

I. Working with method overloading

1. Overloading by differing types of parameter

Phương thức mới đầu tiên `addDigitalVideoDisc(DigitalVideoDisc[] dvdList)` cho phép thêm một danh sách các đĩa DVD vào giỏ hàng.

```
1 public int addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
2     int addCount = 0;
3     for (DigitalVideoDisc disc : dvdList) {
4         if (qtyOrdered == MAX_NUMBERS_ORDERED) {
5             System.out.println("The cart is almost full. Can't add more discs");
6             break;
7         } else {
8             itemsOrdered[qtyOrdered] = disc;
9             qtyOrdered++;
10            System.out.println("The DVD " + '\'' + disc.getTitle() + '\'' + " has been added!");
11            addCount++;
12        }
13    }
14    return addCount;
15 }
```

Phương thức mới thứ hai `addDigitalVideoDisc(DigitalVideoDisc... dvdArray)` sử dụng `varargs` để cho phép thêm một số lượng tùy ý các đĩa DVD vào giỏ hàng.

```
1 public int addDigitalVideoDisc(DigitalVideoDisc... dvdArray) {
2     int addCount = 0;
3     for (DigitalVideoDisc disc : dvdArray) {
4         if (qtyOrdered == MAX_NUMBERS_ORDERED) {
5             System.out.println("The cart is almost full. Can't add more discs");
6             break;
7         } else {
8             itemsOrdered[qtyOrdered] = disc;
9             qtyOrdered++;
10            System.out.println("The DVD " + '\'' + disc.getTitle() + '\'' + " has been added!");
11            addCount++;
12        }
13    }
14    return addCount;
15 }
```

2. Overloading by differing the number of parameters

```
1 public int addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
2     if (qtyOrdered + 1 ≥ MAX_NUMBERS_ORDERED) {
3         System.out.println("The cart is almost full. Can't add more discs");
4         return 0;
5     } else {
6         itemsOrdered[qtyOrdered] = dvd1;
7         qtyOrdered++;
8         System.out.println("The DVD " + "\"" + dvd1.getTitle() + "\" + " has been added!");
9
10        itemsOrdered[qtyOrdered] = dvd2;
11        qtyOrdered++;
12        System.out.println("The DVD " + "\"" + dvd2.getTitle() + "\" + " has been added!");
13
14        return 2;
15    }
```

II. Passing parameter

1. Trả lời câu hỏi

Câu hỏi: Java là ngôn ngữ lập trình "Pass by Value" hay "Pass by Reference"?

Trả lời: Java là ngôn ngữ lập trình "Pass by Value". Trong Java, khi bạn truyền một tham số cho một phương thức, giá trị của tham số được sao chép và truyền vào phương thức. Điều này có nghĩa là nếu bạn thay đổi giá trị của tham số bên trong phương thức, giá trị của biến gọi phương thức không bị ảnh hưởng.

2. Tạo class TestPassingParameter

Kết quả sau khi chạy:

```
Jungle dvd title: Jungle
Cinderella dvd title: Cinderella
Jungle dvd title: Cinderella
```

⇒ Pass by value

3. Trả lời câu hỏi

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

Trả lời: Sau khi thực hiện phương thức swap(jungleDVD, cinderellaDVD), tiêu đề của hai đối tượng vẫn giữ nguyên vì trong Java, tham số được truyền vào phương thức là giá trị của đối tượng, không phải là tham chiếu đến đối tượng. Khi ta thay đổi giá trị của tham số bên trong phương thức (như việc đổi chỗ giữa o1 và o2), sự thay đổi này không ảnh hưởng đến giá trị của các đối tượng gốc.

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

Trả lời: Sau khi gọi changeTitle(jungleDVD, cinderellaDVD.getTitle()), tiêu đề của jungleDVD bị thay đổi vì trong phương thức changeTitle, ta thực hiện thay đổi trực tiếp trên đối tượng dvd (được truyền vào phương thức) bằng cách gọi dvd.setTitle(title). Điều này ảnh hưởng trực tiếp đến đối tượng gốc được truyền vào phương thức.

