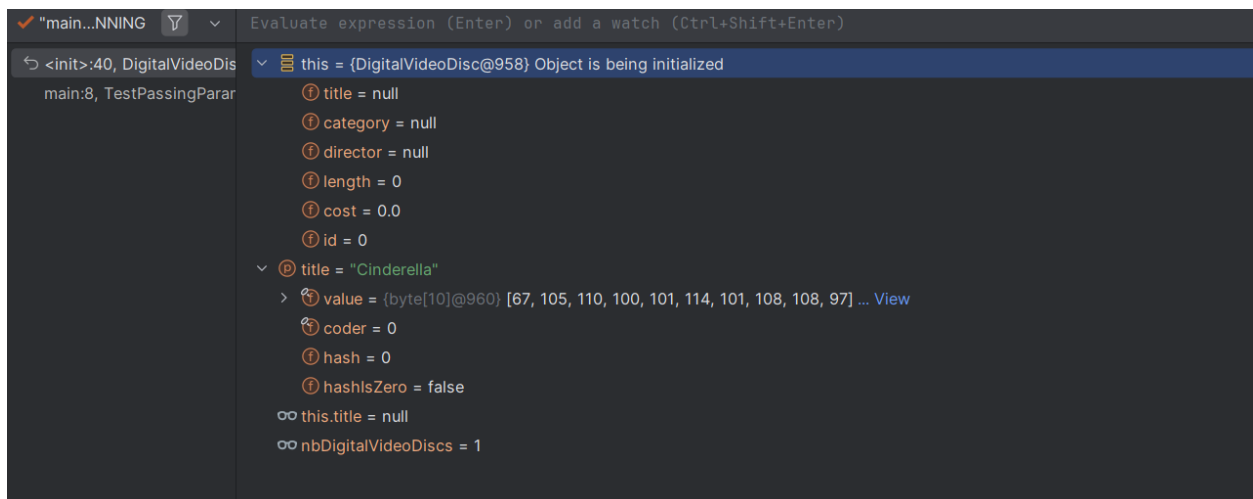
```
5 public class TestPassingParameter {
6
7
8
9
10
11        System.out.println("Jungle dvd title: " + jungleD
12        System.out.println("Cinderella dvd title: " + cin
13
14        changeTitle(jungleDVD, cinderellaDVD.getTitle());
15        System.out.println("Jungle dvd title: " + jungleD
```

A screenshot of the Eclipse IDE showing a code editor with Java code. A context menu is open over the code, displaying four options: 'Run \'TestPassingPar....main()\'' (with a green play icon and keyboard shortcut Ctrl+Shift+F10), 'Debug \'TestPassingPar....main()\'' (with a green bug icon and highlighted in blue), 'Run \'TestPassingPar....main()\' with Coverage' (with a green play icon and a coverage icon), and 'Modify Run Configuration...' (with a gear icon).

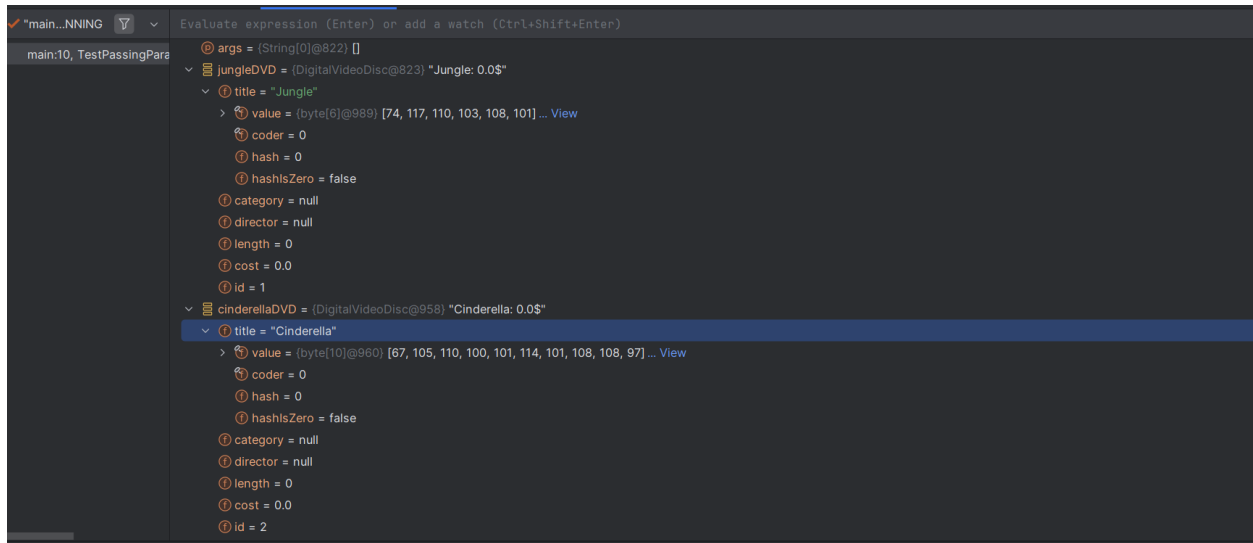
#### 4.2.3. Step Into, Step Over, Step Return, Resume:



