

# BÁO CÁO THỰC HÀNH LAB 03

## THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Contents

1. Working with method overloading.....	2
1.1. Overloading by differing types of parameter .....	2
Code .....	2
Result.....	2
1.2. Overloading by differing the number of parameters .....	3
Code .....	3
Result.....	5
2. Passing parameter .....	6
Code .....	6
Result.....	6
3. Class Member and Instance Member.....	7
Code .....	7
Result.....	9
4. Open the Cart class.....	9
Code .....	9
Result.....	11
5. Implement the Store class.....	12
Code .....	12
Result.....	13
6. String, StringBuilder and StringBuffer.....	14
Code .....	14
Result.....	14
7 Class Diagram .....	16
Table of Figures .....	17

## 1. Working with method overloading

### 1.1. Overloading by differing types of parameter

Code

```
public void addDigitalVideoDisc_hacd(DigitalVideoDisc [] dvdlist){
    for(int i = 0; i < dvdlist.length; i++){
        if(qtyOrdered >= MAX_NUMBER_ORDERED){
            System.out.println(x:"The cart is almost full");
        }else{
            itemsOrdered[qtyOrdered] = dvdlist[i];
            qtyOrdered ++;
            System.out.println(x:"The disc has been added");
        }
    }
}
```

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

```
5 public class Aims {
6     Run | Debug
7     public static void main(String[] args){
8         //Create a new cart
9         Cart anOrder = new Cart();
10
11         //Create new dvd objects and add them to the cart
12         DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",director:"Roger Allers", length:87, cost:19.95f);
13         anOrder.addDigitalVideoDisc_hacd(dvd1);
14
15         DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
16         anOrder.addDigitalVideoDisc_hacd(dvd2);
17
18         DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin", category:"Animation", cost:18.99f);
19         anOrder.addDigitalVideoDisc_hacd(dvd3);
20
21         //print total cost of the items in the cart
22         System.out.println("Total cost is: " + anOrder.totalCost_hacd());
23
24         //remove a dvd from the cart and print the new total cost
25         anOrder.removeDigitalVideoDisc_hacd(dvd2);
26         System.out.println("Total cost is: " + anOrder.totalCost_hacd());
27
28         DigitalVideoDisc dvd4 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",director:"Roger Allers", length:87, cost:19.95f);
29
30         DigitalVideoDisc dvd5 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
31
32         DigitalVideoDisc dvd6 = new DigitalVideoDisc(title:"Aladin", category:"Animation", cost:18.99f);
33
34         DigitalVideoDisc[] dvdlist= {dvd4, dvd5, dvd6};
35         anOrder.addDigitalVideoDisc_hacd(dvdlist);
36     }
37 }
```

Figure 2 Aims Class

Result

```
The disc has been added
The disc has been added
The disc has been added
Total cost is: 63.89
The disc has been removed
Total cost is: 38.940002
The disc has been added
The disc has been added
The disc has been added
```

Figure 3 Result

## 1.2. Overloading by differing the number of parameters

### Code

```

31     public void addDigitalVideoDisc_hacd(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2){
32         if(qtyOrdered >= MAX_NUMBER_ORDERED - 1){
33             System.out.println(x:"The cart is almost full");
34         }else{
35             itemsOrdered[qtyOrdered] = dvd1;
36             qtyOrdered++;
37             itemsOrdered[qtyOrdered] = dvd2;
38             qtyOrdered++;
39             System.out.println(x:"The discs have| been added");
40         }
41     }
42

```

Figure 4 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

```

//remove a dvd from the cart and print the new total cost
anOrder.removeDigitalVideoDisc_hacd(dvd2);
System.out.println("Total cost is: " + anOrder.totalCost_hacd());

DigitalVideoDisc dvd4 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",director:"Roger Allers", length:87, cost:19.95f);
DigitalVideoDisc dvd5 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f)
DigitalVideoDisc dvd6 = new DigitalVideoDisc(title:"Aladin", category:"Animation", cost:18.99f);

DigitalVideoDisc[] dvdlist= {dvd4, dvd5, dvd6};
anOrder.addDigitalVideoDisc_hacd(dvdlist);

System.out.println(x:"");
DigitalVideoDisc dvd7 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",director:"Roger Allers", length:87, cost:19.95f);
DigitalVideoDisc dvd8 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f)

anOrder.addDigitalVideoDisc_hacd(dvd7, dvd8);
}

