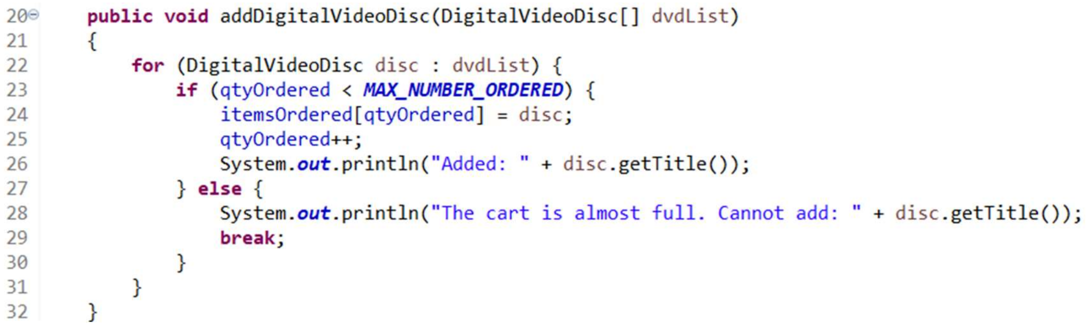
Họ và tên: Trần Ngọc Hưng – 20225635

Mã lớp: 744523

Báo cáo OOP Lab 03

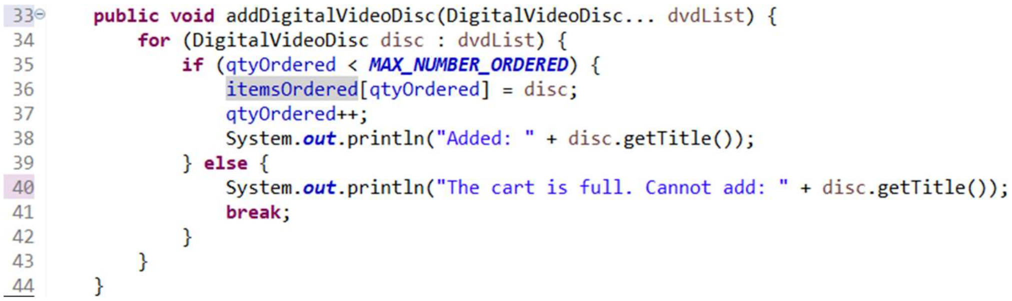
1. New written code

- **addDigitalVideoDisc(DigitalVideoDisc [] dvdList):** a new method that has the same name but with different type of parameter



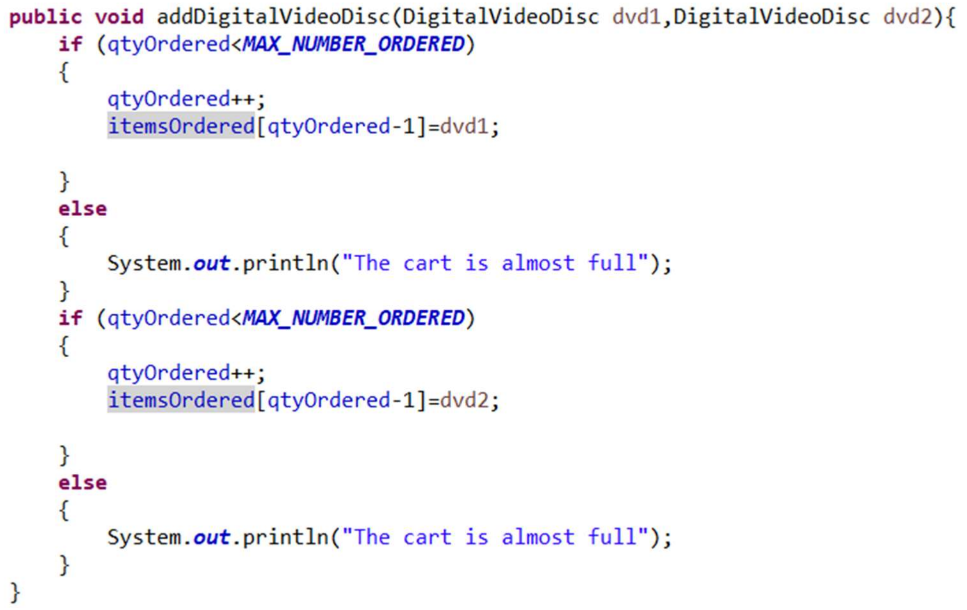
```java
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList)
{
    for (DigitalVideoDisc disc : dvdList) {
        if (qtyOrdered < MAX_NUMBER_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("Added: " + disc.getTitle());
        } else {
            System.out.println("The cart is almost full. Cannot add: " + disc.getTitle());
            break;
        }
    }
}
```

- addDigitalVideoDisc(DigitalVideoDisc... dvdList): a method **addDigitalVideoDisc** which allows to pass an arbitrary number of arguments for dvd



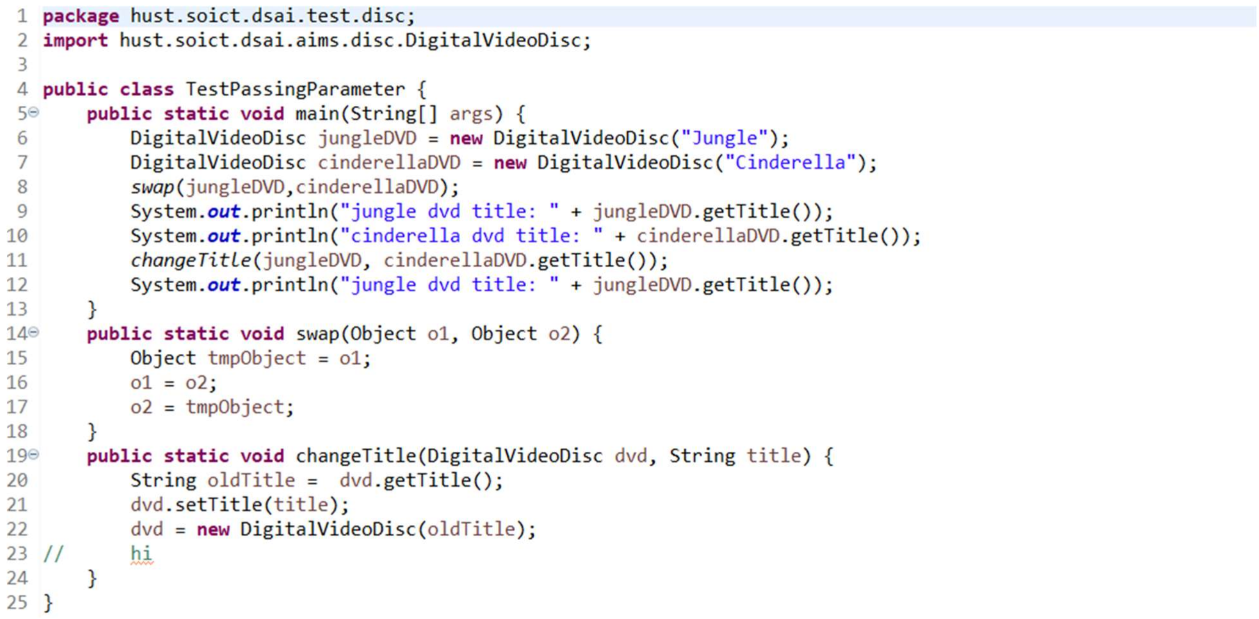
```java
public void addDigitalVideoDisc(DigitalVideoDisc... dvdList) {
    for (DigitalVideoDisc disc : dvdList) {
        if (qtyOrdered < MAX_NUMBER_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("Added: " + disc.getTitle());
        } else {
            System.out.println("The cart is full. Cannot add: " + disc.getTitle());
            break;
        }
    }
}
```

- addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2):



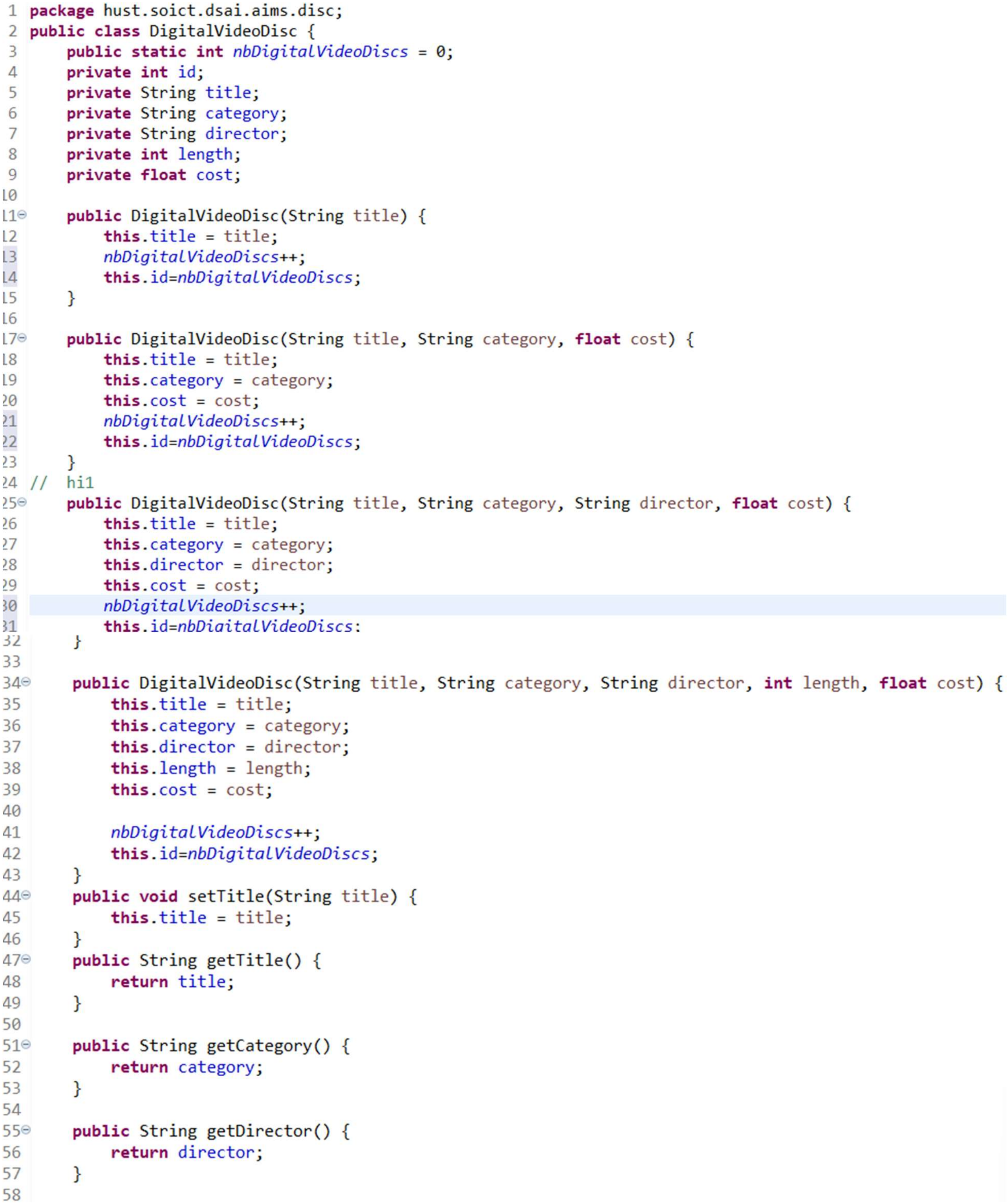
```
public void addDigitalVideoDisc(DigitalVideoDisc dvd1,DigitalVideoDisc dvd2){
    if (qtyOrdered<MAX_NUMBER_ORDERED)
    {
        qtyOrdered++;
        itemsOrdered[qtyOrdered-1]=dvd1;

    }
    else
    {
        System.out.println("The cart is almost full");
    }
    if (qtyOrdered<MAX_NUMBER_ORDERED)
    {
        qtyOrdered++;
        itemsOrdered[qtyOrdered-1]=dvd2;

    }
    else
    {
        System.out.println("The cart is almost full");
    }
}
```