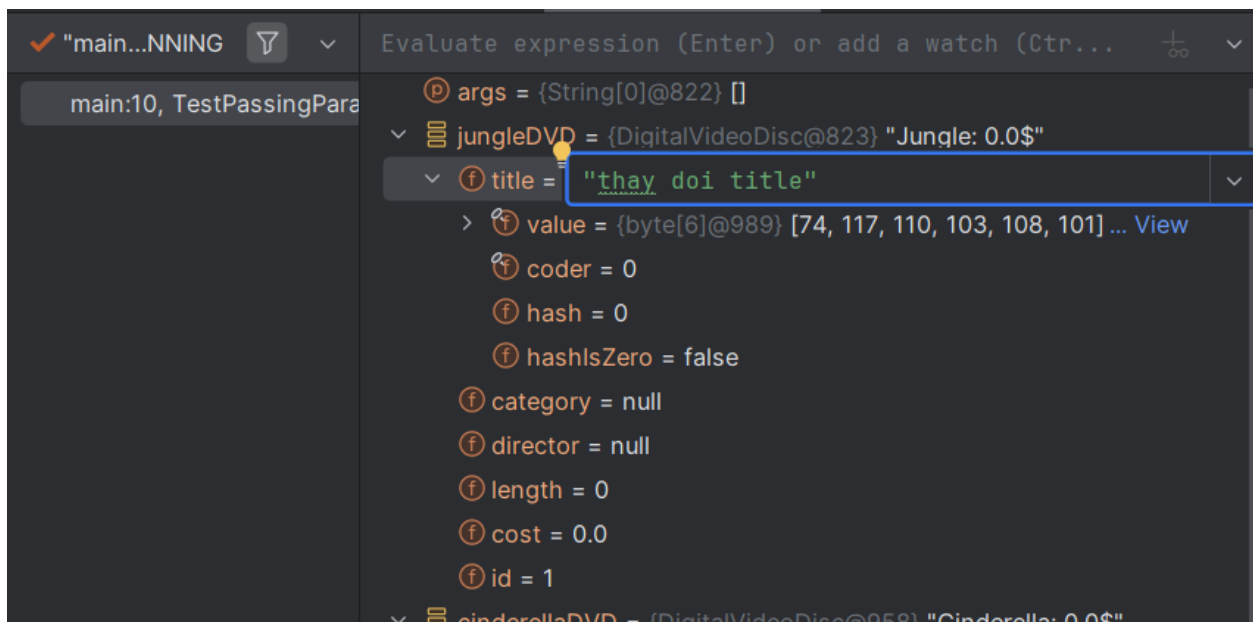
#### 4.2.4. Investigate value of variables:







#### 4.2.5. Change value of variables:



```
Connected to the target VM, address: '127.0.0.1:62052', transport: 'socket'
Jungle dvd title: Cinderella
Cinderella dvd title: thay doi title
Jungle dvd title: thay doi title
```

## 5. Classifier Member and Instance Member

```
public class DigitalVideoDisc { 56 usages  👤 hello24680
    private String title; 8 usages
    private String category; 6 usages
    private String director; 5 usages
    private int length = 0; 4 usages
    private float cost = 0; 5 usages
    private int id; 5 usages

    private static int nbDigitalVideoDiscs = 0; 8 usages
```

```
public DigitalVideoDisc(String title) { 10 usages  👤 hello24680
    this.title = title;

    nbDigitalVideoDiscs++;
    this.id = nbDigitalVideoDiscs;
}
```

(các constructor khác tương tự)

