Oriented Objects Programming

Trần Ngọc Minh

20235608

Lab 03: Basic Object-Oriented Techniques

1. Working with method overloading:

a. Overloading by differing types of parameter:

```java
public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private DigitalVideoDisc[] items = new DigitalVideoDisc[MAX_NUMBERS_ORDERED];
    private int qtyOrdered = 0;

    public void addDigitalVideoDisc(DigitalVideoDisc disc){
        if(qtyOrdered >= MAX_NUMBERS_ORDERED){
            System.out.println("The cart is full.");
        }
        else{
            items[qtyOrdered++] = disc;
            System.out.println("The disc has been added.");
        }
    }
```

b. Overloading by differing the number of parameters

```java
public void addDigitalVideoDisc(DigitalVideoDisc[] list){
    if(qtyOrdered >= MAX_NUMBERS_ORDERED){
        System.out.println("The cart is full.");
    }
    else{
        for(int i = 0; i < list.length(); i++){
            items[qtyOrdered++] = list[i];
            System.out.println("The cart has been added.");
        }
    }
}

public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2){
    if(qtyOrdered >= MAX_NUMBERS_ORDERED){
        System.out.println("The cart is full.");
    }
    else{
        items[qtyOrdered++] = dvd1;
        if(qtyOrdered >= MAX_NUMBERS_ORDERED){
            System.out.println("The first cart has been added. The another one is impossible because the cart is full.");
        }
        else{
            items[qtyOrdered++] = dvd2;
            System.out.println("The cart has been added.");
        }
    }
}
```

2. Passing parameters

```java
package hust.soict.dsai.test.disc;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

public class Testing{
    public static void main(String[] args){
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

        swap(jungleDVD, cinderellaDVD);
        System.out.println("Title jungle DVD: " + jungleDVD.getTitle());
        System.out.println("Title cinderella DVD: " + cinderellaDVD.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Title jungle DVD: " + jungleDVD.getTitle());
    }

    public static void swap(Object Obj1, Object Obj2){
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    public static void changeTitle(DigitalVideoDisc dvd, String title){
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
    }
}
```

```
Title jungle DVD: Jungle
Title cinderella DVD: Cinderella
Title jungle DVD: Jungle
```

Modified Source Code

```java
package hust.soict.dsai.test.disc;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

public class Testing{
    Run | Debug
    public static void main(String[] args){
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"Cinderella");

        swap(jungleDVD, cinderellaDVD);
        System.out.println("Title jungle DVD: " + jungleDVD.getTitle());
        System.out.println("Title cinderella DVD: " + cinderellaDVD.getTitle());

        changeTitle(jungleDVD, cinderellaDVD.getTitle());
        System.out.println("Title jungle DVD: " + jungleDVD.getTitle());
    }

    public static void swap(DigitalVideoDisc Obj1, DigitalVideoDisc Obj2){
        DigitalVideoDisc tmp = new DigitalVideoDisc(title:"");
        tmp.setTitle(Obj1.getTitle());
        Obj1.setTitle(Obj2.getTitle());
        Obj2.setTitle(tmp.getTitle());
    }

    public static void changeTitle(DigitalVideoDisc dvd, String title){
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
    }
}
```

Question:

1. After the call of `swap(jungleDVD, cinderellaDVD)` why does the title of these two objects still remain?
   Answer: When the swap function is invoked, it generates new references for the two objects, jungleDVD and cinderellaDVD, which we can call o1 and o2. As a result, any modifications made to o1 and o2 do not impact the original objects. Consequently, the titles of jungleDVD and cinderellaDVD remain unchanged after the swap function executes.
2. **After the call of changeTitle(jungleDVD, cinderellaDVD.getTitle()) why is the title of the JungleDVD changed?**
   Answer: When the function is called, it creates a new reference named dvd that points to the jungleDVD object. Therefore, any changes made to the title through the setTitle method using the dvd reference will directly affect the jungleDVD object. This is because both dvd and jungleDVD refer to the same underlying object in memory. So, updating the title via dvd will also update it for jungleDVD.

