Report Lab 03: Basic Object-Oriented Techniques

2. Working with method overloading

New code:

```
12      // Add a list of DVDs using an array
13      public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
14          for (DigitalVideoDisc disc : dvdList) {
15              if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
16                  itemsOrdered.add(disc);
17                  System.out.println("The disc \"" + disc.getTitle() + "\" has been added");
18              } else {
19                  System.out.println("The cart is full. Cannot add \"" + disc.getTitle() + "\"");
20                  break;
21              }
22          }
23      }
24
```

```
25      // Add two DVDs at once
26      public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
27          if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
28              itemsOrdered.add(dvd1);
29              System.out.println("The disc \"" + dvd1.getTitle() + "\" has been added");
30          } else {
31              System.out.println("The cart is full. Cannot add \"" + dvd1.getTitle() + "\"");
32          }
33
34          if (itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
35              itemsOrdered.add(dvd2);
36              System.out.println("The disc \"" + dvd2.getTitle() + "\" has been added");
37          } else {
38              System.out.println("The cart is full. Cannot add \"" + dvd2.getTitle() + "\"");
39          }
40      }
```

Add some DVDs:

```
10      // Create some DVDs
11      DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:87, cos
12      DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:87, c
13      DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladdin", category:"Animation", cost:18.99f);
14
15      // Add individual DVDs to the cart
16      anOrder.addDigitalVideoDisc(dvd1, dvd2);
17
18
19      // Create a DVD array using existing DVDs
20      DigitalVideoDisc[] dvdArray = { dvd1, dvd2, dvd3 };
21
22      // Add multiple DVDs to the cart using an array
23      anOrder.addDigitalVideoDisc(dvdArray);
24
```

Result:

```
PS C:\project code\OOPLab> cd AimsProject
PS C:\project code\OOPLab\AimsProject> java -cp bin hust.soict.dsai.aims.Aims
The disc "The Lion King" has been added
The disc "Star Wars" has been added
The disc "The Lion King" has been added
The disc "Star Wars" has been added
The disc "Aladdin" has been added
```

3. Passing parameter

Update new code in DigitalVideoDisc.java:

```
17      public String getTitle() {
18          return title;
19      }
```

Create TestPassingParameter.java:

```
public static void main(String[] args) {
    // Create two DVD objects
    DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");
    DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"Cinderella");

    // Print the titles before swapping
    System.out.println(x:"Before swap:");
    System.out.println("jungle dvd title: " + jungleDVD.getTitle());
    System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());

    // Use an array to swap the two DVD objects
    DigitalVideoDisc[] dvdArray = {jungleDVD, cinderellaDVD};
    swap(dvdArray);

    // Print the titles after swapping
    System.out.println(x:"\nAfter swap (correct):");
    System.out.println("jungle dvd title: " + dvdArray[0].getTitle());
    System.out.println("cinderella dvd title: " + dvdArray[1].getTitle());

    // Change the title of jungleDVD to cinderellaDVD's title
    changeTitle(jungleDVD, cinderellaDVD.getTitle());
    System.out.println(x:"\nAfter changeTitle:");
    System.out.println("jungle dvd title: " + jungleDVD.getTitle());
}
```

```
30
31      // Swap the first two DVD objects in the array
32      public static void swap(DigitalVideoDisc[] dvds) {
33          if (dvds.length >= 2) {
34              DigitalVideoDisc temp = dvds[0];
35              dvds[0] = dvds[1];
36              dvds[1] = temp;
37          }
38      }
39
40      // Change the title of the given DVD object
41      public static void changeTitle(DigitalVideoDisc dvd, String title) {
42          dvd.setTitle(title);
43      }
44  }
45
```

Result:

```
PS C:\project code\OOPLab\AimsProject> java -cp bin hust.soict.dsai.test.disc.TestPassingParameter
Before swap:
jungle dvd title: Jungle
cinderella dvd title: Cinderella

After swap (correct):
jungle dvd title: Cinderella
cinderella dvd title: Jungle

After changeTitle:
jungle dvd title: Cinderella
PS C:\project code\OOPLab\AimsProject>
```

4. Use debug run:

Debugging successful:

```
30
31      public static void swap(DigitalVideoDisc[] dvds) {
32          DigitalVideoDisc temp = dvds[0];
33          dvds[0] = dvds[1];
34          dvds[1] = temp;
35      }
```