```

Figure 5 Aims Class

## Result

```
The disc has been added
The disc has been added
The disc has been added
Total cost is: 63.89
The disc has been removed
Total cost is: 38.940002
The disc has been added
The disc has been added
The disc has been added

The discs have been added
```

*Figure 6 Result*

## 2. Passing parameter

### Code

```

1  package hust.soict.dsai.test.disc;
2  import hust.soict.dsai.disc.DigitalVideoDisc;
3
4  public class TestPassingParameter {
5      //Run | Debug
6      public static void main(String[] args) {
7          //To-do Auto-generated method stub
8          DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");
9          DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"Cinderella");
10
11          swap_hacd(jungleDVD, cinderellaDVD);
12          System.out.println("jungle dvd title: " + jungleDVD.getTitle_hacd());
13          System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle_hacd());
14
15          newswap_hacd(jungleDVD, cinderellaDVD);
16          System.out.println("jungle dvd title: " + jungleDVD.getTitle_hacd());
17          System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle_hacd());
18
19          changeTitle_hacd(jungleDVD, cinderellaDVD.getTitle_hacd());
20          System.out.println("jungle dvd title: " + jungleDVD.getTitle_hacd());
21      }
22
23      public static void swap_hacd(Object o1, Object o2){
24          Object tmp = o1;
25          o1 = o2;
26          o2 = tmp;
27      }
28
29      public static void newswap_hacd(DigitalVideoDisc disc1, DigitalVideoDisc disc2){
30          DigitalVideoDisc tmp = new DigitalVideoDisc(disc1.getTitle_hacd(), disc1.getCategory_hacd(), disc1.getDirector_hacd(), disc1.getLength_hacd(), disc1.getCost_hacd());
31
32          disc1.setTitle_hacd(disc2.getTitle_hacd());
33          disc1.setCategory_hacd(disc2.getCategory_hacd());
34          disc1.setDirector_hacd(disc2.getDirector_hacd());
35          disc1.setLength_hacd(disc2.getLength_hacd());
36          disc1.setCost_hacd(disc2.getCost_hacd());
37
38          disc2.setTitle_hacd(tmp.getTitle_hacd());
39          disc2.setCategory_hacd(tmp.getCategory_hacd());
40          disc2.setDirector_hacd(tmp.getDirector_hacd());
41          disc2.setLength_hacd(tmp.getLength_hacd());
42          disc2.setCost_hacd(tmp.getCost_hacd());
43      }
44
45      public static void changeTitle_hacd(DigitalVideoDisc dvd, String title){
46          String oldTitle = dvd.getTitle_hacd();
47          dvd.setTitle_hacd(title);
48          dvd = new DigitalVideoDisc(oldTitle);
49      }
50  }
51