3. Class Member & Insane:

```java
public class DigitalVideoDisc {
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;
    // Create attribute id and nbDigitalVideoDisc
    private static int nbDigitalVideoDisc = 0;
    private int id;
```

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

4. Cart Class

- Writing method to print items in cart

```java
public String toString(){
    return "DVD: " + title + " " + category + " " + director + " " + length + " " + cost;
}
```

```
public void printStore(){
    total = 0;
    System.out.println(x:"Ordered Items:");
    for (int i : items){
        System.out.println((i+1) + ". " + items.get(i));
    }
    System.out.println("Total cost: " + totalCost());
}
```

- Writing method to searching:

```
public boolean Match(String Title){
    return this.title.equals(title);
}
```

```
public void SearchTitle(String Title){
    boolean found = false;
    for(int i = 0; i < qtyOrdered; i++){
        if(items[i] != null && items[i].Match(i)){
            System.out.println(items[i].getTitle());
        }
    }
    found = true;
}
if(!found){
    System.out.println(x:"The DVD you search is not matching!");
}
}
```

```
public void SearchId(int id){
    for(int i = 0; i < qtyOrdered; i++){
        if(items[i] != null && items[i].getID() == id){
            System.out.println(id);
            return;
        }
    }
}
```

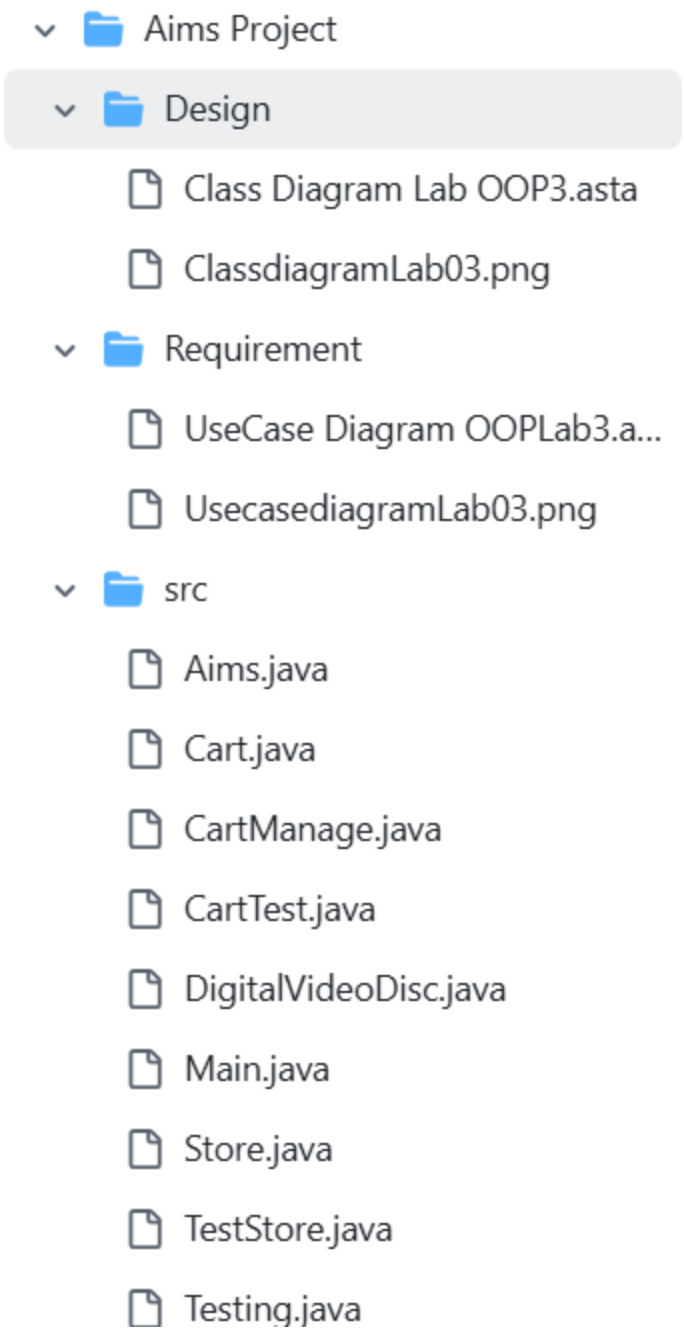
```java
package hust.soict.dsai.cart;
import hust.soict.dsai.cart.Cart;
import hust.soict.dsai.disc.DigitalVideoDisc;

public class CartTest{
    Run | Debug
    public static void main(String[] args){
        Cart cart = new Cart();
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        cart.addDigitalVideoDisc(dvd1);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addDigitalVideoDisc(dvd2);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.89f);
        cart.addDigitalVideoDisc(dvd3);
        cart.print();
    }
}
```