5. Classifier Member and Instance Member

Update new code in DigitalVideoDisc.java:

```
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       private static int nbDigitalVideoDiscs = 0;
10      private int id;
```

Result:

```
PS C:\project code\OOPLab> cd AimsProject
PS C:\project code\OOPLab\AimsProject> java -cp bin hust.soict.dsai.aims.Aims
The disc "The Lion King" has been added
The disc "Star Wars" has been added
The disc "The Lion King" has been added
The disc "Star Wars" has been added
The disc "Aladdin" has been added
Cart Contents:
ID: 1
Title: The Lion King
Category: Animation
Director: Roger Allers
Length: 87 minutes
Cost: $19.95

ID: 2
Title: Star Wars
Category: Science Fiction
Director: George Lucas
Length: 87 minutes
Cost: $24.95

ID: 1
Title: The Lion King
Category: Animation
Director: Roger Allers
Length: 87 minutes
```

```
ID: 2
Title: Star Wars
Category: Science Fiction
Director: George Lucas
Length: 87 minutes
Cost: $24.95

ID: 3
Title: Aladdin
Category: Animation
Director: Unknown
Length: Unknown
Cost: $18.99

Total cost: $108.79
Playing a DVD:
Playing DVD: The Lion King
DVD length: 87 minutes
```

6. Open the Cart class

Update totalCost, searchById and print in Cart.java:

```java
// Calculate total cost of the DVDs in the cart
public float totalCost() {
    float total = 0;
    for (DigitalVideoDisc disc : itemsOrdered) {
        total += disc.getCost();
    }
    return total;
}
```

```java
// Search for DVDs by ID
public DigitalVideoDisc searchById(int id) {
    for (DigitalVideoDisc disc : itemsOrdered) {
        if (disc.getId() == id) {
            return disc;
        }
    }
    return null;
}
```

```java
// Print the list of DVDs in the cart
public void print() {
    System.out.println(x:"**********CART**********");
    System.out.println(x:"Ordered Items:");
    float totalCost = 0;
    for (int i = 0; i < itemsOrdered.size(); i++) {
        DigitalVideoDisc disc = itemsOrdered.get(i);
        System.out.println((i + 1) + ". " + disc.toString());
        totalCost += disc.getCost();
    }
    System.out.println("Total cost: $" + totalCost);
    System.out.println(x:"**************************************************");
}
```

Create new file CartTest.java:

```java
package hust.soict.dsai.test.cart;

import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

public class CartTest {
    Run | Debug
    public static void main(String[] args) {
        // Create a new cart
        Cart cart = new Cart();

        // Create DVDs and add them to the cart
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:87, cos
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:124,
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladdin", category:"Animation", director:"John Musker", length:90, cost:18.99
        DigitalVideoDisc[] dvdArray = { dvd1, dvd2, dvd3 };
        cart.addDigitalVideoDisc(dvdArray);

        // Test the print method
        cart.print();

        // Test searching by ID
        System.out.println(x:"\n******************Search by ID******************");
        DigitalVideoDisc searchById = cart.searchById(id:2); // Searching for ID = 2
        System.out.println(searchById != null ? searchById.toString() : "No match found.");
        System.out.println(x:"************************************************");

    }
}

```

Result:

```
PS C:\project code\OOPLab\AimsProject> javac -d bin -sourcepath src src/hust/soict/dsai/test/cart/CartTest.java
PS C:\project code\OOPLab\AimsProject> java -cp bin hust.soict.dsai.test.cart.CartTest
The disc "The Lion King" has been added
The disc "Star Wars" has been added
The disc "Aladdin" has been added
**********CART**********
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers - 87 minutes: $19.95
2. DVD - Star Wars - Science Fiction - George Lucas - 124 minutes: $24.95
3. DVD - Aladdin - Animation - John Musker - 90 minutes: $18.99
Total cost: $63.89
**********

**********Search by ID**********
DVD - Star Wars - Science Fiction - George Lucas - 124 minutes: $24.95
**********
PS C:\project code\OOPLab\AimsProject>
```