Test:

```
54 DigitalVideoDisc two_dvd0 = new DigitalVideoDisc( title: "two_dvd0");
55 DigitalVideoDisc two_dvd1 = new DigitalVideoDisc( title: "two_dvd1");
56 anOrder.addDigitalVideoDisc(two_dvd0,two_dvd1);
57
58 //test id
59 System.out.println(two_dvd0.getId());
60 System.out.println(two_dvd1.getId());
```

Kết quả:

```
📄 The disc 1 has been added
📄 The disc 2 has been added
🗑 8
9
```

## 6. Open the Cart class

### 6.1. printCart():

```
124     public void PrintCart(){ 2 usages  hello24680
125         System.out.println("*****CART*****");
126         System.out.println("Ordered Items:");
127
128         int stt = 1; // in ra man hinh dvd theo stt tang dan
129         for(int i = 0; i < 20; i++){
130             if(itemsOrdered[i] != null){
131                 System.out.println(stt + ".DVD" + " - " + itemsOrdered[i].toString());
132                 stt++;
133             }
134         }
135
136         System.out.println("Total cost: " + totalCost() + "$");
137         System.out.println("*****");
138     }
139 }
```

### Kết quả chạy trong CartTest:

```
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: 18.99$
Total cost: 63.89$
*****
```

### 6.2. searchCart()

- Theo id:

```

140     public void search(int id){ 2 usages  hello24680
141         int check = 0; //neu check khac 0 thi tim ra duoc id
142         for (int i = 0; i < 20; i++){
143             if(itemsOrdered[i] != null) {
144                 if (itemsOrdered[i].getId() == id) {
145                     System.out.println("tim thay id:" + id + " - " + itemsOrdered[i].toString());
146                     check = 1; //tim ra duoc id thi danh dau check = 1
147                     break;
148                 }
149             }
150         }
151
152         if(check == 0){
153             System.out.println("no match is found by id");
154         }
155     }

```

-theo title:

```

158     public void search(String title){ 3 usages  hello24680
159         int check = 0;
160         for (int i = 0; i < 20; i++){
161             if(itemsOrdered[i] != null) {
162                 if (itemsOrdered[i].isMatch(title) == true) {
163                     System.out.println("tim thay title:" + itemsOrdered[i].getId() + " - " + itemsOrdered[i].toString());
164                     check = 1; //tim ra duoc it nhat 1 title thoa man thi danh dau check = 1
165                 }
166             }
167         }
168
169         if(check == 0){
170             System.out.println("no match is found by title");
171         }
172     }
173
174 }
175

```

### -test:

```
27 //Test the search methods
28 //Test the search by id
29 //tim thay id
30 cart.search(id: 2);
31 //khong tim thay id
32 cart.search(id: 5);
33 //Test the search by title
34 //tim thay title
35 cart.search(title: "The Lion King");
36 //khong tim thay title
37 cart.search(title: "no found");
38 //tim thay 2 dvd cung title
39 //add 1 dvd cung ten "Aladin"
40 DigitalVideoDisc dvd_test_two_title = new DigitalVideoDisc(title: "Aladin", category: "MU", cost: 1.5f);
41 cart.addDigitalVideoDisc(dvd_test_two_title);
42 //test search
43 cart.search(title: "Aladin");
44 }
45 }
46
```

### -kết quả:

```
tim thay id:2 - Star Wars - Science Fiction - George Lucas - 87: 24.95$
no match is found by id
tim thay title:1 - The Lion King - Animation - Roger Allers - 87: 19.95$
no match is found by title
The disc has been added
tim thay title:3 - Aladin - Animation: 18.99$
tim thay title:4 - Aladin - MU: 1.5$

Process finished with exit code 0
```

## 7. Implement the Store class

```
public class Store { 3 usages  👤 hello24680 *  
    private DigitalVideoDisc[] itemsInStore = new DigitalVideoDisc[100]; 8 usages  
    int qtyInStore = 0; 3 usages
```

### 7.1. addDVD():

```
    public void addDVD(DigitalVideoDisc disc) { 3 usages  👤 hello24680  
        for (int i = 0; i < 100; i++) {  
            if (qtyInStore != 100) {  
                if (itemsInStore[i] == null) {  
                    itemsInStore[i] = disc;  
                    System.out.println("The disc has been added");  
                    qtyInStore += 1;  
                    break;  
                }  
            } else {  
                System.out.println("The disc can't add to Store");  
                break;  
            }  
        }  
    }  
}
```