- TestPassingParameter.java:

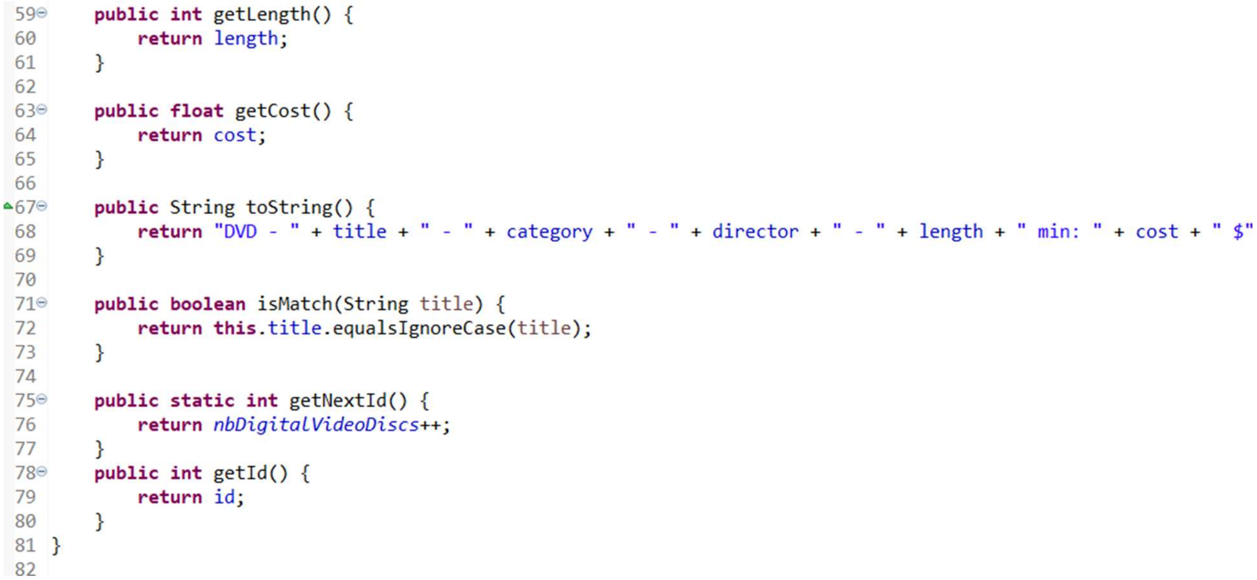


```
package hust.soict.dsai.test.disc;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

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 tmpObject = o1;
        o1 = o2;
        o2 = tmpObject;
    }
    public static void changeTitle(DigitalVideoDisc dvd, String title) {
        String oldTitle =  dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
//      hi
    }
}
```

- Keep track of the number of DVDs created:



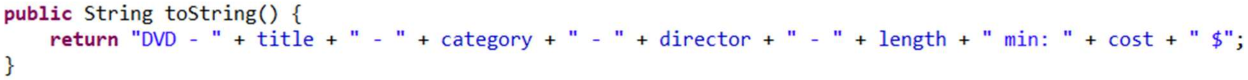
```java
package hust.soict.dsai.aims.disc;
public class DigitalVideoDisc {
    public static int nbDigitalVideoDiscs = 0;
    private int id;
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;

    public DigitalVideoDisc(String title) {
        this.title = title;
        nbDigitalVideoDiscs++;
        this.id=nbDigitalVideoDiscs;
    }

    public DigitalVideoDisc(String title, String category, float cost) {
        this.title = title;
        this.category = category;
        this.cost = cost;
        nbDigitalVideoDiscs++;
        this.id=nbDigitalVideoDiscs;
    }
//  hi1
    public DigitalVideoDisc(String title, String category, String director, float cost) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.cost = cost;
        nbDigitalVideoDiscs++;
        this.id=nbDiaitalVideoDiscs:
    }

    public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;

        nbDigitalVideoDiscs++;
        this.id=nbDigitalVideoDiscs;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getTitle() {
        return title;
    }

    public String getCategory() {
        return category;
    }

    public String getDirector() {
        return director;
    }

```



```
    public int getLength() {
        return length;
    }

    public float getCost() {
        return cost;
    }

    public String toString() {
        return "DVD - " + title + " - " + category + " - " + director + " - " + length + " min: " + cost + " $"
    }

    public boolean isMatch(String title) {
        return this.title.equalsIgnoreCase(title);
    }

    public static int getNextId() {
        return nbDigitalVideoDiscs++;
    }
    public int getId() {
        return id;
    }
}
```

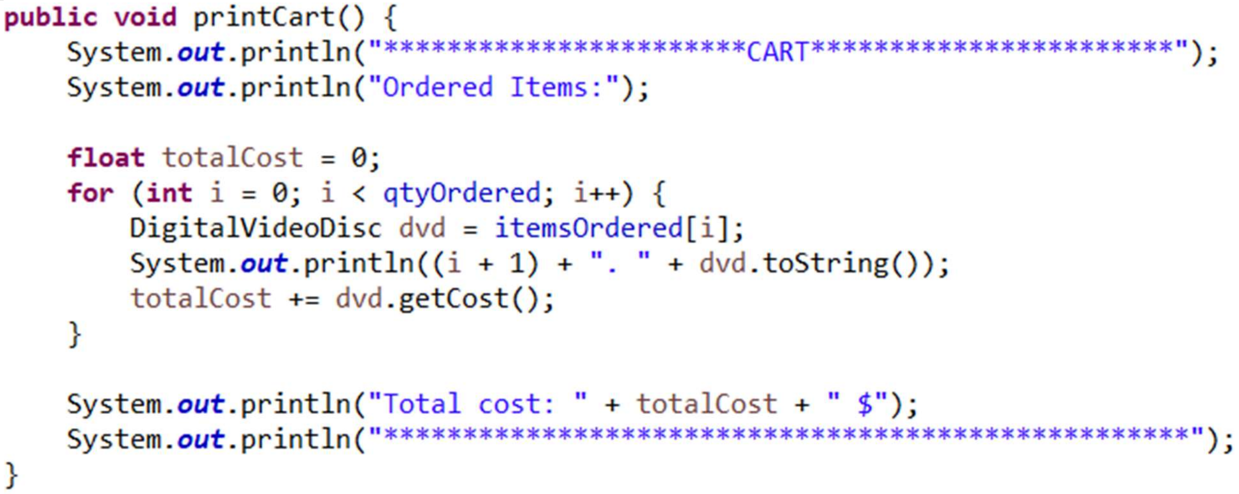
- Method to print the list of ordered items of a cart, the price of each item, and the total price:

toString() method:



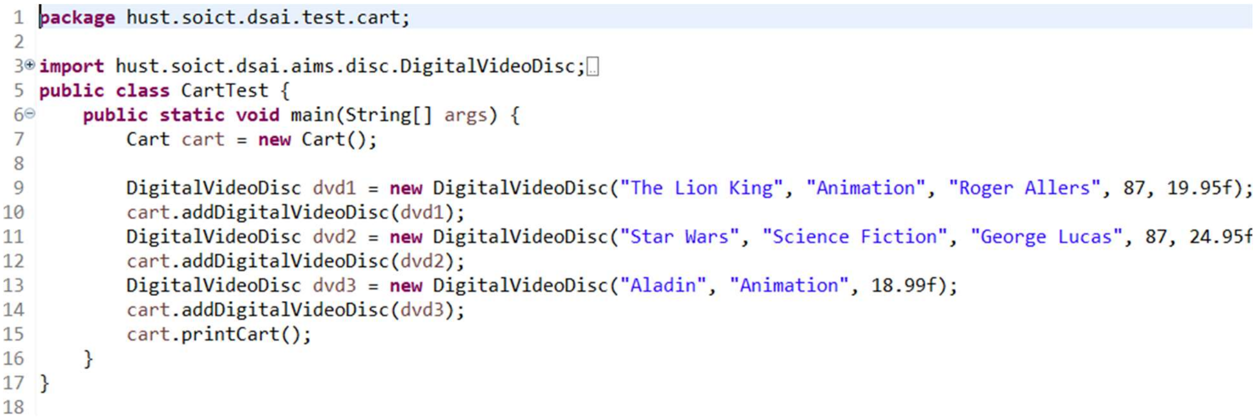
```
public String toString() {
    return "DVD - " + title + " - " + category + " - " + director + " - " + length + " min: " + cost + " $";
}
```

printCart() method:



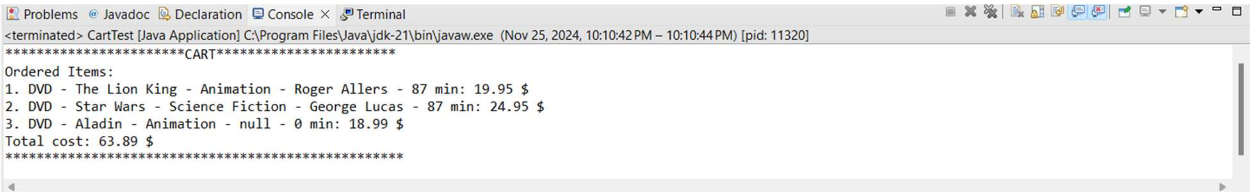
```
public void printCart() {
    System.out.println("***********************CART***********************");
    System.out.println("Ordered Items:");

    float totalCost = 0;
    for (int i = 0; i < qtyOrdered; i++) {
        DigitalVideoDisc dvd = itemsOrdered[i];
        System.out.println((i + 1) + ". " + dvd.toString());
        totalCost += dvd.getCost();
    }

    System.out.println("Total cost: " + totalCost + " $");
    System.out.println("***************************************************");
}
```

CartTest.java



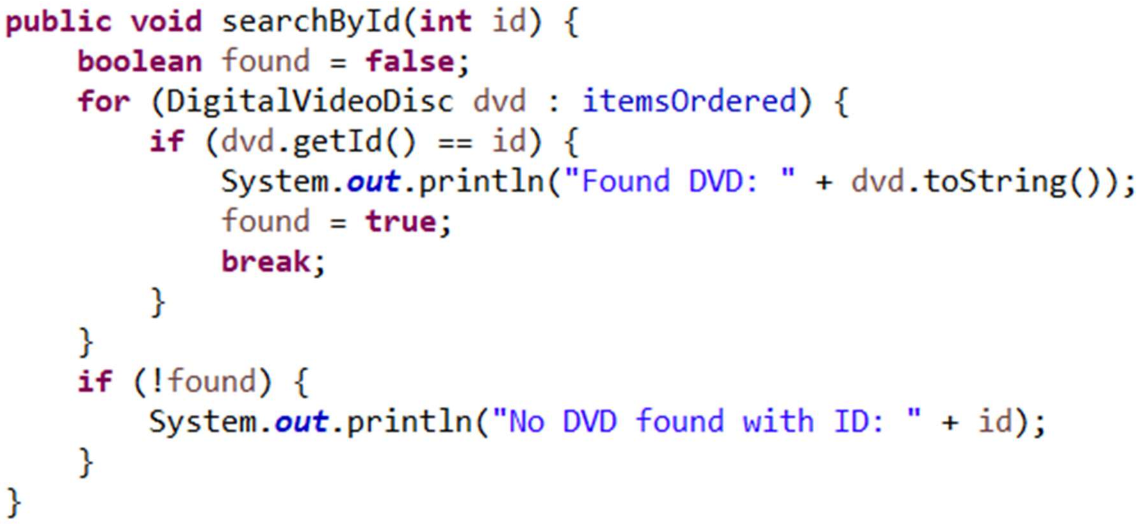
```java
package hust.soict.dsai.test.cart;

import hust.soict.dsai.aims.disc.DigitalVideoDisc;
public class CartTest {
    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", 18.99f);
        cart.addDigitalVideoDisc(dvd3);
        cart.printCart();
    }
}

```

Run CartTest.java:

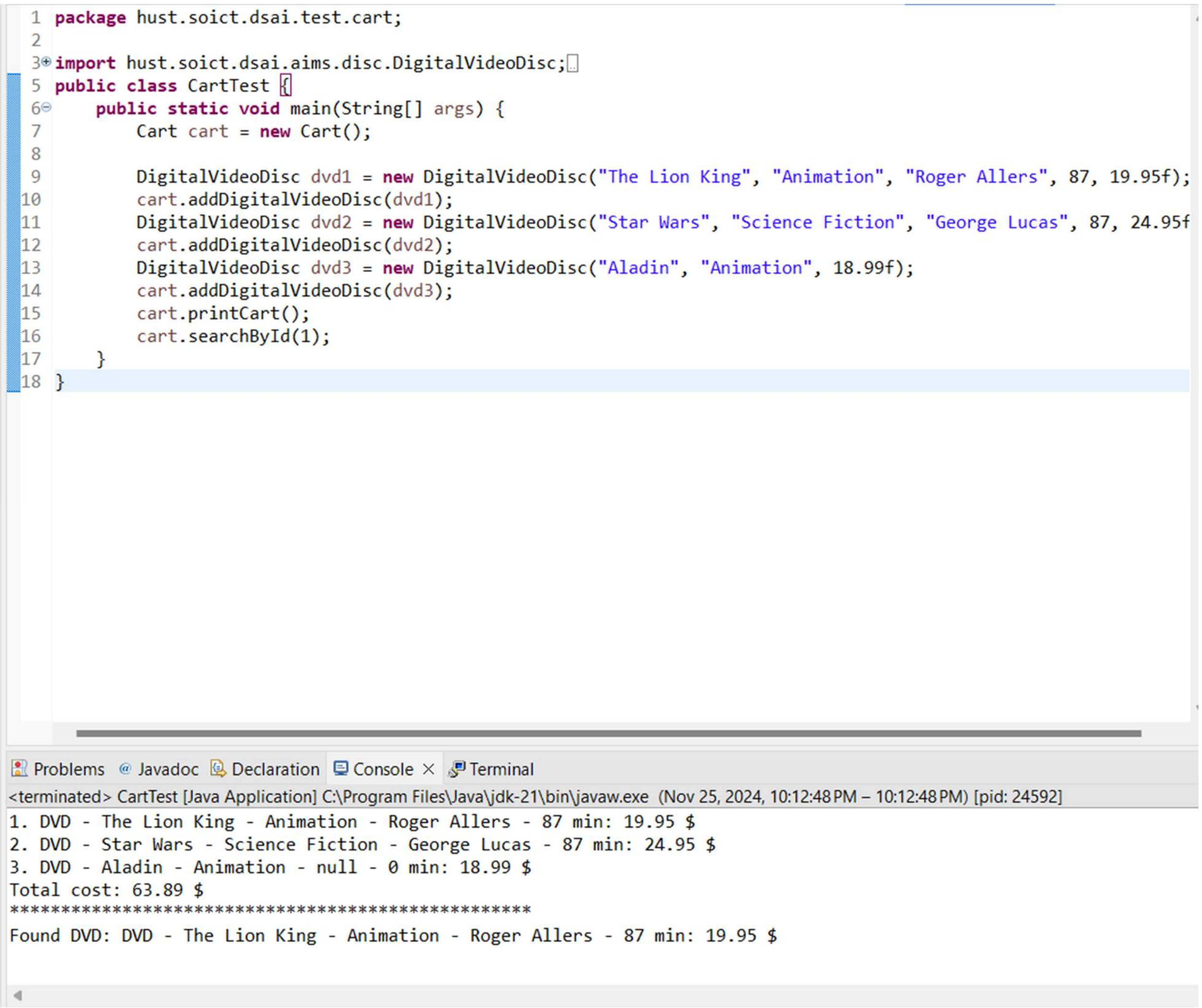


```
<terminated> CartTest [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 25, 2024, 10:10:42 PM – 10:10:44 PM) [pid: 11320]
**********************CART***********************
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers - 87 min: 19.95 $
2. DVD - Star Wars - Science Fiction - George Lucas - 87 min: 24.95 $
3. DVD - Aladin - Animation - null - 0 min: 18.99 $
Total cost: 63.89 $
**************************************************
```

- Search for DVDs in the cart by ID and display the search results:



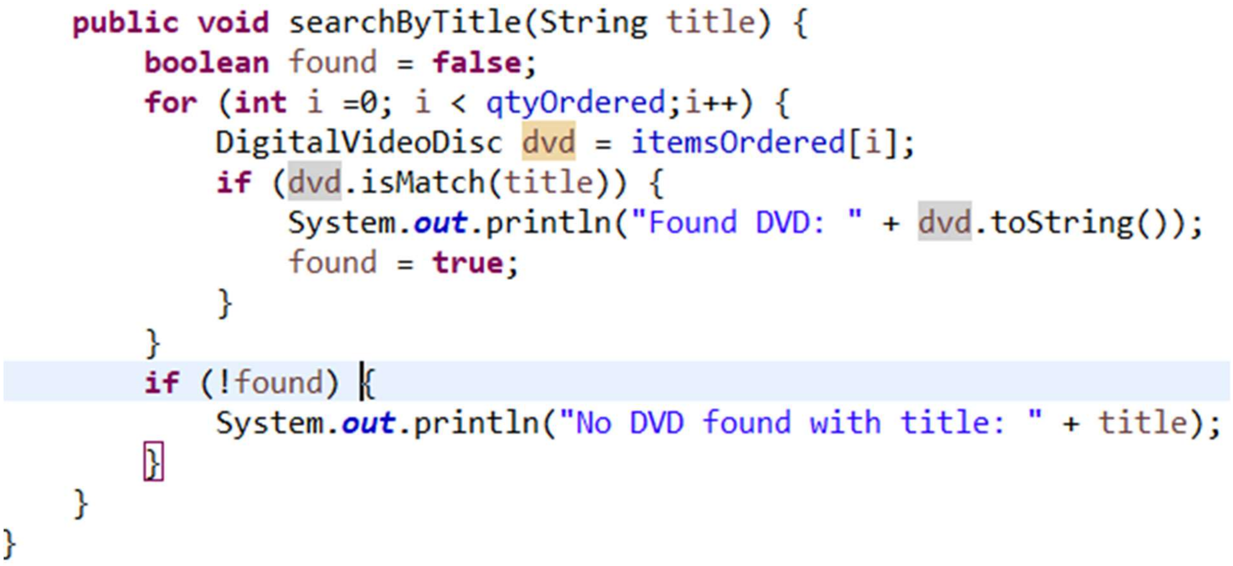
```java
public void searchById(int id) {
    boolean found = false;
    for (DigitalVideoDisc dvd : itemsOrdered) {
        if (dvd.getId() == id) {
            System.out.println("Found DVD: " + dvd.toString());
            found = true;
            break;
        }
    }
    if (!found) {
        System.out.println("No DVD found with ID: " + id);
    }
}
```



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

import hust.soict.dsai.aims.disc.DigitalVideoDisc;□
public class CartTest {
    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", 18.99f);
        cart.addDigitalVideoDisc(dvd3);
        cart.printCart();
        cart.searchById(1);
    }
}
```

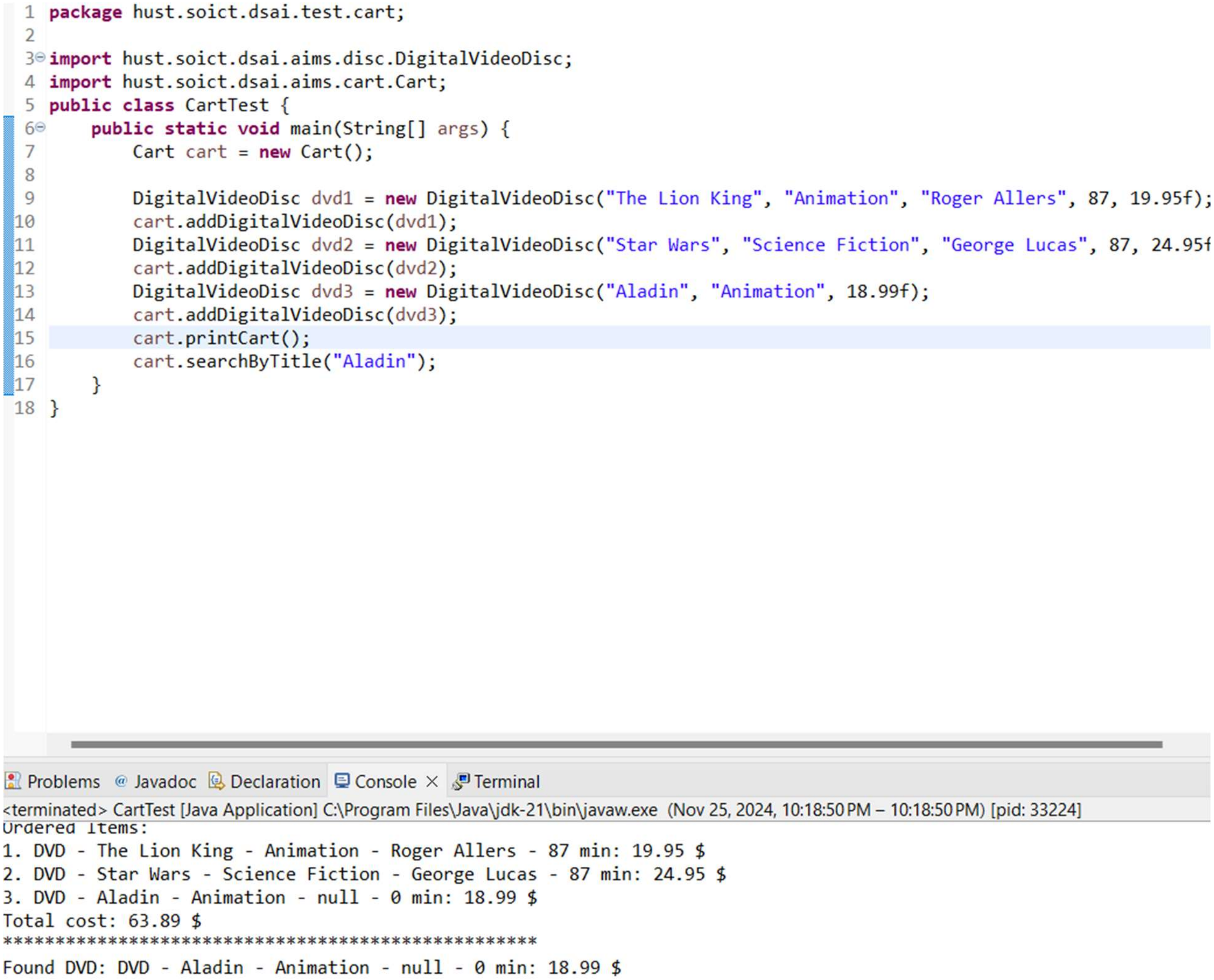
```
<terminated> CartTest [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 25, 2024, 10:12:48 PM – 10:12:48 PM) [pid: 24592]
1. DVD - The Lion King - Animation - Roger Allers - 87 min: 19.95 $
2. DVD - Star Wars - Science Fiction - George Lucas - 87 min: 24.95 $
3. DVD - Aladin - Animation - null - 0 min: 18.99 $
Total cost: 63.89 $
***************************************************
Found DVD: DVD - The Lion King - Animation - Roger Allers - 87 min: 19.95 $
```

- Search for DVDs in the cart by title and print the results:



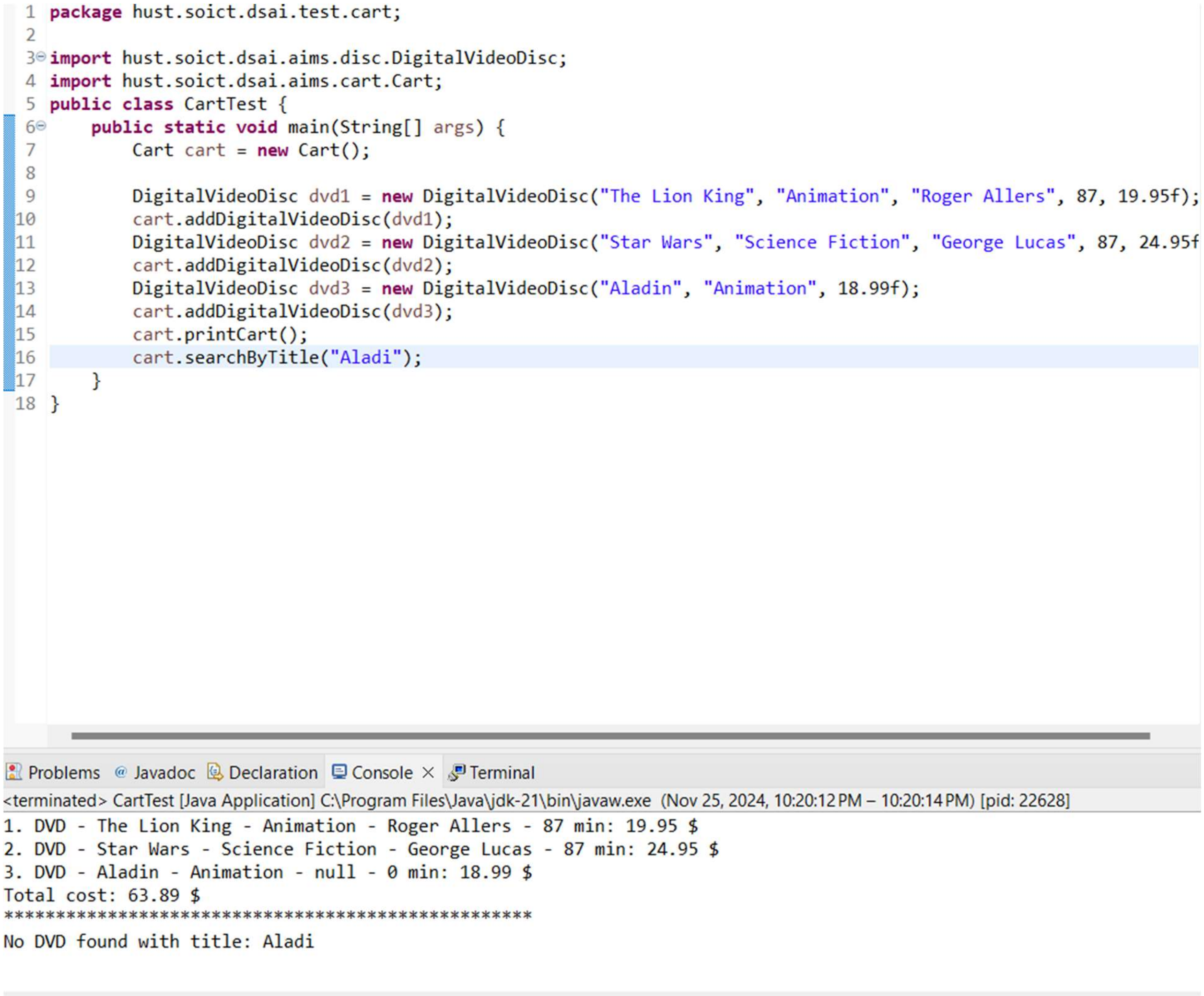
```java
    public void searchByTitle(String title) {
        boolean found = false;
        for (int i =0; i < qtyOrdered;i++) {
            DigitalVideoDisc dvd = itemsOrdered[i];
            if (dvd.isMatch(title)) {
                System.out.println("Found DVD: " + dvd.toString());
                found = true;
            }
        }
        if (!found) {
            System.out.println("No DVD found with title: " + title);
        }
    }
}
```

Result:



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

import hust.soict.dsai.aims.disc.DigitalVideoDisc;
import hust.soict.dsai.aims.cart.Cart;
public class CartTest {
    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", 18.99f);
        cart.addDigitalVideoDisc(dvd3);
        cart.printCart();
        cart.searchByTitle("Aladin");
    }
}
```