```

Figure 7 Passing parameter code

### Result

```

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

```

Figure 8 Passing parameter result

### 3. Class Member and Instance Member

#### Code

```
1  package hust.soict.dsai.disc;
2  public class DigitalVideoDisc {
3      private String title;
4      private String category;
5      private String director;
6      private int length;
7      private float cost;
8      private int id;
9      private static int nbDigitalVideoDisc = 0;
10
11     public String getTitle_hacd(){
12         return title;
13     }
14
15     public String getCategory_hacd(){
16         return category;
17     }
18
19     public String getDirector_hacd(){
20         return director;
21     }
22
23     public int getLength_hacd(){
24         return length;
25     }
26
27     public float getCost_hacd(){
28         return cost;
29     }
30
31     public int getId_hacd(){
32         return id;
33     }
```

```
34
35     public void setTitle_hacd(String title){
36         this.title = title;
37     }
38
39     public void setCategory_hacd(String category){
40         this.category = category;
41     }
42
43     public void setDirector_hacd(String director){
44         this.director = director;
45     }
46
47     public void setLength_hacd(int length){
48         this.length = length;
49     }
50
51     public void setCost_hacd(float cost){
52         this.cost = cost;
53     }
54
```

```
55
56     public DigitalVideoDisc(String title){
57         super();
58         nbDigitalVideoDisc++;
59         this.id = nbDigitalVideoDisc;
60         this.title = title;
61     }
62
63
64     public DigitalVideoDisc(String title,String category, float cost){
65         super();
66         nbDigitalVideoDisc++;
67         this.id = nbDigitalVideoDisc;
68         this.title = title;
69         this.category = category;
70         this.cost = cost;
71     }
72
73     public DigitalVideoDisc(String title, String category, String director, float cost){
74         super();
75         nbDigitalVideoDisc++;
76         this.id = nbDigitalVideoDisc;
77         this.title = title;
78         this.category = category;
79         this.director= director;
80         this.cost = cost;
81     }
82
```



```

public DigitalVideoDisc(String title, String category, String director, int length, float cost){
    super();
    nbDigitalVideoDisc++;
    this.id = nbDigitalVideoDisc;
    this.title = title;
    this.category = category;
    this.director = director;
    this.length = length;
    this.cost = cost;
}

public boolean isMatched_hacd(String searchString){
    if(this.getTitle_hacd().equals(searchString)) return true;
    return false;
}

public boolean isMatched_hacd(int id){
    if(this.getId_hacd() == id) return true;
    return false;
}

public String printTitle_hacd(){
    String str = " - " + this.title;
    return str;
}

public String printCategory_hacd(){
    String str;
    if(this.category != null)
        str = " - " + this.category;
    else str = "";
    return str;
}

public String printDirector_hacd(){
    String str;
    if(this.director != null)
        str = " - " + this.director;
    else str = "";
    return str;
}

public String printLength_hacd(){
    String str;
    if(this.length != 0)
        str = " - " + String.valueOf(this.length);
    else str = "";
    return str;
}

public String printCost_hacd(){
    String str;
    if(this.cost != 0)
        str = " : " + String.valueOf(this.cost) + "$";
    else str = "";
    return str;
}

public String toString_hacd(){
    String display = this.printTitle_hacd() + this.printCategory_hacd() + this.printDirector_hacd() + this.printLength_hacd() + this.printCost_hacd();
    return display;
}
}

```

Figure 9 Class member and Instance member

## Result

```
//Create a new cart
Cart cart = new Cart();

//Create a new dvd objects and add them to the cart
DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",director:"Roger Allers", length:87, cost:19.95f);
cart.addDigitalVideoDisc_hacd(dvd1);

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

DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin", category:"Animation", cost:18.99f);
cart.addDigitalVideoDisc_hacd(dvd3);

cart.print_hacd();
```

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

Figure 10 Result

## 4. Open the Cart class

### Code

```
public void print_hacd(){
    for(int i = 0; i < 20; i++)
        System.out.print(s:"*");
    System.out.print(s:"Cart");
    for(int i = 0; i < 20; i++)
        System.out.print(s:"*");
    System.out.println(x:"");
    System.out.println(x:"Ordered Items:");
    for(int i = 0; i < qtyOrdered; i++){
        System.out.println(i+1 +". DVD" + itemsOrdered[i].toString_hacd());
    }
    System.out.println("Total cost: " + totalCost_hacd());
    //System.out.println("");
    for(int i = 0; i < 40; i++)
        System.out.print(s:"*");
}
```

