

```

// Classifier Member & Instance Member
private int id;
private static int nbDigitalVideoDiscs = 0;

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

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

```

```

// Method Overloading
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
    for (int i = 0; i < dvdList.length; i++) {
        if (qtyOrdered >= MAX_NUMBERS_ORDERS) {
            System.out.println("The cart is almost full");
            break;
        }
        itemsOrdered[qtyOrdered++] = dvdList[i];
        System.out.println("The disc " + i + " has been added");
    }
}

// Method Overloading
public void addDigitalVideoDisc(DigitalVideoDisc disc1, DigitalVideoDisc disc2) {
    if (qtyOrdered < MAX_NUMBERS_ORDERS) {
        itemsOrdered[qtyOrdered] = disc1;
        qtyOrdered++;
    }
    if (qtyOrdered < MAX_NUMBERS_ORDERS) {
        itemsOrdered[qtyOrdered] = disc2;
        qtyOrdered++;
    }
}

```

```

// print cart
public void printCart() {
    System.out.println("*****CART*****");
    System.out.println("Ordered Items:");
    for (int i = 0; i < itemsOrdered.length; i++) {
        if(itemsOrdered[i] != null) {
            System.out.println((i + 1) + ". DVD - " + itemsOrdered[i].toString());
        }
    }
    System.out.println("Total cost: " + this.totalCost());
    System.out.println("*****");
}

```

Test package:

```

package test;

import aimsproject.DigitalVideoDisc;
import aimsproject.Carts;

/**
 *
 * @author LENOVO
 */
public class TestPassingParameter {
    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");

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

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

    public static void swap(Object o1, Object o2){
        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);
    }
}

```

TestPassingParameter > main > jungleDVD >

```

    */
    package test;

    import aimsproject.DigitalVideoDisc;
    import aimsproject.Carts;

    /**
     *
     * @author LENOVO
     */
    public class CartTest {

        public static void main(String[] args) {
            Carts cart = new Carts();

            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", 19.95f);
            cart.addDigitalVideoDisc(dvd3);

            cart.printCart();
        }
    }

```

```

    package test;

    import aimsproject.DigitalVideoDisc;
    import aimsproject.Store;

    /**
     * To change this license header, choose License Headers in Project Properties.
     * To change this template file, choose Tools | Templates
     * and open the template in the editor.
     */

    /**
     *
     * @author LENOVO
     */
    public class StoreTest {
        public static void main(String[] args) {
            Store store = new Store();
            DigitalVideoDisc dvd = new DigitalVideoDisc("Test");
            store.addDVD(dvd);
            store.removeDVD(dvd);
        }
    }

```

String, String Builder:

```
* @author LENOVO
*/
import java.util.Random;
public class ConcatenationInLoops {
    public static void main(String[] args) {
        Random r = new Random(123);
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i++) {
            s += r.nextInt(2);
        }

        System.out.println(System.currentTimeMillis() - start);

        r = new Random(132);

        start = System.currentTimeMillis();

        StringBuilder sb = new StringBuilder();

        for (int i = 0; i < 65536; i++) {
            sb.append(r.nextInt(2));
        }
        s = sb.toString();
        System.out.println(System.currentTimeMillis() - start);
    }
}
```

```

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;

public class GarbageCreator {
    public static void main(String[] args) {
        String filePath = "C:\\Users\\LENOVO\\Desktop\\OOP\\LAB03_20225653\\OtherProjects\\hust\\soict\\←
garbage\\text.txt";
        long startTime, endTime;
        try {
            startTime = System.currentTimeMillis();
            List<String> lines = Files.readAllLines(Paths.get(filePath));

            String outputFile = "";
            for (String line : lines) {
                outputFile += line;
            }

            endTime = System.currentTimeMillis();

            System.out.println(endTime - startTime);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

```

import java.io.*;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;

public class NoGarbage {
    public static void main(String[] args) {
        String filePath = "C:\\Users\\LENOVO\\Desktop\\OOP\\LAB03_20225653\\OtherProjects\\hust\\soict\\←
garbage\\text.txt";
        long startTime, endTime;
        try {
            startTime = System.currentTimeMillis();
            List<String> lines = Files.readAllLines(Paths.get(filePath));