7. Implement the Store class

Create Store.java:

```
8   public class Store {
9       // List of DVDs available in the store
10      private List<DigitalVideoDisc> itemsInStore;
11
12      // Constructor
13      public Store() {
14          itemsInStore = new ArrayList<>(); // Initialize the list
15      }
16
17      // Add a DVD to the store
18      public void addDVD(DigitalVideoDisc dvd) {
19          itemsInStore.add(dvd);
20          System.out.println("The DVD \"" + dvd.getTitle() + "\" has been added to the store.");
21      }
22
23      // Remove a DVD from the store
24      public void removeDVD(DigitalVideoDisc dvd) {
25          if (itemsInStore.remove(dvd)) {
26              System.out.println("The DVD \"" + dvd.getTitle() + "\" has been removed from the store.");
27          } else {
28              System.out.println("The DVD \"" + dvd.getTitle() + "\" was not found in the store.");
29          }
30      }
31
32      // Display all DVDs in the store
33      public void displayStore() {
34          System.out.println(x:"Items in the store:");
35          for (DigitalVideoDisc dvd : itemsInStore) {
36              System.out.println(dvd.toString());
37          }
38          System.out.println();
```

Create StoreTest.java:

```
6   public class StoreTest {
        Run | Debug
7       public static void main(String[] args) {
8           // Create a new store
9           Store store = new Store();
10
11          // Create some DVDs
12          DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:87, co
13          DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Science Fiction", director:"George Lucas", length:124,
14          DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladdin", category:"Animation", director:"John Musker", length:90, cost:18.9
15
16          // Add DVDs to the store
17          store.addDVD(dvd1);
18          store.addDVD(dvd2);
19          store.addDVD(dvd3);
20
21          // Display items in the store
22          System.out.println(x:"\n**********Store Contents**********");
23          store.displayStore();
24
25          // Remove a DVD from the store
26          System.out.println(x:"\n**********Remove a DVD**********");
27          store.removeDVD(dvd2);
28
29          // Display items in the store after removal
30          System.out.println(x:"\n**********Store Contents After Removal**********");
31          store.displayStore();
```

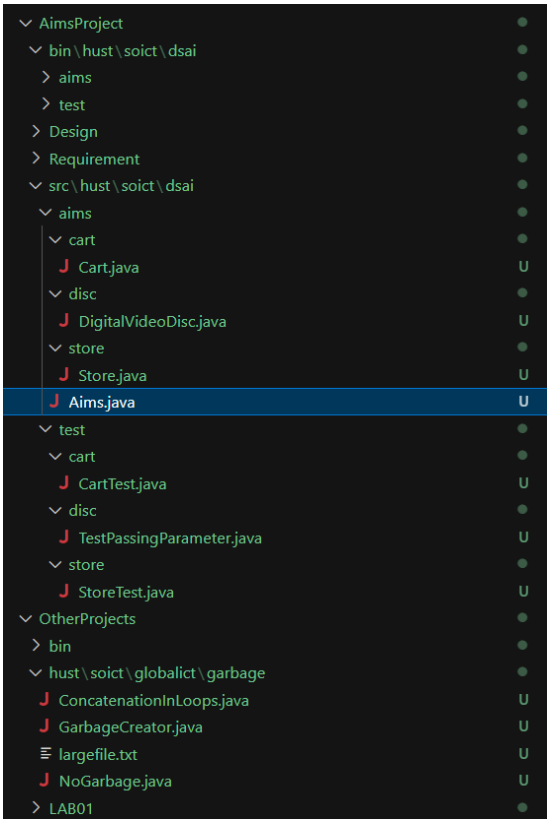
```
32
33          // Try to remove a non-existent DVD
34          System.out.println(x:"\n**********Attempt to Remove Non-existent DVD**********");
35          store.removeDVD(dvd2); // Star Wars was already removed
36      }
37  }
38
```

Result:

```
PS C:\project code\OOPLab\AimsProject> java -cp bin hust.soict.dsai.test.store.StoreTest
The DVD "The Lion King" has been added to the store.
The DVD "Star Wars" has been added to the store.
The DVD "Aladdin" has been added to the store.

**********Store Contents**********
Items in the store:
DVD - The Lion King - Animation - Roger Allers - 87 minutes: $19.95
DVD - Star Wars - Science Fiction - George Lucas - 124 minutes: $24.95
DVD - Aladdin - Animation - John Musker - 90 minutes: $18.99


**********Remove a DVD**********
The DVD "Star Wars" has been removed from the store.

**********Store Contents After Removal**********
Items in the store:
DVD - The Lion King - Animation - Roger Allers - 87 minutes: $19.95
DVD - Aladdin - Animation - John Musker - 90 minutes: $18.99


**********Attempt to Remove Non-existent DVD**********
The DVD "Star Wars" was not found in the store.
PS C:\project code\OOPLab\AimsProject>
```