Figure 11 Code method to print the content in Cart

```
public void search_hacd(int id){
    int k = -1;
    for(int i = 0; i < qtyOrdered; i++){
        boolean t = itemsOrdered[i].isMatched_hacd(id);
        if(t){
            k = i;
        }
    }
    if(k == -1){
        System.out.println(x:"The item has not been found.");
    }else{
        System.out.println(x:"The item has been found: ");
        System.out.println(k+1 + ". DVD " + itemsOrdered[k].toString_hacd());
    }
}
```

Figure 12 Code method search by id

```
public void search_hacd(String str){
    int k = -1;
    for(int i = 0; i < qtyOrdered; i++){
        boolean t = itemsOrdered[i].isMatched_hacd(str);
        if(t){
            k = i;
        }
    }
    if(k == -1){
        System.out.println(x:"The item has not been found.");
    }else{
        System.out.println(x:"The item has been found: ");
        System.out.println(k+1 + ". DVD " + itemsOrdered[k].toString_hacd());
    }
}
```

Figure 13 Code method search by title

```
public boolean isMatched_hacd(String searchString){  
    if(this.getTitle_hacd().equals(searchString)) return true;  
    return false;  
}  
  
public boolean isMatched_hacd(int id){  
    if(this.getId_hacd() == id) return true;  
    return false;  
}
```

Figure 14 IsMatch() in DigitalVideoDisc class

```
1 package hust.soict.dsai.test.cart;
2 import hust.soict.dsai.aims.cart.Cart;
3 import hust.soict.dsai.disc.DigitalVideoDisc;
4
5 public class CartTest {
6     Run | Debug
7     public static void main(String[] args) {
8         //Create a new cart
9         Cart cart = new Cart();
10
11         //Create a new dvd objects and add them to the cart
12         DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",director:"Roger Allers", length:87, cost:19.95f);
13         cart.addDigitalVideoDisc_hacd(dvd1);
14
15         DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
16         cart.addDigitalVideoDisc_hacd(dvd2);
17
18         DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin", category:"Animation", cost:18.99f);
19         cart.addDigitalVideoDisc_hacd(dvd3);
20
21         cart.print_hacd();
22         System.out.println(x:"");
23         String searchString = "The Lion King";
24         cart.search_hacd(searchString);
25         int id = 2;
26         cart.search_hacd(id);
27         //Test the method
28     }
29 }
30
```

Figure 15 CartTest class

## Result

```
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: 18.99$
Total cost: 63.89
*****
The item has been found:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
The item has been found:
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$
```

Figure 16 Result

## 5. Implement the Store class

### Code

```
1 package hust.soict.dsai.aims.store;
2 import hust.soict.dsai.disc.DigitalVideoDisc;
3 public class Store {
4     public static final int MAX_NUMBER_INSTORE = 1000;
5     private DigitalVideoDisc itemsInStore[] = new DigitalVideoDisc[MAX_NUMBER_INSTORE];
6
7     public int qtyInStore;
8
9     public void addDVD_hacd(DigitalVideoDisc disc){
10         if(qtyInStore >= MAX_NUMBER_INSTORE){
11             System.out.println(x:"The store is almost full");
12         }else{
13             itemsInStore[qtyInStore] = disc;
14             qtyInStore++;
15             System.out.println(x:"The disc has been added");
16         }
17     }
18
19     public void removedVD_hacd(DigitalVideoDisc disc){
20         int findDisc = -1;
21         if(qtyInStore > 0){
22             for(int i = 0; i < qtyInStore; i++){
23                 if(itemsInStore[i].getTitle_hacd().equals((disc.getTitle_hacd()))){
24                     findDisc = i;
25                 }
26             }
27             if(findDisc != -1){
28                 for(int i = findDisc; i < qtyInStore - 1; i++){
29                     itemsInStore[i] = itemsInStore[i + 1];
30                 }
31                 itemsInStore[qtyInStore - 1] = null;
32                 qtyInStore--;
33                 System.out.println(x:"The disc has been removed");
34             }else{
35                 System.out.println(x:"The disc has not been found");
36             }
37         }
38     }
39 }
```

```

40     public void print_hacd(){
41         for(int i = 0; i < 20; i++){
42             System.out.print(s:"*");
43             System.out.print(s:"Store");
44             for(int i = 0; i < 20; i++){
45                 System.out.print(s:"*");
46             }
47             System.out.println(x:"");
48             System.out.println(x:"Available Items:");
49             for(int i = 0; i < qtyInStore; i++){
50                 System.out.println(i+1 + ". DVD " + itemsInStore[i].toString_hacd());
51             }
52             //System.out.println("");
53             for(int i = 0; i < 40; i++){
54                 System.out.print(s:"*");
55             }
56         }
57
58         public void search_hacd(String str){
59             int k = -1;
60             for(int i = 0; i < qtyInStore; i++){
61                 boolean t = itemsInStore[i].isMatched_hacd(str);
62                 if(t){
63                     k = i;
64                 }
65             }
66             if(k == -1){
67                 System.out.println(x:"The item has not been found.");
68             }else{
69                 System.out.println(x:"The item has been found: ");
70                 System.out.println(k+1 + ". DVD " + itemsInStore[k].toString_hacd());
71             }
72         }
73     }