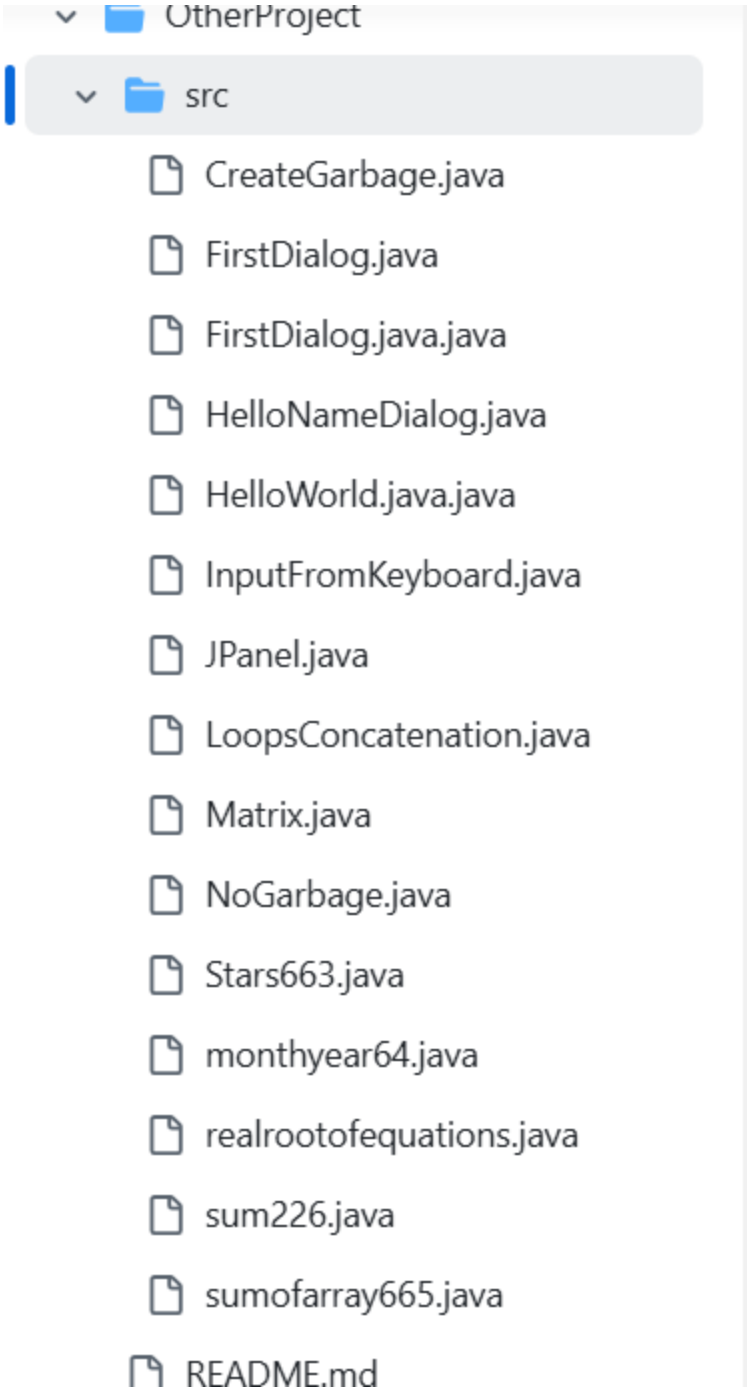
5. Store Class

```java
package hust.soict.dsai.store;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;
import java.util.List;
import java.util.ArrayListed;

public class Store{
    private List<DigitalVideoDisc> items = new ArrayListed<DigitalVideoDisc>();

    public void addDVD(DigitalVideoDisc dvd){
        items.add(dvd);
        System.out.println(x:"The dvd has been added.");
    }

    public void removeDVD(DigitalVideoDisc dvd){
        boolean delete = items.remove(dvd);
        if(deleted){
            System.out.println(dvd.getTitle() + " has been removed.");
        }
        else{
            System.out.println(dvd.getTitle() + " is not founded in items.");
        }
    }

    public void printStore(){
        total = 0;
        System.out.println(x:"Ordered Items:");
        for (int i : items){
            System.out.println((i+1) + ". " + items.get(i));
        }
        System.out.println("Total cost: " + totalCost());
    }
}
```

```java
package hust.soict.dsai.test.store;
import hust.soict.dsai.aims.store.Store;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

public class TestStore{
    Run | Debug
    public static void main(String[] args){
        Store store = new Store();
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King", "Animation", "Roger Allers", 87, 19.95f);
        store.addDVD(dvd1);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        store.addDVD(dvd2);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin", "Animation", 18.89f);
        store.addDVD(dvd3);
        store.printStore();
        store.removeDVD(dvd2);
        store.removeDVD(dvd3);
        store.printStore();
        DigitalVideoDisc dvd4 = new DigitalVideoDisc("Frozen", "Animation", "Disney", 95, 16.0f);
        store.remove(dvd4);
        store.printStore();
    }
}
```