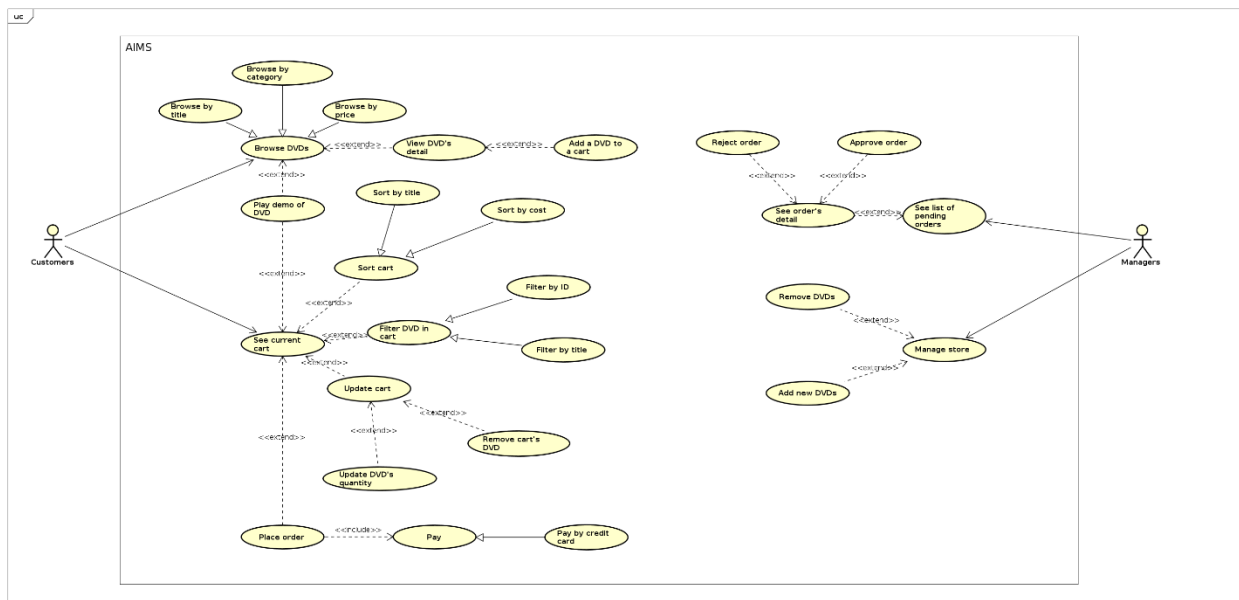
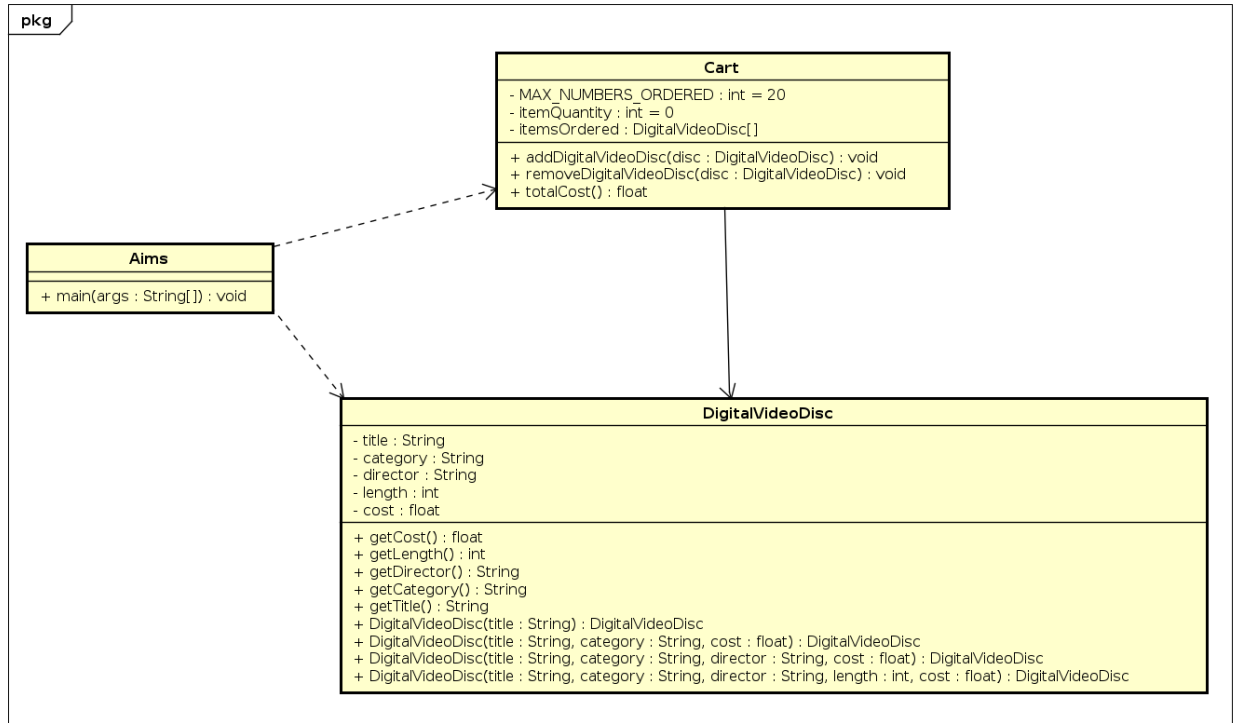
            StringBuilder outputFile = new StringBuilder();
            for (String line : lines) {
                outputFile.append(line);
            }

            endTime = System.currentTimeMillis();