```

```

73     public void search_hacd(int id){
74         int k = -1;
75         for(int i = 0; i < qtyInStore; i++){
76             boolean t = itemsInStore[i].isMatched_hacd(id);
77             if(t){
78                 k = i;
79             }
80         }
81         if(k == -1){
82             System.out.println(x:"The item has not been found.");
83         }else{
84             System.out.println(x:"The item has been found: ");
85             System.out.println(k+1 + ". DVD " + itemsInStore[k].toString_hacd());
86         }
87     }
88 }
89
90
91

```

Figure 17 Store Class



```

1 package hust.soict.dsai.test.store;
2 import hust.soict.dsai.aims.store.Store;
3 import hust.soict.dsai.disc.DigitalVideoDisc;
4
5 public class StoreTest {
6     public static void main(String[] args) {
7         //Create a new store
8         Store store = new Store();
9
10        //Create a new dvd objects and add them to the cart
11        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation",director:"Roger Allers", length:87, cost:19.95f);
12        store.addDVD_hacd(dvd1);
13
14        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, cost:24.95f);
15        store.addDVD_hacd(dvd2);
16
17        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin", category:"Animation", cost:18.99f);
18        store.addDVD_hacd(dvd3);
19
20        //Test the print method
21        store.print_hacd();
22        System.out.println(x:"");
23        store.removeDVD_hacd(dvd3);
24        store.print_hacd();
25        System.out.println(x:"");
26        String searchString = "The Lion King";
27        store.search_hacd(searchString);
28        int id = 2;
29        store.search_hacd(id);
30    }
31 }
32

```

Figure 18 StoreTest class

## Result

```

The disc has been added
The disc has been added
The disc has been added
*****Store*****
Available 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$
*****
The disc has been removed
*****Store*****
Available Items:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$
*****
The item has been found:
1. DVD - The Lion King - Animation - Roger Allers - 87: 19.95$
The item has been found:
2. DVD - Star Wars - Science Fiction - George Lucas - 87: 24.95$

```

Figure 19 Result

## 6. tring, StringBuilder and StringBuffer

### Code

```
1 package hust.soict.dsai.globalict.garbage;
2
3 import java.util.Random;
4
5 public class ConcatenationInLoops {
6     Run | Debug
7     public static void main(String[] args) {
8         Random r = new Random(seed:123);
9         long start = System.currentTimeMillis();
10        String s = "";
11        for(int i = 0; i < 65536; i++){
12            s += r.nextInt(bound:2);
13        }
14        System.out.println(System.currentTimeMillis() - start); //This prints roughly 4500.
15
16        r = new Random(seed:123);
17        start = System.currentTimeMillis();
18        StringBuilder sb = new StringBuilder();
19        for(int i = 0; i < 65536; i++){
20            sb.append(r.nextInt(bound:2));
21        }
22        s = sb.toString();
23        System.out.println(System.currentTimeMillis() - start); //This prints 5.
24    }
25 }
```