6. Reorganize projects


Aims Project
Design
Class Diagram Lab OOP3.asta
ClassdiagramLab03.png
Requirement
UseCase Diagram OOPLab3.a...
UsecasediagramLab03.png
src
Aims.java
Cart.java
CartManage.java
CartTest.java
DigitalVideoDisc.java
Main.java
Store.java
TestStore.java
Testing.java



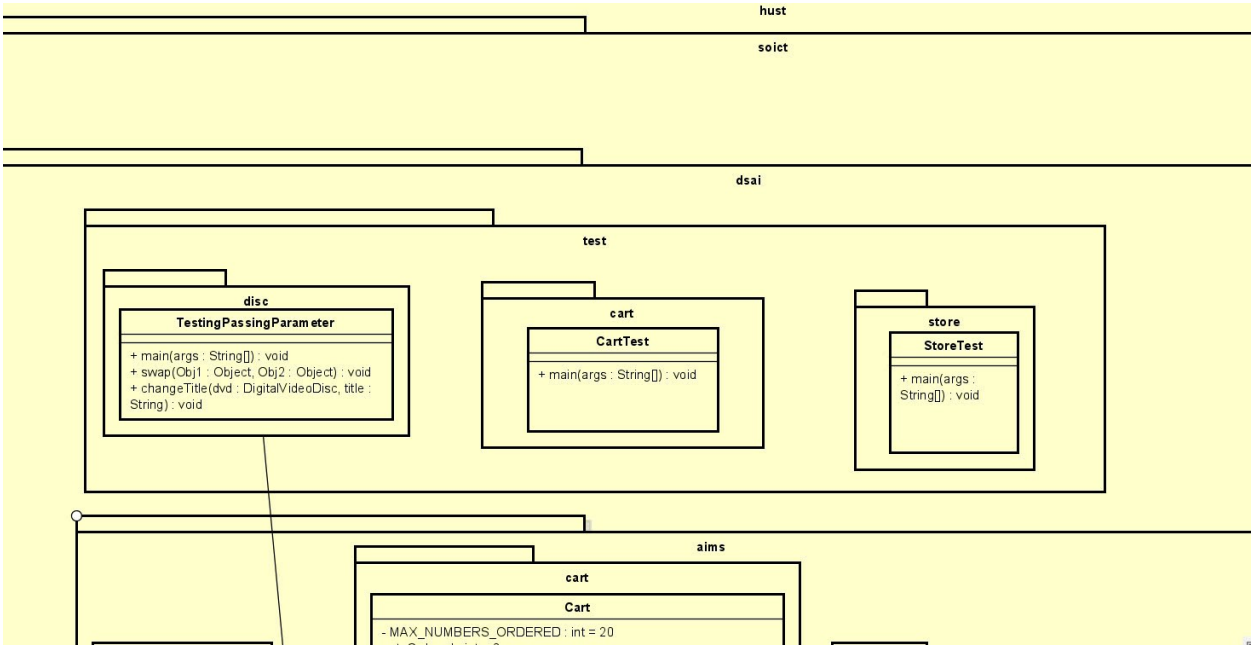
7. String, StringBuilder and StringBuffer