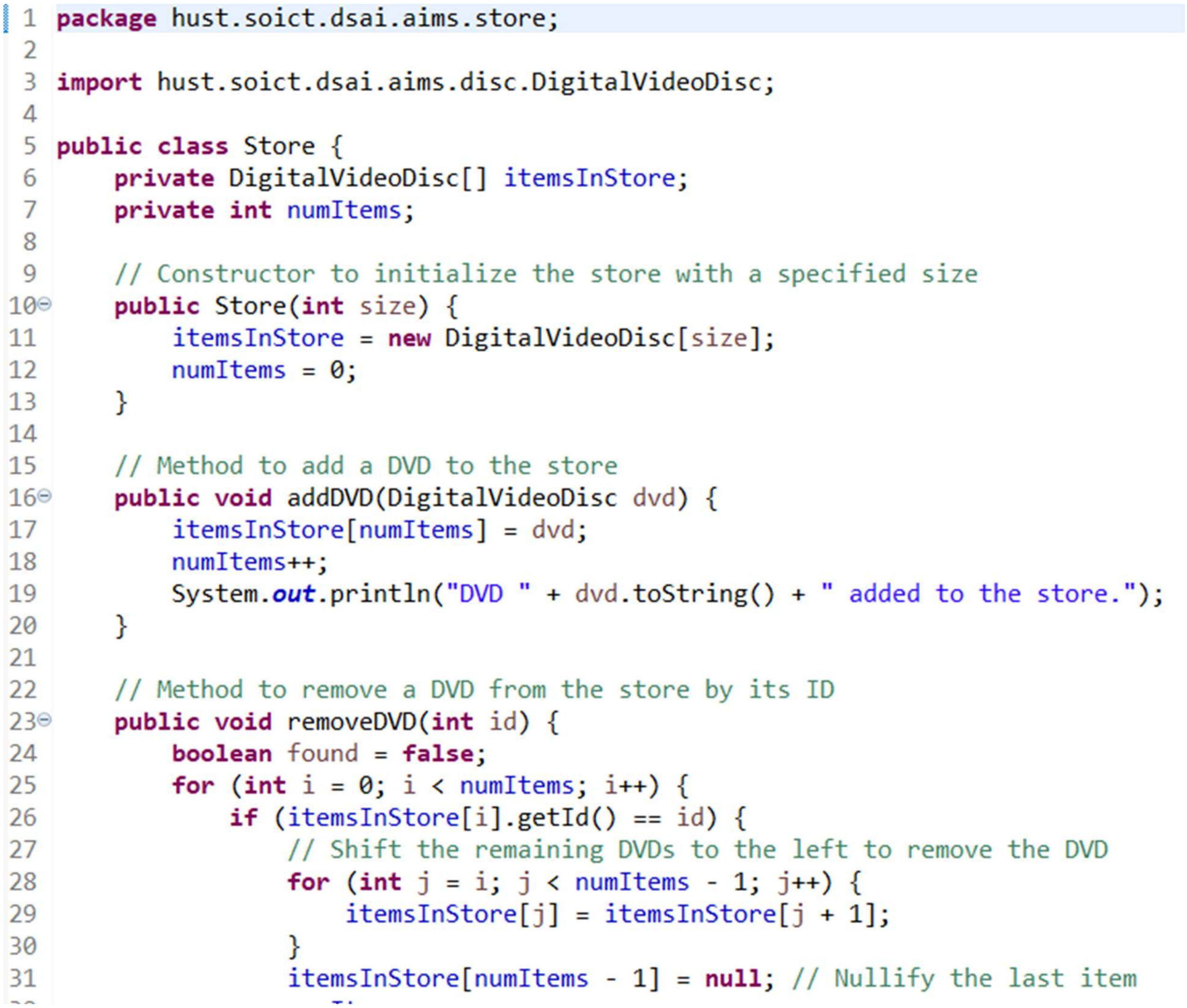
```
<terminated> CartTest [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 25, 2024, 10:18:50 PM – 10:18:50 PM) [pid: 33224]
Ordered Items:
1. DVD - The Lion King - Animation - Roger Allers - 87 min: 19.95 $
2. DVD - Star Wars - Science Fiction - George Lucas - 87 min: 24.95 $
3. DVD - Aladin - Animation - null - 0 min: 18.99 $
Total cost: 63.89 $
***************************************************
Found DVD: DVD - Aladin - Animation - null - 0 min: 18.99 $
```



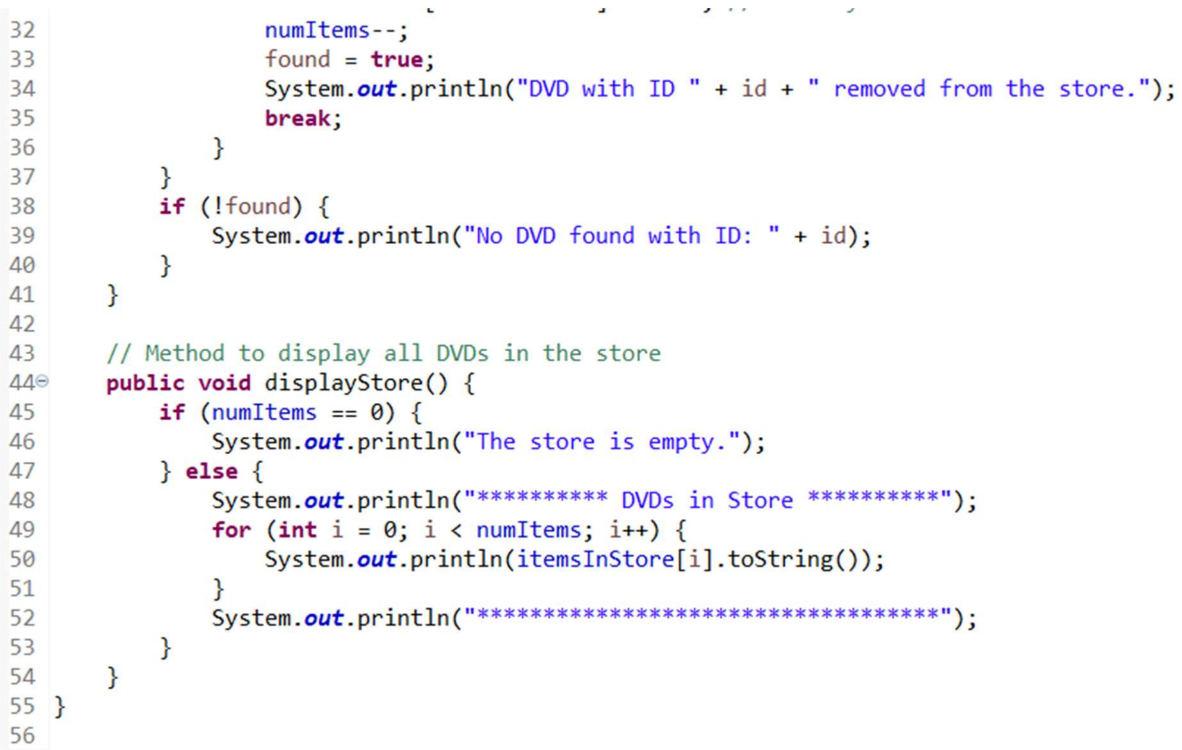
```
package hust.soict.dsai.test.cart;

import hust.soict.dsai.aims.disc.DigitalVideoDisc;
import hust.soict.dsai.aims.cart.Cart;
public class CartTest {
    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", 18.99f);
        cart.addDigitalVideoDisc(dvd3);
        cart.printCart();
        cart.searchByTitle("Aladi");
    }
}
```

```
<terminated> CartTest [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 25, 2024, 10:20:12 PM – 10:20:14 PM) [pid: 22628]
1. DVD - The Lion King - Animation - Roger Allers - 87 min: 19.95 $
2. DVD - Star Wars - Science Fiction - George Lucas - 87 min: 24.95 $
3. DVD - Aladin - Animation - null - 0 min: 18.99 $
Total cost: 63.89 $
***************************************************
No DVD found with title: Aladi
```

- Store.java:

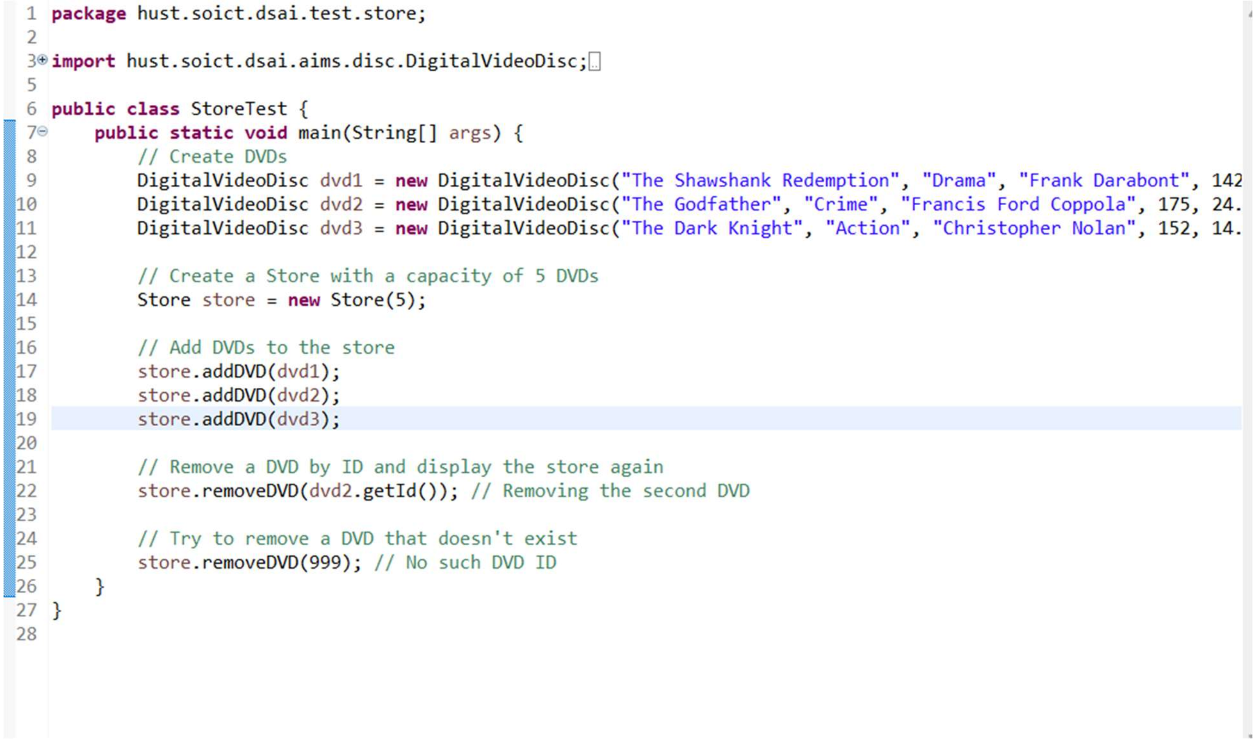


```java
package hust.soict.dsai.aims.store;

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

public class Store {
    private DigitalVideoDisc[] itemsInStore;
    private int numItems;

    // Constructor to initialize the store with a specified size
    public Store(int size) {
        itemsInStore = new DigitalVideoDisc[size];
        numItems = 0;
    }

    // Method to add a DVD to the store
    public void addDVD(DigitalVideoDisc dvd) {
        itemsInStore[numItems] = dvd;
        numItems++;
        System.out.println("DVD " + dvd.toString() + " added to the store.");
    }

    // Method to remove a DVD from the store by its ID
    public void removeDVD(int id) {
        boolean found = false;
        for (int i = 0; i < numItems; i++) {
            if (itemsInStore[i].getId() == id) {
                // Shift the remaining DVDs to the left to remove the DVD
                for (int j = i; j < numItems - 1; j++) {
                    itemsInStore[j] = itemsInStore[j + 1];
                }
                itemsInStore[numItems - 1] = null; // Nullify the last item
```



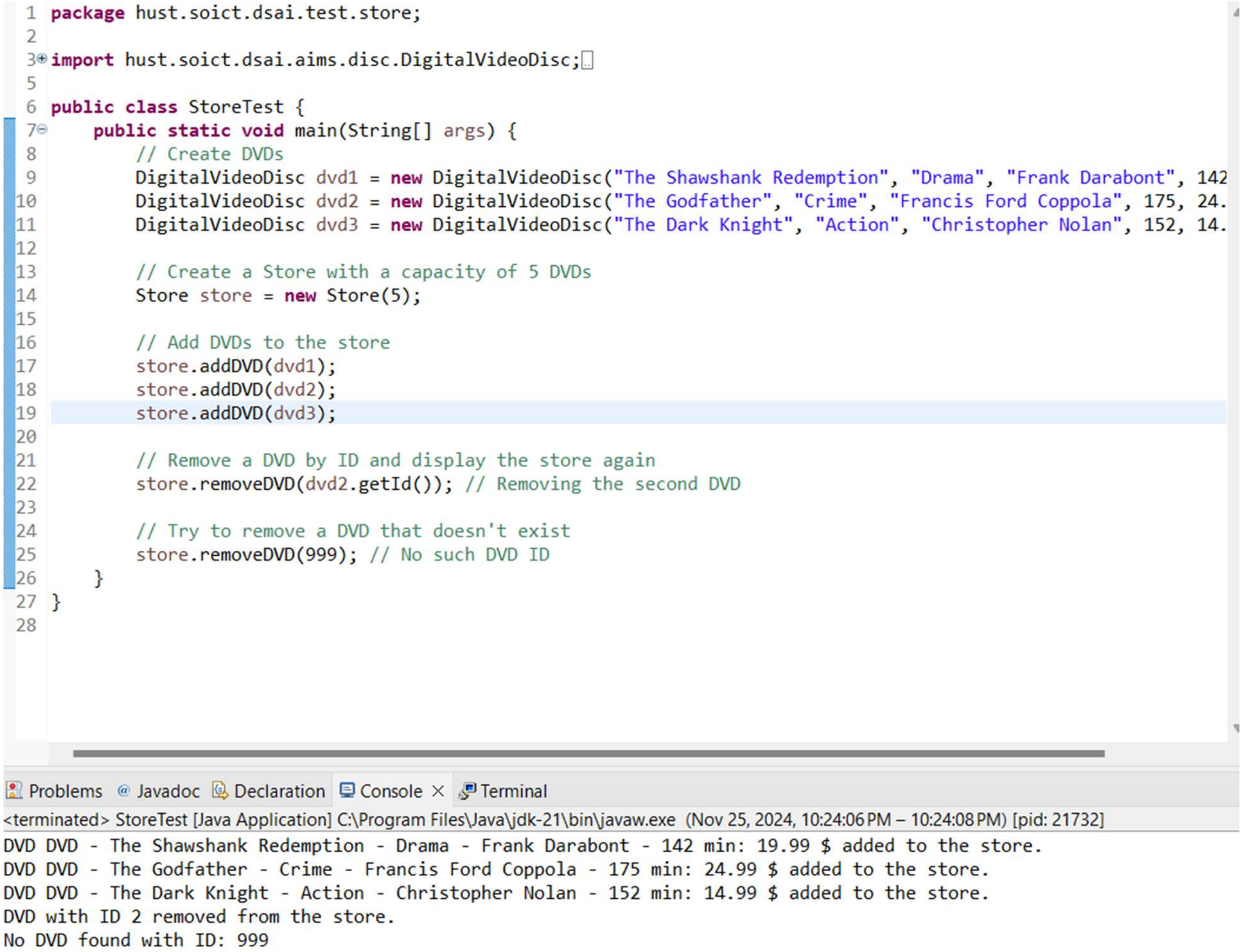
```
                numItems--;
                found = true;
                System.out.println("DVD with ID " + id + " removed from the store.");
                break;
            }
        }
        if (!found) {
            System.out.println("No DVD found with ID: " + id);
        }
    }

    // Method to display all DVDs in the store
    public void displayStore() {
        if (numItems == 0) {
            System.out.println("The store is empty.");
        } else {
            System.out.println("********** DVDs in Store **********");
            for (int i = 0; i < numItems; i++) {
                System.out.println(itemsInStore[i].toString());
            }
            System.out.println("***********************************");
        }
    }
}
```

- StoreTest.java:



```
package hust.soict.dsai.test.store;

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

public class StoreTest {
    public static void main(String[] args) {
        // Create DVDs
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Shawshank Redemption", "Drama", "Frank Darabont", 142
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("The Godfather", "Crime", "Francis Ford Coppola", 175, 24.
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("The Dark Knight", "Action", "Christopher Nolan", 152, 14.

        // Create a Store with a capacity of 5 DVDs
        Store store = new Store(5);

        // Add DVDs to the store
        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);

        // Remove a DVD by ID and display the store again
        store.removeDVD(dvd2.getId()); // Removing the second DVD

        // Try to remove a DVD that doesn't exist
        store.removeDVD(999); // No such DVD ID
    }
}
```

Result:

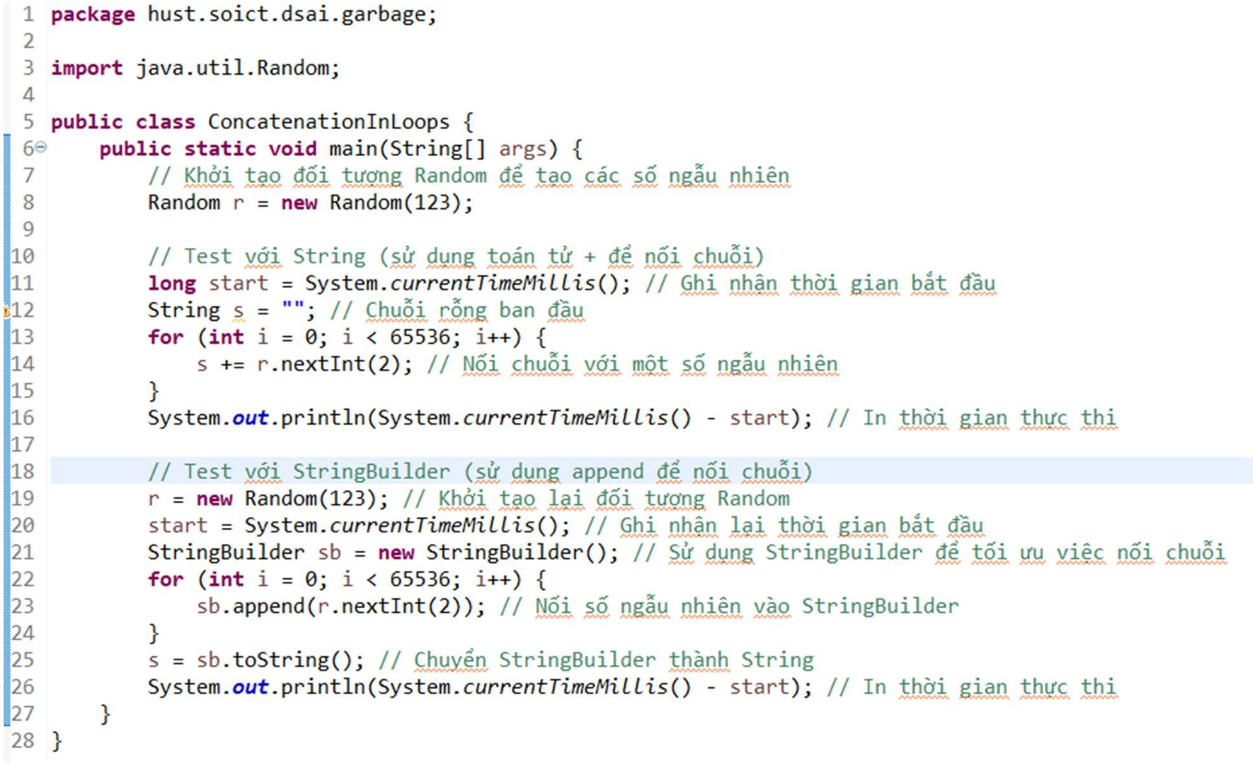


```java
package hust.soict.dsai.test.store;

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

public class StoreTest {
    public static void main(String[] args) {
        // Create DVDs
        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Shawshank Redemption", "Drama", "Frank Darabont", 142
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("The Godfather", "Crime", "Francis Ford Coppola", 175, 24.
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("The Dark Knight", "Action", "Christopher Nolan", 152, 14.

        // Create a Store with a capacity of 5 DVDs
        Store store = new Store(5);

        // Add DVDs to the store
        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);

        // Remove a DVD by ID and display the store again
        store.removeDVD(dvd2.getId()); // Removing the second DVD

        // Try to remove a DVD that doesn't exist
        store.removeDVD(999); // No such DVD ID
    }
}
```

```
<terminated> StoreTest [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 25, 2024, 10:24:06 PM – 10:24:08 PM) [pid: 21732]
DVD DVD - The Shawshank Redemption - Drama - Frank Darabont - 142 min: 19.99 $ added to the store.
DVD DVD - The Godfather - Crime - Francis Ford Coppola - 175 min: 24.99 $ added to the store.
DVD DVD - The Dark Knight - Action - Christopher Nolan - 152 min: 14.99 $ added to the store.
DVD with ID 2 removed from the store.
No DVD found with ID: 999
```