4. Fix

Sửa hàm swap:

```
1 public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
2     DigitalVideoDisc tmp = new DigitalVideoDisc(dvd1.getTitle(), dvd1.getCategory(), dvd1.getDirector(), dvd1.getLength(), dvd1.getCost());
3
4     dvd1.setTitle(dvd2.getTitle());
5     dvd1.setCategory(dvd2.getCategory());
6     dvd1.setDirector(dvd2.getDirector());
7     dvd1.setLength(dvd2.getLength());
8     dvd1.setCost(dvd2.getCost());
9
10
11     dvd2.setTitle(tmp.getTitle());
12     dvd2.setCategory(tmp.getCategory());
13     dvd2.setDirector(tmp.getDirector());
14     dvd2.setLength(tmp.getLength());
15     dvd2.setCost(tmp.getCost());
16 }
```

Kết quả:

```
Jungle dvd title: Cinderella
Cinderella dvd title: Jungle
Jungle dvd title: Jungle
```

III. Use debug run

1. Debug với hàm swap cũ:

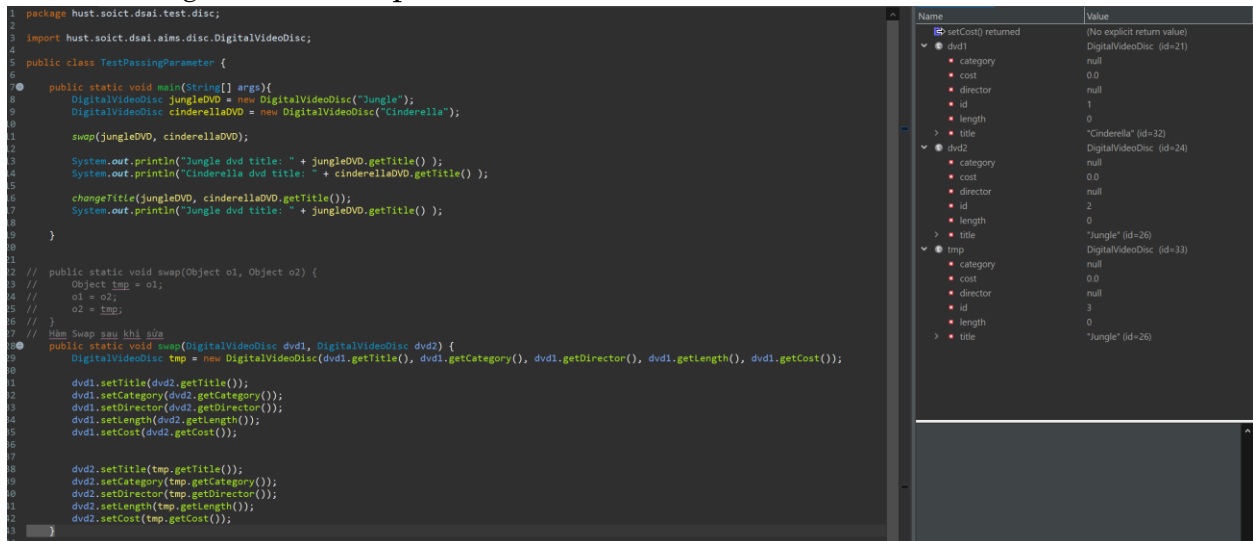


```
1 package hust.soict.dsai.test.disc;
2
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter {
6
7     public static void main(String[] args){
8         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
9         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
10
11         swap(jungleDVD, cinderellaDVD);
12
13         System.out.println("Jungle dvd title: " + jungleDVD.getTitle() );
14         System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle() );
15
16         changeTitle(jungleDVD, cinderellaDVD.getTitle());
17         System.out.println("Jungle dvd title: " + jungleDVD.getTitle() );
18
19     }
20
21     public static void swap(Object o1, Object o2) {
22         Object tmp = o1;
23         o1 = o2;
24         o2 = tmp;
25     }
26 }
27 // Hàm Swap sau khi sửa
28 // public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
```