8. Re-organize your projects

Project's entire tree:



8.1. String, StringBuilder and StringBuffer

Create ConcetanationInLoops.java:

```java
package hust.soict.globalict.garbage;

import java.util.Random;

public class ConcatenationInLoops {
    Run | Debug
    public static void main(String[] args) {
        Random r = new Random(seed:123);

        // Using String with "+"
        long start = System.currentTimeMillis();
        @SuppressWarnings("unused")
        String s = "";
        for (int i = 0; i < 65536; i++) {
            s += r.nextInt(bound:2);
        }
        System.out.println("String with +: " + (System.currentTimeMillis() - start) + " ms");

        // Using StringBuilder
        r = new Random(seed:123);
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 65536; i++) {
            sb.append(r.nextInt(bound:2));
        }
        s = sb.toString();
        System.out.println("StringBuilder: " + (System.currentTimeMillis() - start) + " ms");

        // Using StringBuffer
        r = new Random(seed:123);
        start = System.currentTimeMillis();
        StringBuffer sbf = new StringBuffer();
        for (int i = 0; i < 65536; i++) {
            sbf.append(r.nextInt(bound:2));
        }
        s = sbf.toString();
        System.out.println("StringBuffer: " + (System.currentTimeMillis() - start) + " ms");
```

Create GarbageCreator.java:

OtherProjects > hust > soict > globalict > garbage > GarbageCreator.java > GarbageCreator > main(String[])

```java
package hust.soict.globalict.garbage;

import java.nio.file.Files;
import java.nio.file.Paths;

public class GarbageCreator {
    Run | Debug
    public static void main(String[] args) throws Exception {
        String filename = "hust/soict/globalict/garbage/largefile.txt"; // File văn bản lớn
        byte[] inputBytes = Files.readAllBytes(Paths.get(filename));
        long startTime = System.currentTimeMillis();

        @SuppressWarnings("unused")
        String outputString = "";
        for (byte b : inputBytes) {
            outputString += (char) b;
        }

        long endTime = System.currentTimeMillis();
        System.out.println("Time taken with String concatenation: " + (endTime - startTime) + " ms");
    }
}
```

Create NoGarbage.java:

OtherProjects > hust > soict > globalict > garbage > NoGarbage.java > NoGarbage > main(String[])

```java
package hust.soict.globalict.garbage;

import java.nio.file.Files;
import java.nio.file.Paths;

public class NoGarbage {
    Run | Debug
    public static void main(String[] args) throws Exception {
        String filename = "hust/soict/globalict/garbage/largefile.txt"; // File văn bản lớn
        byte[] inputBytes = Files.readAllBytes(Paths.get(filename));
        long startTime = System.currentTimeMillis();

        StringBuilder outputBuilder = new StringBuilder();
        for (byte b : inputBytes) {
            outputBuilder.append((char) b);
        }
        @SuppressWarnings("unused")
        String outputString = outputBuilder.toString();

        long endTime = System.currentTimeMillis();
        System.out.println("Time taken with StringBuilder: " + (endTime - startTime) + " ms");
    }
}

```

Results:

```
PS C:\project code\OOPLab\OtherProjects> java -cp bin hust.soict.globalict.garbage.ConcatenationInLoops
String with +: 1788 ms
StringBuilder: 1 ms
StringBuffer: 76 ms
PS C:\project code\OOPLab\OtherProjects> java -cp bin hust.soict.globalict.garbage.GarbageCreator
Time taken with String concatenation: 1112 ms
PS C:\project code\OOPLab\OtherProjects> java -cp bin hust.soict.globalict.garbage.NoGarbage
Time taken with StringBuilder: 1 ms
PS C:\project code\OOPLab\OtherProjects>
```

Answer the questions in the section:

<u>English version:</u>

Question 1: Is JAVA a Pass by Value or a Pass by Reference programming language?

Answer:

Java is a Pass by Value programming language.