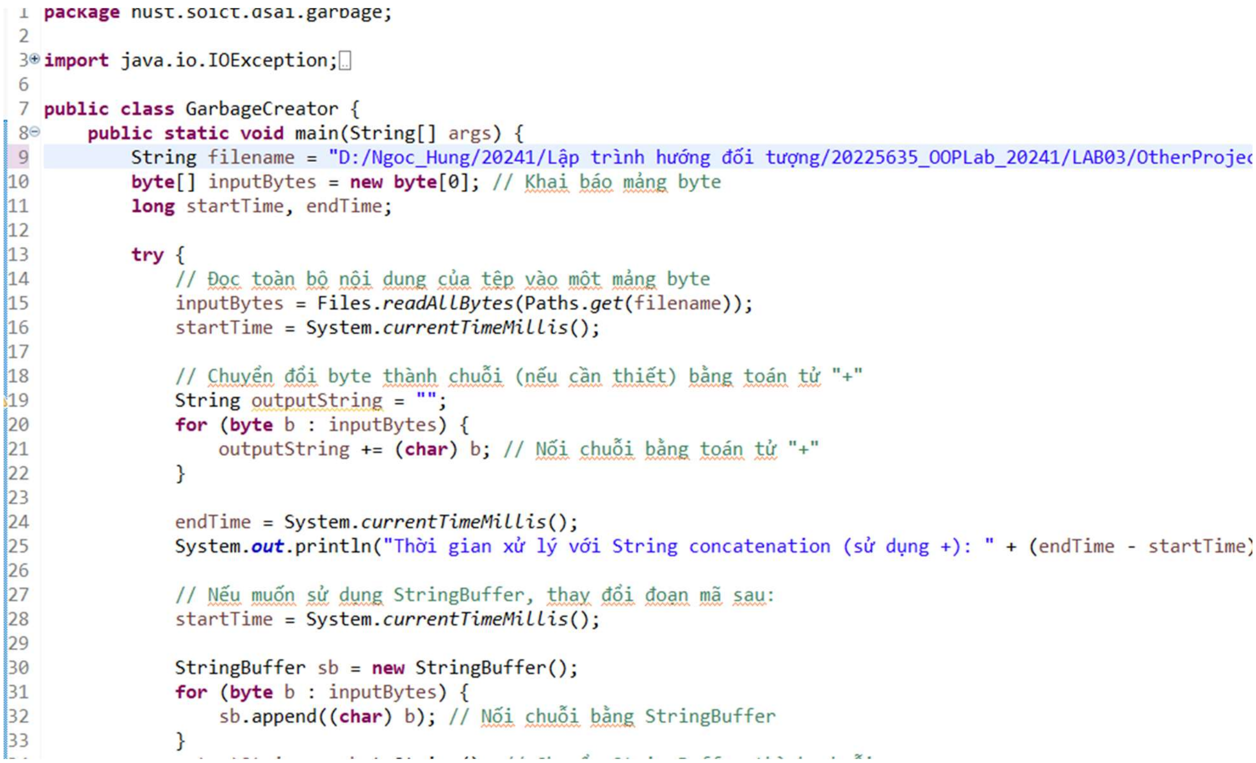
- String, StringBuilder and StringBuffer:

ConcatenationInLoops.java:

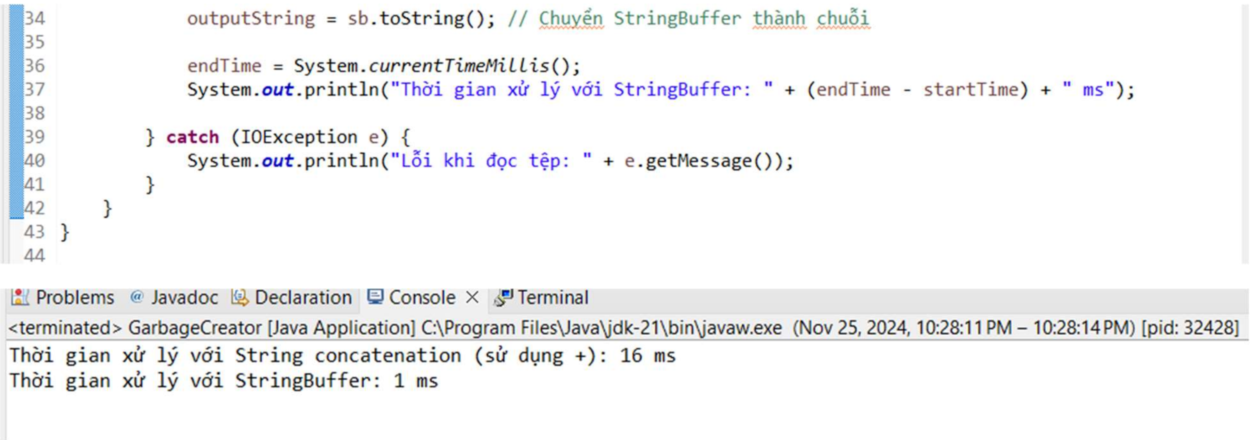


```java
package hust.soict.dsai.garbage;

import java.util.Random;

public class ConcatenationInLoops {
    public static void main(String[] args) {
        // Khởi tạo đối tượng Random để tạo các số ngẫu nhiên
        Random r = new Random(123);

        // Test với String (sử dụng toán tử + để nối chuỗi)
        long start = System.currentTimeMillis(); // Ghi nhận thời gian bắt đầu
        String s = ""; // Chuỗi rỗng ban đầu
        for (int i = 0; i < 65536; i++) {
            s += r.nextInt(2); // Nối chuỗi với một số ngẫu nhiên
        }
        System.out.println(System.currentTimeMillis() - start); // In thời gian thực thi

        // Test với StringBuilder (sử dụng append để nối chuỗi)
        r = new Random(123); // Khởi tạo lại đối tượng Random
        start = System.currentTimeMillis(); // Ghi nhận lại thời gian bắt đầu
        StringBuilder sb = new StringBuilder(); // Sử dụng StringBuilder để tối ưu việc nối chuỗi
        for (int i = 0; i < 65536; i++) {
            sb.append(r.nextInt(2)); // Nối số ngẫu nhiên vào StringBuilder
        }
        s = sb.toString(); // Chuyển StringBuilder thành String
        System.out.println(System.currentTimeMillis() - start); // In thời gian thực thi
    }
}
```

GarbageCreator.java:



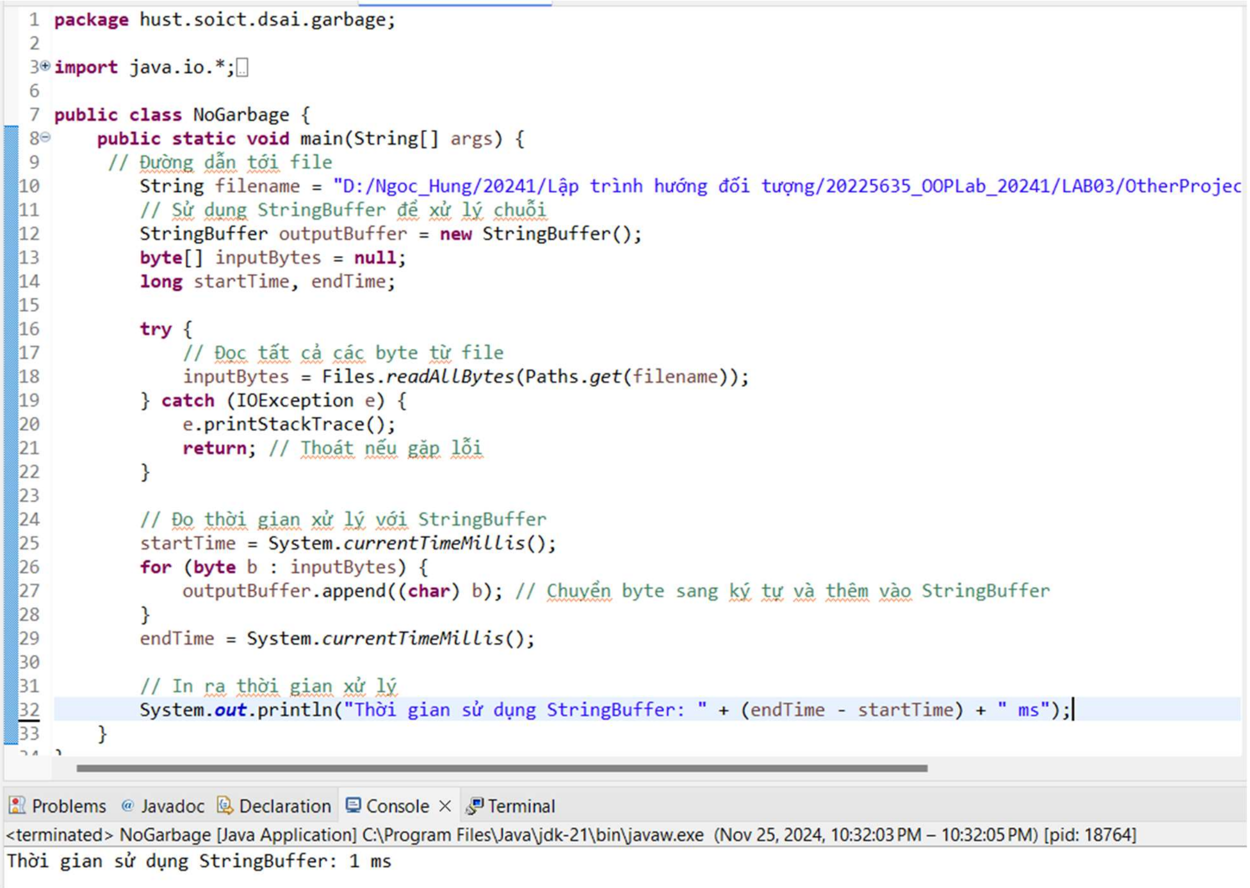
```java
package hust.soict.dsai.garbage;

import java.io.IOException;

public class GarbageCreator {
    public static void main(String[] args) {
        String filename = "D:/Ngoc_Hung/20241/Lập trình hướng đối tượng/20225635_OOPLab_20241/LAB03/OtherProjec
        byte[] inputBytes = new byte[0]; // Khai báo mảng byte
        long startTime, endTime;

        try {
            // Đọc toàn bộ nội dung của tệp vào một mảng byte
            inputBytes = Files.readAllBytes(Paths.get(filename));
            startTime = System.currentTimeMillis();

            // Chuyển đổi byte thành chuỗi (nếu cần thiết) bằng toán tử "+"
            String outputString = "";
            for (byte b : inputBytes) {
                outputString += (char) b; // Nối chuỗi bằng toán tử "+"
            }

            endTime = System.currentTimeMillis();
            System.out.println("Thời gian xử lý với String concatenation (sử dụng +): " + (endTime - startTime)

            // Nếu muốn sử dụng StringBuffer, thay đổi đoạn mã sau:
            startTime = System.currentTimeMillis();

            StringBuffer sb = new StringBuffer();
            for (byte b : inputBytes) {
                sb.append((char) b); // Nối chuỗi bằng StringBuffer
            }
```



```java
            outputString = sb.toString(); // Chuyển StringBuffer thành chuỗi

            endTime = System.currentTimeMillis();
            System.out.println("Thời gian xử lý với StringBuffer: " + (endTime - startTime) + " ms");

        } catch (IOException e) {
            System.out.println("Lỗi khi đọc tệp: " + e.getMessage());
        }
    }
}