Figure 20 So sánh String và StringBuilder

### Result

```
399
2
```

## Code

```
1 package hust.soict.dsai.globalict.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     Run | Debug
9     public static void main(String[] args) {
10         try{
11             String filename = "C:\\Users\\ADMIN\\OneDrive\\Documents\\OOP\\OtherProjects\\src\\hust\\soict\\dsai\\globalict\\garbage\\text.txt";
12             byte[] inputBytes = {0};
13             long startTime, endTime;
14
15             inputBytes = Files.readAllBytes(Paths.get(filename));
16             startTime = System.currentTimeMillis();
17             String outputString = "";
18             for(byte b: inputBytes){
19                 outputString += (char)b;
20             }
21             endTime = System.currentTimeMillis();
22             System.out.println((endTime - startTime));
23         }catch(IOException ie){
24             ie.printStackTrace();
25         }
26     }
27 }
```

Figure 21 GarbageCreator class

## Result

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ADMIN\OneDrive\Documents\OOP\OtherProjects> & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:49749' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ADMIN\OneDrive\Documents\OOP\OtherProjects\bin' 'hust.soict.dsai.globalict.garbage.GarbageCreator'
```

(OtherProjects) Java: Ready Ln 18, Col 41 Spaces: 4 UTF-8 CRLF {} Java

Figure 22 Result: program hangs

```

1  package hust.soict.dsai.globalict.garbage;
2
3  import java.io.IOException;
4  import java.nio.file.Files;
5  import java.nio.file.Paths;
6  public class NoGarbage {
7      Run | Debug
8      public static void main(String[] args) {
9          try{
10             String filename = "C:\\Users\\ADMIN\\OneDrive\\Documents\\OOP\\OtherProjects\\src\\hust\\soict\\dsai\\globalict\\garbage\\text.txt";
11             byte[] inputBytes = { 0 };
12             long startTime, endTime;
13
14             inputBytes = Files.readAllBytes(Paths.get(filename));
15             startTime = System.currentTimeMillis();
16
17             // using StringBuilder
18             StringBuilder outputStringBuilder = new StringBuilder();
19             for (byte b : inputBytes) {
20                 outputStringBuilder.append((char) b);
21             }
22             endTime = System.currentTimeMillis();
23             System.out.println(endTime - startTime);
24         }catch(IOException ie){
25             ie.printStackTrace();
26         }
27     }
28 }

```

Figure 23 NoGarbage class