Name	Value
no method return value	
o1	DigitalVideoDisc (id=24)
category	null
cost	0.0
director	null
id	2
length	0
title	"Cinderella" (id=32)
o2	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=26)
tmp	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=26)

⇒ 2 biến thay đổi là o1 và o2 chứ JungleDVD và CinderellaDVD ko đổi (Pass by Value)

2. Debug với hàm swap mới



```
1 package hust.soict.dsai.test.disc;
2
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4
5 public class TestPassingParameter {
6
7     public static void main(String[] args){
8         DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
9         DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
10
11         swap(jungleDVD, cinderellaDVD);
12
13         System.out.println("Jungle dvd title: " + jungleDVD.getTitle() );
14         System.out.println("Cinderella dvd title: " + cinderellaDVD.getTitle() );
15
16         changeTitle(jungleDVD, cinderellaDVD.getTitle());
17         System.out.println("Jungle dvd title: " + jungleDVD.getTitle() );
18
19     }
20
21     // public static void swap(Object o1, Object o2) {
22     //     Object tmp = o1;
23     //     o1 = o2;
24     //     o2 = tmp;
25     // }
26
27 // Hàm Swap sau khi sửa
28 public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
29     DigitalVideoDisc tmp = new DigitalVideoDisc(dvd1.getTitle(), dvd1.getCategory(), dvd1.getDirector(), dvd1.getLength(), dvd1.getCost());
30
31     dvd1.setTitle(dvd2.getTitle());
32     dvd1.setCategory(dvd2.getCategory());
33     dvd1.setDirector(dvd2.getDirector());
34     dvd1.setLength(dvd2.getLength());
35     dvd1.setCost(dvd2.getCost());
36
37     dvd2.setTitle(tmp.getTitle());
38     dvd2.setCategory(tmp.getCategory());
39     dvd2.setDirector(tmp.getDirector());
40     dvd2.setLength(tmp.getLength());
41     dvd2.setCost(tmp.getCost());
42 }
43 }
```

Name	Value
setCost() returned	(No explicit return value)
dvd1	DigitalVideoDisc (id=21)
category	null
cost	0.0
director	null
id	1
length	0
title	"Cinderella" (id=32)
dvd2	DigitalVideoDisc (id=24)
category	null
cost	0.0
director	null
id	2
length	0
title	"Jungle" (id=26)
tmp	DigitalVideoDisc (id=33)
category	null
cost	0.0
director	null
id	3
length	0
title	"Jungle" (id=26)

⇒ 2 biến JungleDVD và CinderellaDVD thay đổi (Pass By Reference)

IV. Classifier Member and Instance Member

```
1 package hust.soict.dsai.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
10    //Section 5: Classifier Member & Instance Member
11    private int id;
12    private static int nbDigitalVideoDisc = 0;
13
14    public DigitalVideoDisc(String title) {
15        this.title = title;
16        this.id = ++nbDigitalVideoDisc;
17    }
18
19    public DigitalVideoDisc(String title, String category, float cost) {
20        this.title = title;
21        this.category = category;
22        this.cost = cost;
23        this.id = ++nbDigitalVideoDisc;
24    }
25
26    public DigitalVideoDisc(String title, String category, String director, float cost) {
27        this.title = title;
28        this.category = category;
29        this.director = director;
30        this.cost = cost;
31        this.id = ++nbDigitalVideoDisc;
32    }
33
34    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
35        this.title = title;
36        this.category = category;
37        this.director = director;
38        this.length = length;
39        this.cost = cost;
40        this.id = ++nbDigitalVideoDisc;
41    }
42
43    public String getTitle(){
44        return title;
45    }
46    public String getCategory(){
47        return category;
48    }
49    public String getDirector(){
50        return director;
51    }
52    public int getLength(){
53        return length;
54    }
55    public float getCost(){
56        return cost;
57    }
58    public int getId() {
59        return id;
60    }
61
62    public void setTitle(String titleSet){
63        title = titleSet;
64    }
65    public void setCategory(String categorySet){
66        category = categorySet;
67    }
68    public void setDirector(String directorSet){
69        director = directorSet;
70    }
71    public void setLength(int lengthSet){
72        length = lengthSet;
73    }
74    public void setCost(float costSet){
75        cost = costSet;
76    }
77
78 }
```