### 7.2. removeDVD():

```

27     public void removeDVD(DigitalVideoDisc disc) { 2 usages  🡕 hello24680
28         for (int i = 0; i < 100; i++) {
29             if (itemsInStore[i] != null) {
30                 if (itemsInStore[i].getTitle().equals(disc.getTitle())) {
31                     itemsInStore[i] = null;
32                     System.out.println("xoa thanh cong");
33                     qtyInStore -- 1;
34                     break;
35                 }
36             }
37             if (i == 99) {
38                 System.out.println("xoa khong thanh cong");
39             }
40         }
41     }

```

## Test:

```

> public class StoreTest {  🡕 hello24680
> ~ public static void main(String[] args) {  🡕 hello24680
    Store storeTest = new Store();

    //test add vao store
    DigitalVideoDisc dvd1 = new DigitalVideoDisc( title: "The Lion King", category: "Animation", director: "Roger Allens", length: 87, cost: 19.95f);
    storeTest.addDVD(dvd1);

    DigitalVideoDisc dvd2 = new DigitalVideoDisc( title: "Star Wars", category: "Science Fiction", director: "George Lucas", length: 87, cost: 24.95f);
    storeTest.addDVD(dvd2);

    DigitalVideoDisc dvd3 = new DigitalVideoDisc( title: "Aladin", category: "Animation", director: "Director", length: 89, cost: 18.99f);
    storeTest.addDVD(dvd3);

    storeTest.printDVD(); //in ra ket qua storeTest add

    //test remove
    storeTest.removeDVD(dvd2);
    storeTest.printDVD(); //in ra ket qua storeTest da duoc remove

    //test khong tim thay dvd remove
    DigitalVideoDisc dvd4 = new DigitalVideoDisc( title: "not found");
    storeTest.removeDVD(dvd4);

    }
}

```

## Kết quả:

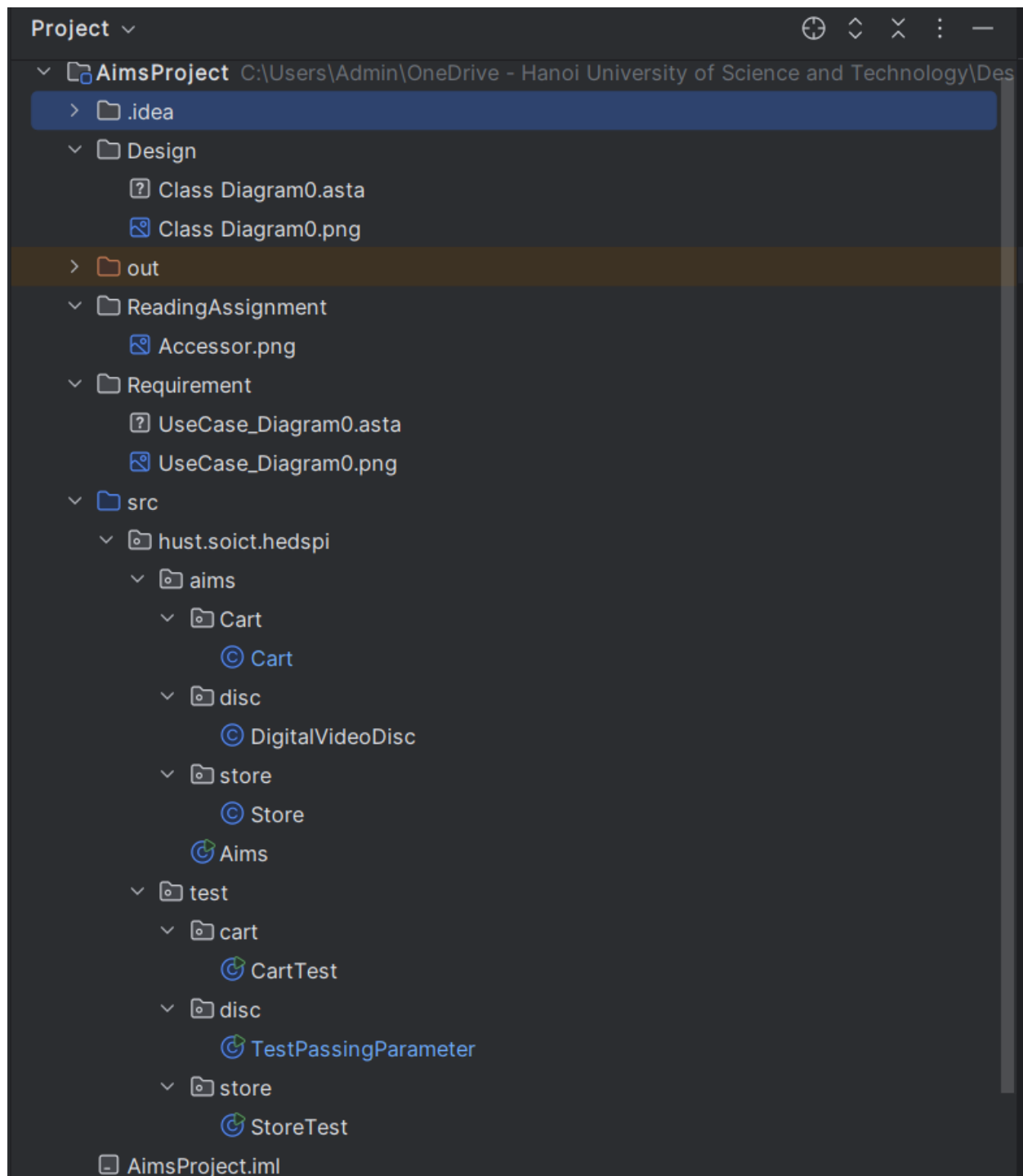
```
C:\Users\Admin\jdk-23\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=62710:C:\Program Files\
The disc has been added
The disc has been added
The disc has been added
id: 1;title: The Lion King
id: 2;title: Star Wars
id: 3;title: Aladin
xoa thanh cong
id: 1;title: The Lion King
id: 3;title: Aladin
xoa khong thanh cong

Process finished with exit code 0
```