```

& 'C:\Program Files\Java\jdk-23\bin\java.exe' '-agentli
Messages' '-cp' 'C:\Users\ADMIN\OneDrive\Documents\OOP\O
12
PS C:\Users\ADMIN\OneDrive\Documents\OOP\OtherProjects

```

Figure 24 Result: Program finishes quickly

## 7 Class Diagram

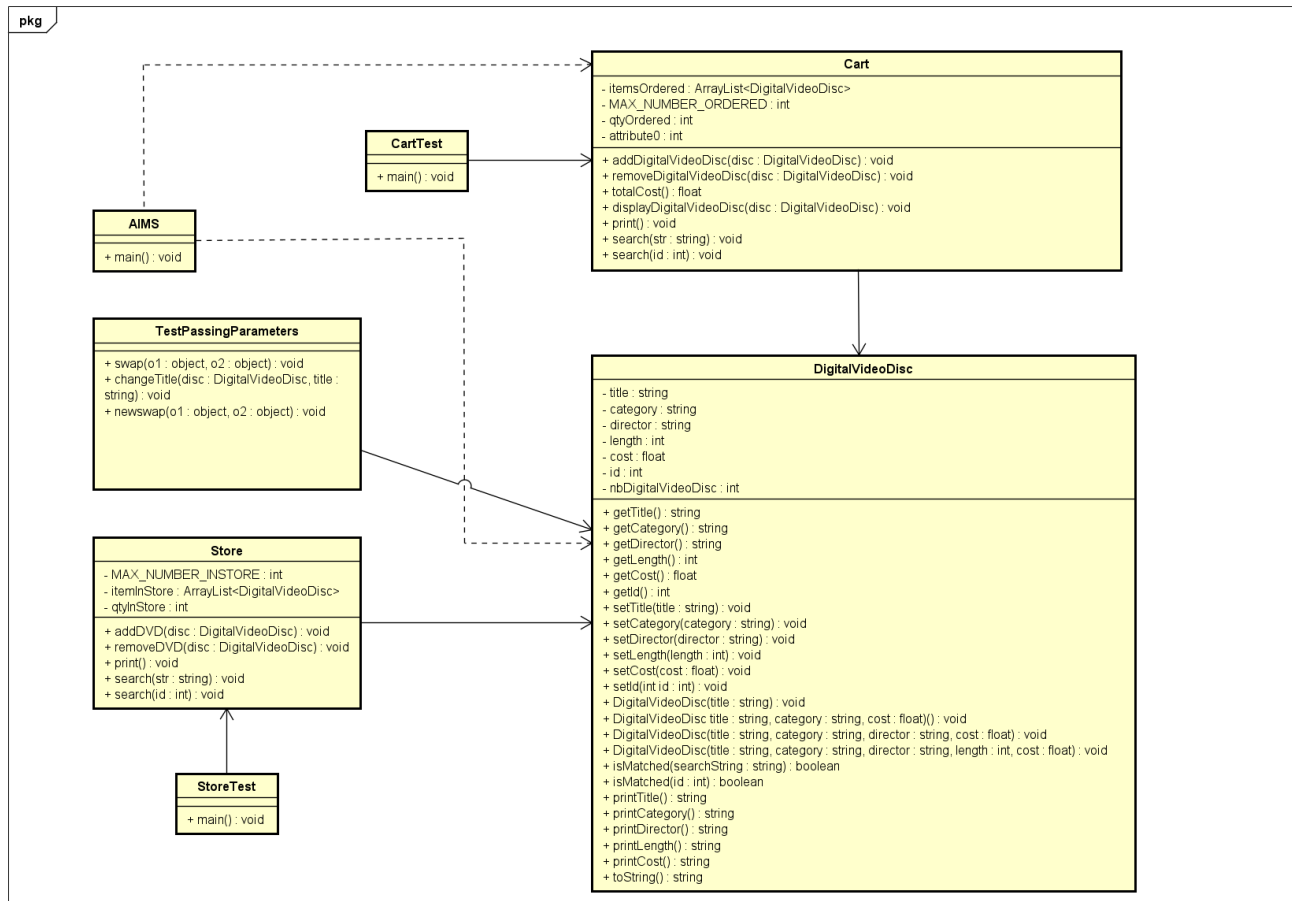


Figure 25 Class Diagram

## Table of Figures

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList) .....	2
Figure 2 Aims Class .....	2
Figure 3 Result.....	3
Figure 4 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList) .....	4
Figure 5 Aims Class .....	4
Figure 6 Result.....	5
Figure 7 Passing parameter code .....	6
Figure 8 Passing parameter result.....	6
Figure 9 Class member and Instance member .....	8
Figure 10 Result.....	9
Figure 11 Code method to print the content in Cart.....	9
Figure 12 Code method search by id.....	10
Figure 13 Code method search by title .....	10
Figure 14 IsMatch() in DigitalVideoDisc class .....	10
Figure 15 CartTest class.....	11
Figure 16 Result.....	11
Figure 17 Store Class .....	12
Figure 18 StoreTest class .....	13
Figure 19 Result.....	13
Figure 20 So sánh String và StringBuilder .....	14
Figure 21 GarbageCreator class.....	14
Figure 22 Result: program hangs.....	15
Figure 23 NoGarbage class.....	15
Figure 24 Result: Program finishes quickly.....	15
Figure 25 Class Diagram .....	16