- *For primitive types:* Java passes the actual value of the variable, meaning any changes inside the method do not affect the variable outside.
- *For objects:* Java passes a copy of the reference to the object. This allows changes to the object's attributes, but the reference itself cannot be altered (e.g., swapping objects does not work).

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

Answer:

When the swap method is called:

- The references to the objects jungleDVD and cinderellaDVD are copied.

- Inside the method, o1 and o2 are the local copies of the original references.
- Assigning o1 = o2 only modifies these local copies and does not affect the original references outside the method.

As a result, the objects jungleDVD and cinderellaDVD outside the method are not swapped, and their titles remain unchanged.

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

Answer:

- When jungleDVD is passed to the changeTitle method:
- The reference to jungleDVD is copied.
- Inside the method, calling dvd.setTitle(title) modifies the title attribute of the object that dvd references, which is the same object as jungleDVD.

The creation of a new object inside the method (e.g., new DigitalVideoDisc) does not affect the original jungleDVD because dvd is only a copy of the reference.

Conclusion:

Java is a Pass by Value programming language:

- For objects: Java passes the value of the reference (a copy of the reference), not the actual reference itself. This means:
    + The contents of the object being referenced (e.g., title) can be modified.
    + The reference itself cannot be modified (e.g., swapping two objects does not work).
- For primitive types: Java passes the actual value of the variable, and changes inside the method do not affect the variable outside.

Vietnamese version:

Câu hỏi 1: Java là ngôn ngữ lập trình truyền tham trị (Pass by Value) hay truyền tham chiếu (Pass by Reference)?

Java là một ngôn ngữ lập trình truyền tham trị (Pass by Value).

- Với kiểu dữ liệu nguyên thủy (**primitive types**): Java truyền giá trị thực của biến, nghĩa là mọi thay đổi trong phương thức không ảnh hưởng đến biến bên ngoài.
- Với đối tượng (**objects**): Java truyền bản sao của tham chiếu đến đối tượng. Điều này cho phép thay đổi các thuộc tính của đối tượng, nhưng tham chiếu gốc không thể bị thay đổi (ví dụ: không thể hoán đổi hai đối tượng).

Câu hỏi 2: Tại sao sau khi gọi swap(jungleDVD, cinderellaDVD), tiêu đề của hai đối tượng vẫn không thay đổi?

Khi phương thức swap được gọi:

- Tham chiếu đến các đối tượng jungleDVD và cinderellaDVD được sao chép.
- Trong phương thức, o1 và o2 là các bản sao cục bộ của tham chiếu gốc.
- Việc gán o1 = o2 chỉ thay đổi các bản sao này, không ảnh hưởng đến tham chiếu gốc bên ngoài phương thức.

=> Do đó, các đối tượng jungleDVD và cinderellaDVD bên ngoài phương thức không bị hoán đổi, và tiêu đề của chúng không thay đổi.

Câu hỏi 3: Tại sao sau khi gọi changeTitle(jungleDVD, cinderellaDVD.getTitle()), tiêu đề của jungleDVD lại thay đổi?

Khi jungleDVD được truyền vào phương thức changeTitle:

- Tham chiếu đến jungleDVD được sao chép.
- Trong phương thức, khi gọi dvd.setTitle(title), thuộc tính title của đối tượng mà dvd tham chiếu được thay đổi. Đây chính là đối tượng jungleDVD bên ngoài.

Việc tạo một đối tượng mới trong phương thức (ví dụ: new DigitalVideoDisc) không ảnh hưởng đến đối tượng jungleDVD ban đầu, vì dvd chỉ là một bản sao của tham chiếu.

Kết luận:

Java là một ngôn ngữ lập trình truyền tham trị (Pass by Value):

- **Với object:** Java truyền giá trị của tham chiếu (bản sao của tham chiếu), không phải tham chiếu thực sự. Điều này có nghĩa là:
    + Nội dung của đối tượng được tham chiếu (ví dụ: thuộc tính title) có thể thay đổi.
    + Tham chiếu gốc không thể thay đổi (ví dụ: không thể hoán đổi hai đối tượng).
- **Với kiểu dữ liệu nguyên thủy**: Java truyền giá trị thực của biến, và mọi thay đổi trong phương thức không ảnh hưởng đến giá trị bên ngoài.