## 8. Re-organize your projects

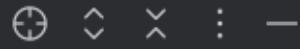
### 8.1. AimsProject:





## 8.2. OtherProjects:

Project ▾



▾ OtherProjects C:\Users\Admin\OneDrive - Hanoi Uni

> .idea

> Lab1

> Lab02

> out

▾ src

▾ hust

▾ soict.hedspi.garbage

ConcatenationInLoops

GarbageCreator

NoGarbage

test.txt

OtherProjects.iml

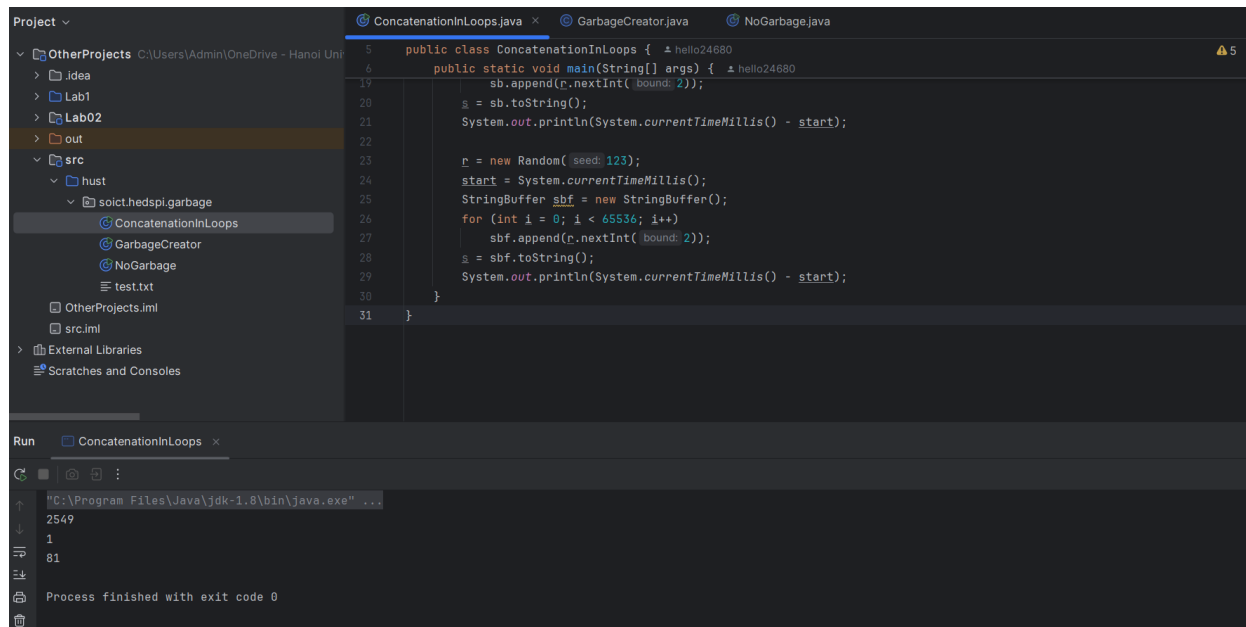
src.iml

> External Libraries

Scratches and Consoles

## 9. String, StringBuilder and StringBuffer

### 9.1. So sánh String, StringBuilder and StringBuffer



The screenshot shows an IDE with a project named 'OtherProjects' containing a sub-project 'src' with files 'ConcatenationInLoops', 'GarbageCreator', and 'NoGarbage'. The 'ConcatenationInLoops.java' file is open, showing a public class with a main method. The code generates a random number, calculates the time taken to concatenate it 65536 times using String, and prints the result. The output window shows the execution of 'C:\Program Files\Java\jdk-1.8\bin\java.exe' with the output '2549', '1', and '81', and 'Process finished with exit code 0'.

```
5 public class ConcatenationInLoops {
6     public static void main(String[] args) {
19         sb.append(r.nextInt( bound: 2));
20         s = sb.toString();
21         System.out.println(System.currentTimeMillis() - start);
22
23         r = new Random( seed: 123);
24         start = System.currentTimeMillis();
25         StringBuffer sbf = new StringBuffer();
26         for (int i = 0; i < 65536; i++)
27             sbf.append(r.nextInt( bound: 2));
28         s = sbf.toString();