```

```
<terminated> GarbageCreator [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 25, 2024, 10:28:11 PM – 10:28:14 PM) [pid: 32428]
Thời gian xử lý với String concatenation (sử dụng +): 16 ms
Thời gian xử lý với StringBuffer: 1 ms
```

NoGarbage.java:

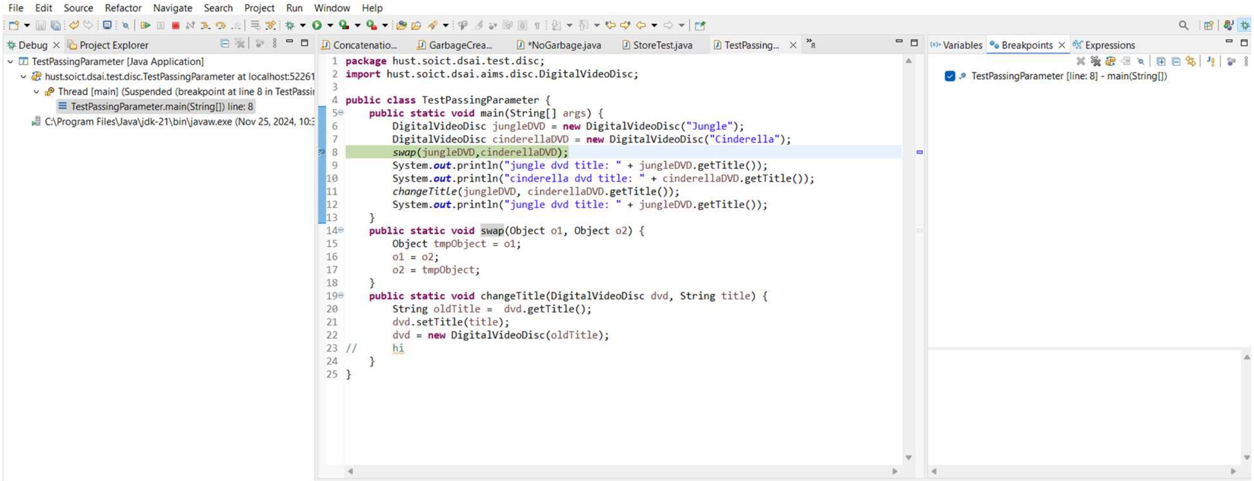


```java
package hust.soict.dsai.garbage;

import java.io.*;

public class NoGarbage {
    public static void main(String[] args) {
     // Đường dẫn tới file
        String filename = "D:/Ngoc_Hung/20241/Lập trình hướng đối tượng/20225635_OOPLab_20241/LAB03/OtherProjec
        // Sử dụng StringBuffer để xử lý chuỗi
        StringBuffer outputBuffer = new StringBuffer();
        byte[] inputBytes = null;
        long startTime, endTime;

        try {
            // Đọc tất cả các byte từ file
            inputBytes = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            e.printStackTrace();
            return; // Thoát nếu gặp lỗi
        }

        // Đo thời gian xử lý với StringBuffer
        startTime = System.currentTimeMillis();
        for (byte b : inputBytes) {
            outputBuffer.append((char) b); // Chuyển byte sang ký tự và thêm vào StringBuffer
        }
        endTime = System.currentTimeMillis();

        // In ra thời gian xử lý
        System.out.println("Thời gian sử dụng StringBuffer: " + (endTime - startTime) + " ms");
    }
```

```
<terminated> NoGarbage [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 25, 2024, 10:32:03 PM – 10:32:05 PM) [pid: 18764]
Thời gian sử dụng StringBuffer: 1 ms
```

2. Code debugging and breakpoints

Running in debugging mode:

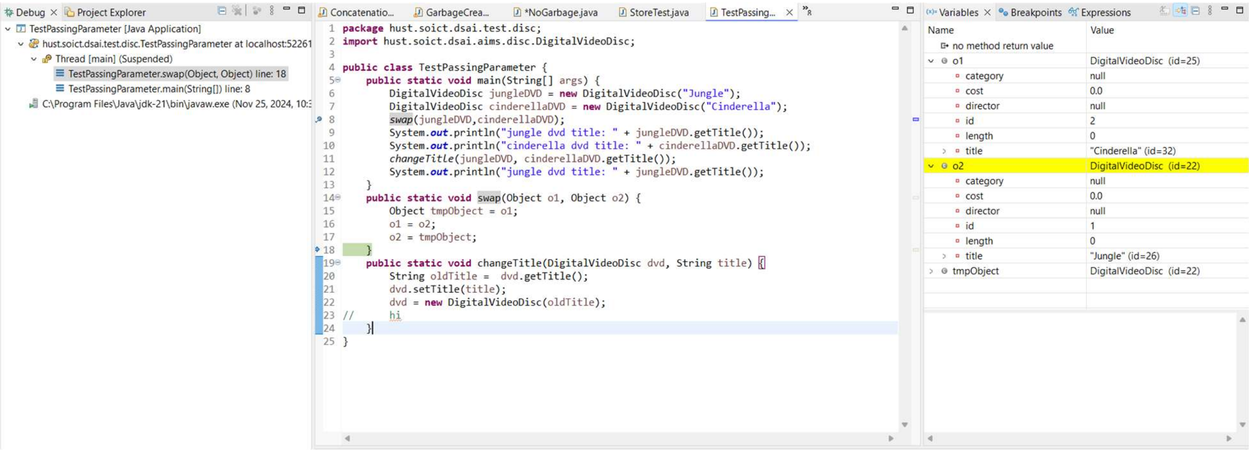


```
TestPassingParameter [Java Application]
  hust.soict.dsai.test.disc.TestPassingParameter at localhost:52261
    Thread [main] (Suspended (breakpoint at line 8 in TestPassir
      TestPassingParameter.main(String[]) line: 8
  C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 25, 2024, 10:3
```

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

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 tmpObject = o1;
        o1 = o2;
        o2 = tmpObject;
    }
    public static void changeTitle(DigitalVideoDisc dvd, String title) {
        String oldTitle =  dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
//        hi
    }
}
```

☑ TestPassingParameter [line: 8] - main(String[])

Variable perspective:

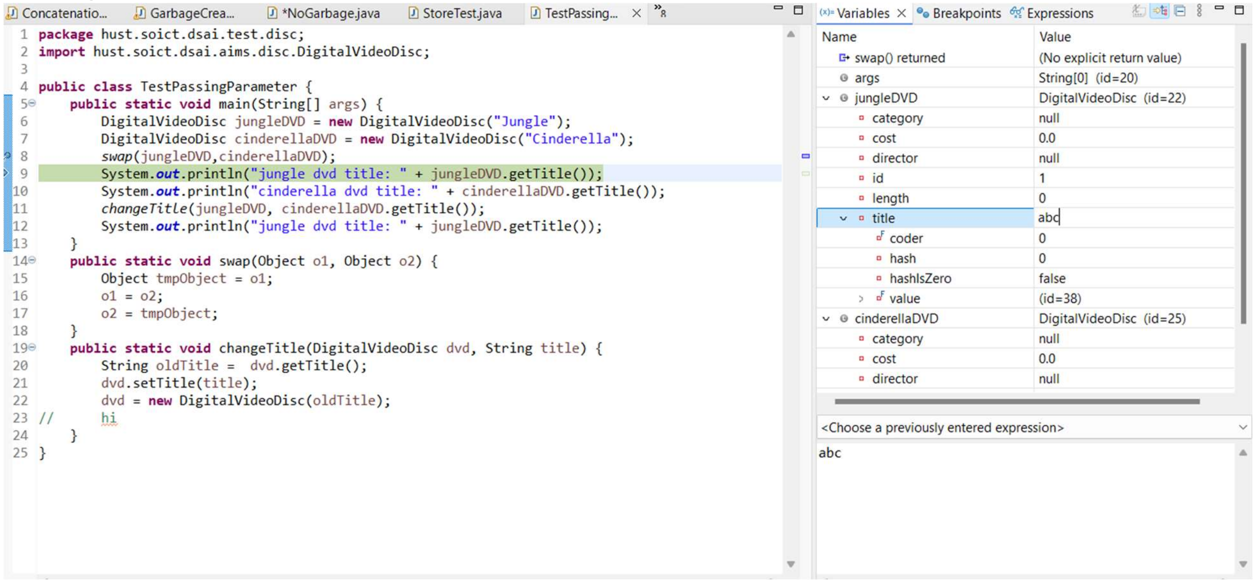


```
TestPassingParameter [Java Application]
  hust.soict.dsai.test.disc.TestPassingParameter at localhost:52261
    Thread [main] (Suspended)
      TestPassingParameter.swap(Object, Object) line: 18
      TestPassingParameter.main(String[]) line: 8
  C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 25, 2024, 10:3
```

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

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 tmpObject = o1;
        o1 = o2;
        o2 = tmpObject;
    }
    public static void changeTitle(DigitalVideoDisc dvd, String title) {
        String oldTitle =  dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
//      hi
    }
}
```

Name	Value
no method return value	
o1	DigitalVideoDisc (id=25)
category	null
cost	0.0
director	null
id	2
length	0
title	"Cinderella" (id=32)
o2	DigitalVideoDisc (id=22)
category	null
cost	0.0
director	null
id	1
length	0
title	"Jungle" (id=26)
tmpObject	DigitalVideoDisc (id=22)

Change title value:

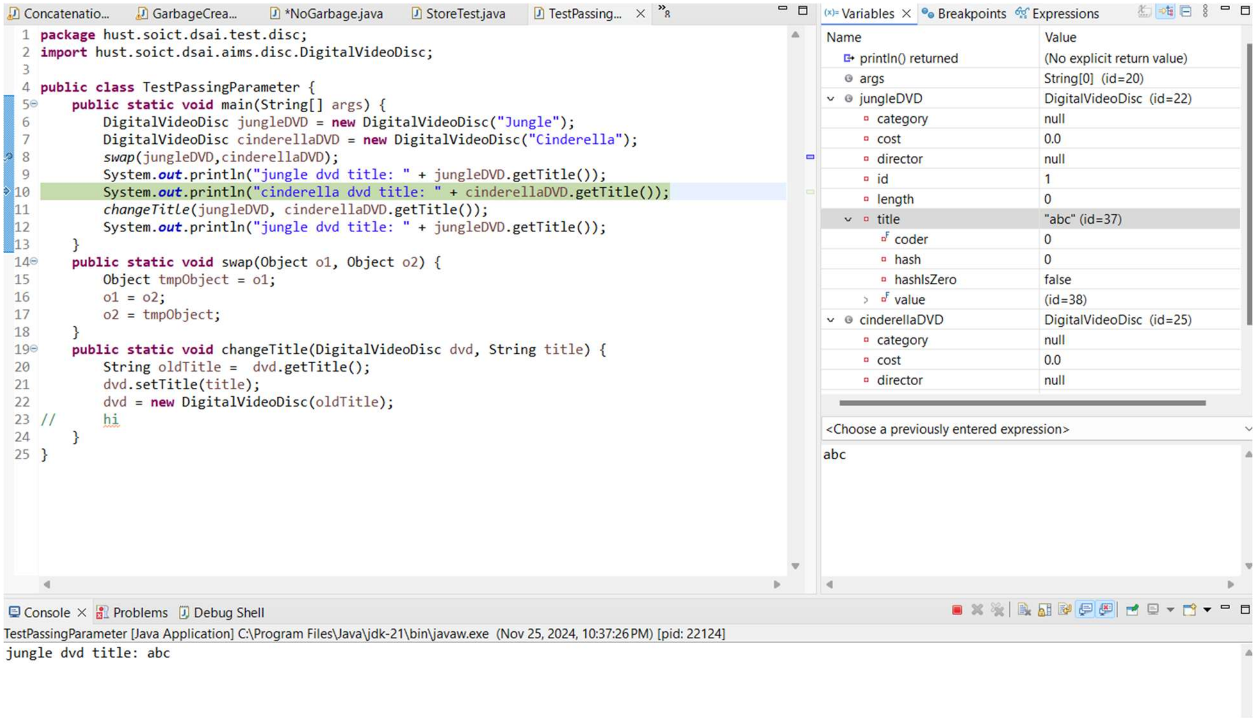


```java
package hust.soict.dsai.test.disc;
import hust.soict.dsai.aims.disc.DigitalVideoDisc;

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 tmpObject = o1;
        o1 = o2;
        o2 = tmpObject;
    }
    public static void changeTitle(DigitalVideoDisc dvd, String title) {
        String oldTitle =  dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
//      hi
    }
}
```