            System.out.println(endTime - startTime);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

use-case diagram and class diagram:



Debug:

295.0 / 338.5MB

Source History

```

15 public static void main(String[] args) {
16     DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
17     DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
18
19     swap(jungleDVD, cinderellaDVD);
20     System.out.println("jungle dvd title: " + jungleDVD.getTitle());
21     System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
22
23     changeTitle(jungleDVD, cinderellaDVD.getTitle());
24     System.out.println("jungle dvd title: " + jungleDVD.getTitle());
25 }
26
27 public static void swap(Object o1, Object o2){
28     Object tmp = o1;
29     o1 = o2;
30     o2 = tmp;
31 }
32
33 public static void changeTitle(DigitalVideoDisc dvd, String title) {
34     String oldTitle = dvd.getTitle();
35     dvd.setTitle(title);
36     dvd = new DigitalVideoDisc(oldTitle);
37 }

```

test.TestPassingParameter > swap

Name	Type	Value
<Enter new watch>		
Static		
o1	DigitalVideoDisc	#95
o2	DigitalVideoDisc	#94
tmp	DigitalVideoDisc	#94

Source History

```

15 public static void main(String[] args) {
16     DigitalVideoDisc jungleDVD = new DigitalVideoDisc("Jungle");
17     DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc("Cinderella");
18
19     swap(jungleDVD, cinderellaDVD);
20     System.out.println("jungle dvd title: " + jungleDVD.getTitle());
21     System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
22
23     changeTitle(jungleDVD, cinderellaDVD.getTitle());
24     System.out.println("jungle dvd title: " + jungleDVD.getTitle());
25 }
26
27 public static void swap(Object o1, Object o2){
28     Object tmp = o1;
29     o1 = o2;
30     o2 = tmp;
31 }
32
33 public static void changeTitle(DigitalVideoDisc dvd, String title) {
34     String oldTitle = dvd.getTitle();
35     dvd.setTitle(title);
36     dvd = new DigitalVideoDisc(oldTitle);
37 }

```

test.TestPassingParameter > main

Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#93(length=0)
jungleDVD	DigitalVideoDisc	#94
cinderellaDVD	DigitalVideoDisc	#95

AimsProject (debug) | running... | 20:1 | IN

Answer the questions:

1. After the call of `swap(jungleDVD, cinderellaDVD)` why does the title of these two objects still remain?

Nguyên nhân:

Trong Java, tham số được truyền bằng giá trị (pass-by-value), ngay cả khi truyền đối tượng.

Khi gọi `swap(jungleDVD, cinderellaDVD)`, giá trị của tham chiếu đến hai đối tượng được sao chép vào các tham số `o1` và `o2`.

Phép gán `o1 = o2`; và `o2 = tmp`; chỉ thay đổi giá trị của tham chiếu cục bộ `o1` và `o2` bên trong phương thức `swap`.

Vì vậy, sau khi `swap`, các đối tượng gốc vẫn giữ nguyên giá trị của chúng.

2. After the call of `changeTitle(jungleDVD, cinderellaDVD.getTitle())` why is the title of the `JungleDVD` changed?

Nguyên nhân:

Khi gọi `changeTitle`, tham chiếu đến đối tượng `jungleDVD` được sao chép vào tham số `dvd`.

Trong dòng `dvd.setTitle(title)`, phương thức `setTitle` thay đổi giá trị thuộc tính `title` của đối tượng mà `dvd` tham chiếu, tức là đối tượng `jungleDVD`.

Dòng `dvd = new DigitalVideoDisc(oldTitle)`; chỉ thay đổi tham chiếu cục bộ `dvd` để trỏ đến một đối tượng mới. Việc này không ảnh hưởng đến đối tượng gốc `jungleDVD` trong hàm `main`.

Do đó, tiêu đề của `jungleDVD` được cập nhật bởi `setTitle(title)` và vẫn giữ nguyên sau khi thoát khỏi phương thức.