29         System.out.println(System.currentTimeMillis() - start);
30     }
31 }
```

Run ConcatenationInLoops

"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...

2549

1

81

Process finished with exit code 0

#### Nhận xét:

- **String** kết hợp "+" sẽ tạo đối tượng mới mỗi lần nối chuỗi, do đó hiệu suất kém hơn 2 cái còn lại.
- **StringBuilder** nhanh nhất vì không tạo ra đối tượng mới mỗi lần nối chuỗi còn **StringBuffer** thì hiệu suất thấp hơn nhưng thích hợp cho một số tình huống cụ thể như xử lý đa luồng.

## 9.2. GarbageCreator và NoGarbage

### 9.2.1. file test.txt để kiểm tra hiệu suất:

```
1  ldbna
2  nucaow
3  ioqwue
4  cnneuf
5  auhdf
6  bdaiuw
7  naiuwe
8  sas
9  btnasd
10 ajwrfa
11 fnuefs
12 nsad
13 dnmiawn
14 nsdna
15 uiwhda
16 nadn
17 virhf
18 fna
19 udbna
20 nucaow
```

### 9.2.2. GarbageCreator

```

public class GarbageCreator {  ± hello24680
    public static void main(String[] args) throws IOException {  ± hello24680
        String filename = "src/hust/soict/hedspi/garbage/test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        inputBytes = Files.readAllBytes(Paths.get(filename));
        startTime = System.currentTimeMillis();

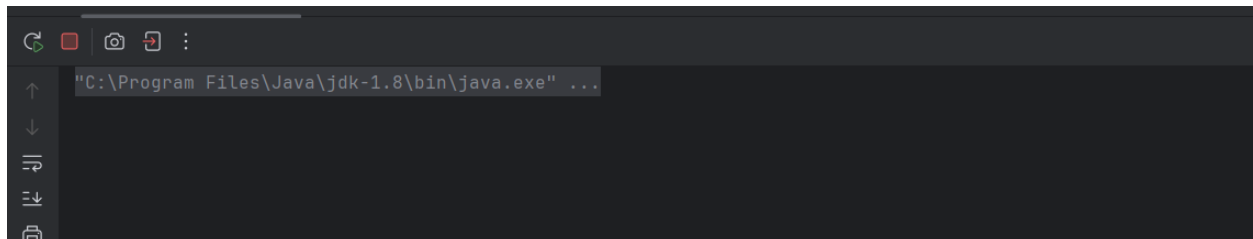
        //su dung string thong thuong
        String outputString = "";
        for (byte b : inputBytes) {
            outputString += (char) b;
        }

        /*
        StringBuilder outputStringBuilder = new StringBuilder();
        for (byte b : inputBytes) {
            outputStringBuilder.append((char) b);
        }
        */

        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}