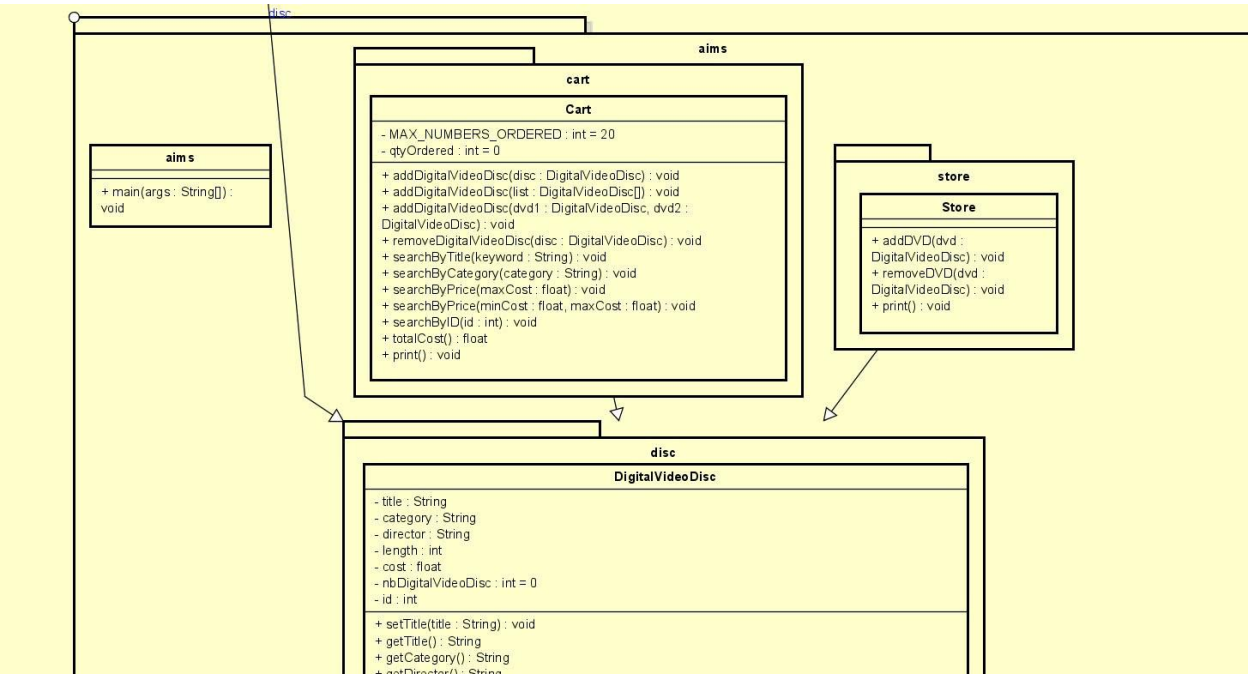
```java
package hust.soict.dsai.garbage;

public class CreateGarbage{
    public static void main(String[] args){
        String gb = "";
        long start = System.CurrentTimeMillis();

        try{
            for(int i = 0; i < Interger.MAX_VALUE; i++){
                gb += i;
            }
        }
        catch(OutOfMemoryError e){
            System.err.println("Outofmemory encounter.");
        }
        long end = System.CurrentTimeMillis();
        System.out.println("Execution time: " + (end - start) + " ms.");
    }
}
```

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

public class NoGarbage{
    public static void main(String[] args){
        StringBuilder Build_Garbage = new StringBuilder();
        long start = System.CurrentTimeMillis();
        for(int i = 0; i < Interger.MAX_VALUE; i++){
            Build_Garbage.append(i);
            if(i%100000 == 0){
                Build_Garbage.length(0);
            }
        }
        long end = System.CurrentTimeMillis();
        System.out.println("Execution time: " + (end - start) + " ms.");
    }
}
```

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

import java.util.Random;
public class LoopsConcatenation{
    public static void main(String[] args){
        Random rd = new Random();
        long start = System.CurrentTimeMillis();
        String str = "";
        for(int i = 0; i < 65536; i++){
            str += rd.nextInt(2);
        }
        System.out.println("Using operator: " + (System.CurrentTimeMillis() - start) + " ms.");
        StringBuffer buff = new StringBuffer();
        for(int i = 0; i < 65536; i++){
            buff.append(rd.nextInt(2));
        }
        str = buff.toString();
        System.out.println("Using StringBuffer: " + (System.CurrentTimeMillis() - start) + " ms.");
        StringBuilder build = new StringBuilder();
        for(int i = 0; i < 65536; i++){
            build.append(rd.nextInt(2));
        }
        str = build.toString();
        System.out.println("Using StringBuilder: " + (System.CurrentTimeMillis() - start) + " ms.");
    }
}
```

8. Class Diagram





9. Use Case Diagram