V. Open the Cart class

1. Thêm hàm print trong cart.java

```
1 public void printCart() {
2     System.out.println("*****CART*****");
3     System.out.println("Ordered Items:");
4     for (int i = 0; i < qtyOrdered;i++) {
5         System.out.println(i+1 + "[" + itemsOrdered[i].getTitle() + "] - [" + itemsOrdered[i].getCategory() + "] - ["
6             + itemsOrdered[i].getDirector() + "] - [" + itemsOrdered[i].getLength() + "]: "
7             + itemsOrdered[i].getCost() + " $\n");
8     }
9     System.out.println("total: " + this.totalCost() + " $\n");
10    System.out.println("*****");
11 }
```

2. Chạy thử và xem kết quả

```
*****CART*****
Ordered Items:
1.[The Lion King] - [Animation] - [Roger Allers] - [87]: 19.95 $
2.[Star Wars] - [Science Fiction] - [George Lucas] - [87]: 24.95 $
3.[Aladin] - [Animation] - [null] - [0]: 18.99 $

total: 63.89 $

*****
```

VI. Implement the Store class

1. Tạo lớp Store

```
1 package hust.soict.dsai.aims.store;
2 import java.util.ArrayList;
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4
5 public class Store {
6     private ArrayList<DigitalVideoDisc> itemsInStore;
7
8     public Store() {
9         itemsInStore = new ArrayList<>();
10    }
11
12    private boolean checkDVD(DigitalVideoDisc disc) {
13        for (DigitalVideoDisc digitalVideoDisc : itemsInStore) {
14            if (digitalVideoDisc.equals(disc)) {
15                return true;
16            }
17        }
18        return false;
19    }
20
21    public void addDVD(DigitalVideoDisc dvd) {
22        if(!itemsInStore.contains(dvd)) {
23            itemsInStore.add(dvd);
24            System.out.println( dvd.getTitle() + " 've been added to the store !");
25        }
26        else
27            System.out.println(dvd.getTitle() + " 'already exists in the store !");
28    }
29
30    public void removeDVD(DigitalVideoDisc dvd) {
31        if(itemsInStore.contains(dvd)) {
32            itemsInStore.remove(dvd);
33            System.out.println( dvd.getTitle() + " 've been deleted from the store !");
34        }
35        else
36            System.out.println("DVD not found!");
37    }
38 }
39
```

2. Tạo lớp TestStore và kiểm thử

```
1 package hust.soict.dsai.test.store;
2
3 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
4 import hust.soict.dsai.aims.store.Store;
5
6 public class StoreTest {
7     public static void main(String[] args) {
8         Store store = new Store();
9         DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation",
10             "Roger Allers", 87, 19.95f);
11         DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star wars", "Science Fiction",
12             "Geogre Lucas", 87, 24.95f);
13         DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
14
15         //Add to store
16         store.addDVD(dvd1);
17         store.addDVD(dvd2);
18         store.addDVD(dvd3);
19
20         //Remove from store
21         store.removeDVD(dvd1);
22         store.removeDVD(dvd2);
23         store.removeDVD(dvd3);
24
25     }
26 }
```

Kết quả kiểm thử:

```
The Lion King 've been added to the store !
Star wars 've been added to the store !
Aladin 've been added to the store !
The Lion King 've been deleted from the store !
Star wars 've been deleted from the store !
Aladin 've been deleted from the store !
```