```

**Kết quả: lỗi đơ màn hình không chạy được**



### 9.2.3. NoGarbage

```

7 public class NoGarbage {  hello24680
8     public static void main(String[] args) throws IOException {  hello24680
9         String filename = "src/hust/soict/hedspi/garbage/test.txt";
10        byte[] inputBytes = { 0 };
11        long startTime, endTime;
12
13        inputBytes = Files.readAllBytes(Paths.get(filename));
14        startTime = System.currentTimeMillis();
15
16        //su dung string builder
17        StringBuilder outputStringBuilder = new StringBuilder();
18        for (byte b : inputBytes) {
19            outputStringBuilder.append((char) b);
20        }
21
22        endTime = System.currentTimeMillis();
23        System.out.println(endTime - startTime);
24    }
25 }

```

**Kết quả:**

```

↑ "C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
↓ 12
|||
-| Process finished with exit code 0
=|

```

**Nx:** String sẽ tạo đối tượng mới do đó số lượng đối tượng String tăng lên nhanh chóng nên sẽ gây mất nhiều thời gian. Còn StringBuilder chỉ thay đổi nội dung của nó không có sự sao chép chuỗi nên tốc độ xử lý nhanh hơn.

## 10. Release flow demonstration

Conversation 0
Commits 1
Checks 0
Files changed 3
+6 -4

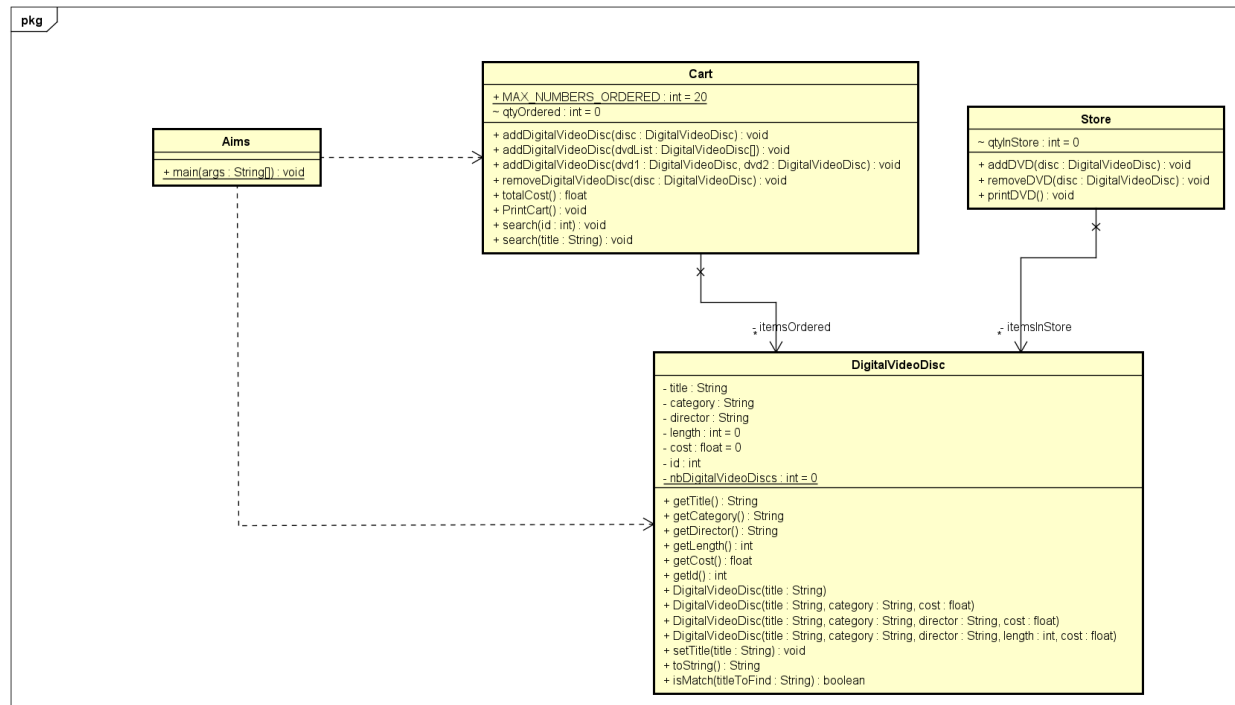
Commits on Nov 26, 2024

add a feature/demonstrate-release-flow

hello24680 committed 5 minutes ago

2bf9177

## 11. Class diagram



## 12. Use case diagram

## AIMS