Name	Value
swap() returned	(No explicit return value)
args	String[0] (id=20)
jungleDVD	DigitalVideoDisc (id=22)
category	null
cost	0.0
director	null
id	1
length	0
title	abc
coder	0
hash	0
hashIsZero	false
value	(id=38)
cinderellaDVD	DigitalVideoDisc (id=25)
category	null
cost	0.0
director	null

abc

Result:



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

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 tmpObject = o1;
        o1 = o2;
        o2 = tmpObject;
    }
    public static void changeTitle(DigitalVideoDisc dvd, String title) {
        String oldTitle =  dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc(oldTitle);
//      hi
    }
}
```

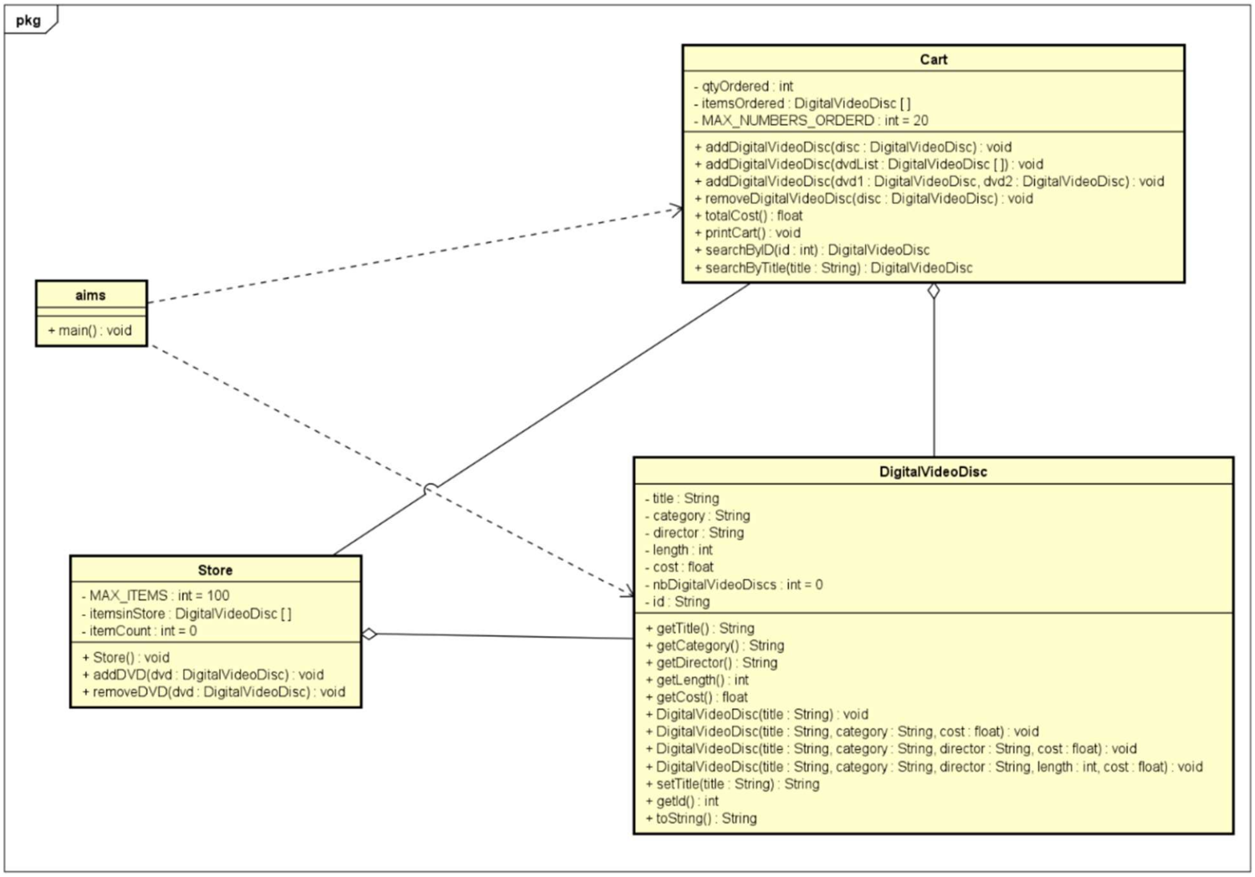
Name	Value
println() returned	(No explicit return value)
args	String[0] (id=20)
jungleDVD	DigitalVideoDisc (id=22)
category	null
cost	0.0
director	null
id	1
length	0
title	"abc" (id=37)
coder	0
hash	0
hashIsZero	false
value	(id=38)
cinderellaDVD	DigitalVideoDisc (id=25)
category	null
cost	0.0
director	null

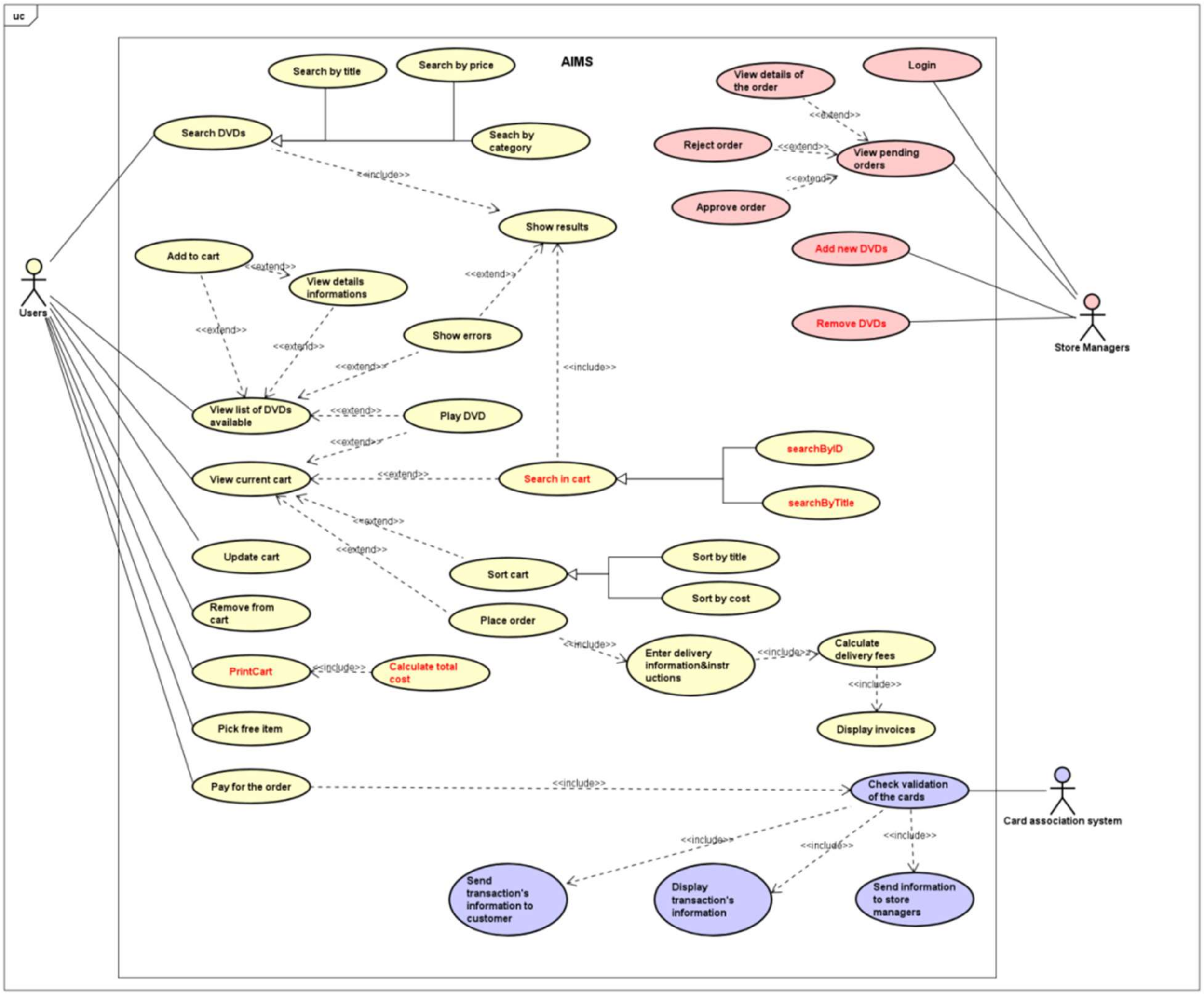
<Choose a previously entered expression>

abc

```
TestPassingParameter [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Nov 25, 2024, 10:37:26 PM) [pid: 22124]
jungle dvd title: abc
```

3. Usecase diagram and Class diagram Updates:


pkg
Cart
- qtyOrdered : int
- itemsOrdered : DigitalVideoDisc [ ]
- MAX_NUMBERS_ORDERD : int = 20
+ addDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ addDigitalVideoDisc(dvdList : DigitalVideoDisc [ ]) : void
+ addDigitalVideoDisc(dvd1 : DigitalVideoDisc, dvd2 : DigitalVideoDisc) : void
+ removeDigitalVideoDisc(disc : DigitalVideoDisc) : void
+ totalCost() : float
+ printCart() : void
+ searchByID(id : int) : DigitalVideoDisc
+ searchByTitle(title : String) : DigitalVideoDisc
aims
+ main() : void
Store
- MAX_ITEMS : int = 100
- itemsinStore : DigitalVideoDisc [ ]
- itemCount : int = 0
+ Store() : void
+ addDVD(dvd : DigitalVideoDisc) : void
+ removeDVD(dvd : DigitalVideoDisc) : void
DigitalVideoDisc
- title : String
- category : String
- director : String
- length : int
- cost : float
- nbDigitalVideoDiscs : int = 0
- id : String
+ getTitle() : String
+ getCategory() : String
+ getDirector() : String
+ getLength() : int
+ getCost() : float
+ DigitalVideoDisc(title : String) : void
+ DigitalVideoDisc(title : String, category : String, cost : float) : void
+ DigitalVideoDisc(title : String, category : String, director : String, cost : float) : void
+ DigitalVideoDisc(title : String, category : String, director : String, length : int, cost : float) : void
+ setTitle(title : String) : String
+ getId() : int
+ toString() : String



4. Answer questions:

Câu hỏi trang 6 phần 2.1

Trả lời: Tôi thích sử dụng tham số mảng hơn.

Câu hỏi trang 6 phần 3

Trả lời: JAVA là ngôn ngữ lập trình theo kiểu Pass by Value.

Câu hỏi trang 8 phần 3

- Tiêu đề của hai đối tượng này vẫn không thay đổi sau khi gọi phương thức swap vì Java truyền tham chiếu đối tượng theo kiểu Pass by Value, và phương thức chỉ hoán đổi các tham chiếu cục bộ, không thay đổi các tham chiếu gốc.

- Tiêu đề của jungleDVD bị thay đổi vì phương thức changeTitle thay đổi trực tiếp trạng thái của đối tượng jungleDVD bằng cách gọi phương thức setTitle, làm cập nhật tiêu đề của đối tượng được truyền vào.

Câu hỏi trang 16 phần 6

- Kiểu dữ liệu trả về nên là String