VII. String, StringBuilder and StringBuffer

1. Tạo lớp ConcatenationInLoops

```
1 package hust.soict.dsai.garbage;
2 import java.util.Random;
3
4 public class ConcatenationInLoops {
5     public static void main(String[] args) {
6         Random r = new Random(123);
7         long start = System.currentTimeMillis();
8         String s = "";
9         for (int i = 0; i < 65536; i++) {
10             s += r.nextInt(2);
11         }
12         System.out.println(System.currentTimeMillis() - start);
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         }
20         s = sb.toString();
21         System.out.println(System.currentTimeMillis() - start);
22     }
23 }
24
```

2. Tạo lớp GarbageCreator

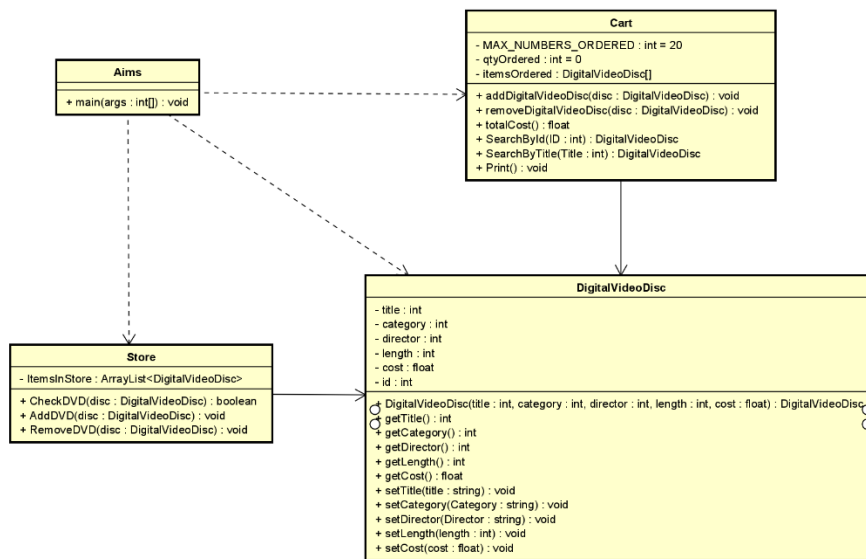
```
1 import java.io.IOException;
2 import java.nio.file.Files;
3 import java.nio.file.Paths;
4
5 public class GarbageCreator {
6     public static void main(String[] args) throws IOException {
7         String filename = "C:\\Users\\MinhTrinh\\eclipse-workspace\\AimsProject\\OtherProjects\\garbage\\text.txt";
8         byte[] inputBytes = { 0 };
9         long startTime, endTime;
10         inputBytes = Files.readAllBytes(Paths.get(filename));
11         startTime = System.currentTimeMillis();
12         String outputString = "";
13         for(byte b : inputBytes) {
14             outputString += (char)b;
15         }
16         endTime = System.currentTimeMillis();
17         System.out.println(endTime - startTime);
18     }
19 }
```

3. Tạo lớp NoGarbage

```
1 package hust.soict.dsai.garbage;
2
3 public class NoGarbage {
4     public static void main(String[] args) throws IOException {
5         String filename = "C:\\Users\\MinhTrinh\\eclipse-workspace\\AimsProject\\OtherProjects\\garbage\\text.txt";
6         byte[] inputBytes = { 0 };
7         long startTime, endTime;
8         inputBytes = Files.readAllBytes(Paths.get(filename));
9         startTime = System.currentTimeMillis();
10        String outputString = "";
11        for(byte b : inputBytes) {
12            outputString += (char)b;
13        }
14        endTime = System.currentTimeMillis();
15        System.out.println(endTime - startTime);
16    }
17 }
18
```

VIII. Cập nhật các Diagram trong project AIMS

1. Class diagram



2. Use-case diagram

