ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH
IT1130-744360-2024.1

BÀI THỰC HÀNH 3

Họ và tên sv:Nguyễn Đức Bình

MSSV: 20225791

Lớp: Việt Nhật 03

GVHD: Lê Thị Hoa

HTGD: Bùi Trọng Dũng

Hà Nội 11/2024

BÁO CÁO THỰC HÀNH LAB 03
THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. Working with method overloading

1.1. Overloading by differing types of parameter

Code

```
public void addDigitalVideoDisc(DigitalVideoDisc[] discs) {
    for (DigitalVideoDisc disc : discs) {
        if (qtyOrdered5791 < MAX_NUMBER_ORDERED5791) {
            itemsOrdered5791[qtyOrdered5791] = disc;
            qtyOrdered5791++;
            System.out.println("The disc has been added.");
            if (qtyOrdered5791 == MAX_NUMBER_ORDERED5791) {
                System.out.println("The cart is almost full");
            }
        } else {
            System.out.println("The cart is full. Cannot add more discs.");
            break; // Stop adding if the cart is full
        }
    }
}
```

Figure 1Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

Result

```
        Cart anOrder5791 = new Cart();
        DigitalVideoDisc[] disc_5791 = new DigitalVideoDisc[] {
                new DigitalVideoDisc("Movie 1", "Action", "Director 1", 120, 15.99f),
                new DigitalVideoDisc("Movie 2", "Comedy", "Director 2", 90, 12.99f),
                new DigitalVideoDisc("Movie 3", "Horror", "Director 3", 100, 9.99f)
            };
        anOrder5791.addDigitalVideoDisc(disc_5791);
        anOrder5791.printCart5791();
    }
}
```

Problems Javadoc Declaration Console Terminal

```
<terminated> Aims (1) [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 22, 2024, 6:01:21 PM – 6:01:22 PM) [pid: 16536]
The disc has been added.
The disc has been added.
The disc has been added.
Cart contents:
DigitalVideoDisc{title='Movie 1', category='Action', director='Director 1', length=120, cost=15.99}
DigitalVideoDisc{title='Movie 2', category='Comedy', director='Director 2', length=90, cost=12.99}
DigitalVideoDisc{title='Movie 3', category='Horror', director='Director 3', length=100, cost=9.99}
```

Figure 2 Result

1.2. Overloading by differing the number of parameters

Code

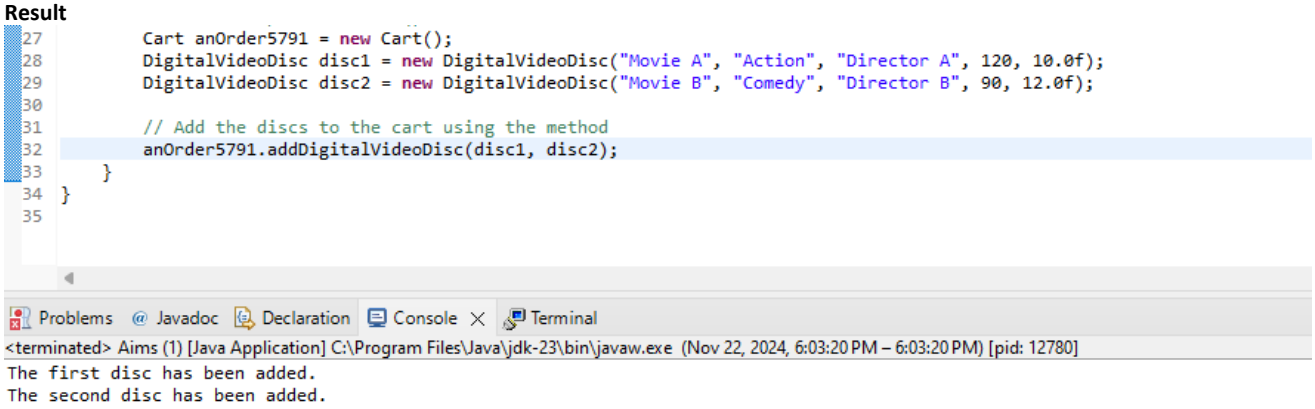
```
public void addDigitalVideoDisc(DigitalVideoDisc disc1, DigitalVideoDisc disc2) {
    if (qtyOrdered5791 < MAX_NUMBER_ORDERED5791) {
        // Add disc1
        itemsOrdered5791[qtyOrdered5791] = disc1;
        qtyOrdered5791++;
        System.out.println("The first disc has been added.");

        // Check if the cart is almost full after adding disc1
        if (qtyOrdered5791 == MAX_NUMBER_ORDERED5791 - 1) {
            System.out.println("The cart is almost full.");
        }

        // Add disc2
        if (qtyOrdered5791 < MAX_NUMBER_ORDERED5791) {
            itemsOrdered5791[qtyOrdered5791] = disc2;
            qtyOrdered5791++;
            System.out.println("The second disc has been added.");
        } else {
            System.out.println("The cart is full. Cannot add the second disc.");
        }
    } else {
        System.out.println("The cart is full. Cannot add more discs.");
    }
}
```

Figure 3Method addDigitalVideoDisc(DigitalVideoDisc disc1.DigitalVideoDisc disc2)

Result



```
        Cart anOrder5791 = new Cart();
        DigitalVideoDisc disc1 = new DigitalVideoDisc("Movie A", "Action", "Director A", 120, 10.0f);
        DigitalVideoDisc disc2 = new DigitalVideoDisc("Movie B", "Comedy", "Director B", 90, 12.0f);

        // Add the discs to the cart using the method
        anOrder5791.addDigitalVideoDisc(disc1, disc2);
    }
}
```

Problems Javadoc Declaration Console Terminal

<terminated> Aims (1) [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 22, 2024, 6:03:20 PM – 6:03:20 PM) [pid: 12780]

```
The first disc has been added.
The second disc has been added.
```

Figure 4 Result

2. Passing parameter

Code

```java
package aimProject;

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 swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
        // Temporarily store the title of the first DVD
        String tempTitle = dvd1.getTitle();

        // Swap the titles
        dvd1.setTitle(dvd2.getTitle());
        dvd2.setTitle(tempTitle);
    }

    public static void changeTitle(DigitalVideoDisc dvd5791, String title5791) {
        String oldTitle5791 = dvd5791.getTitle();
        dvd5791.setTitle(title5791);
        dvd5791 = new DigitalVideoDisc(oldTitle5791);
    }
}
```

Figure 5Passing parameter code

Result

```
jungle dvd title: Cinderella
cinderella dvd title: Jungle
jungle dvd title: Jungle
```

Figure 6Passing parameter result

3. Class Member and Instance Member

Code

```
private float cost=0;
private static int nbDigigtalVideoDiscs=0;
private int id;
public int getId() {
    return id;
}
// Constructor to create a DVD object by title
public DigitalVideoDisc(String _title5791) {
    this.title = _title5791;
    nbDigigtalVideoDiscs++;
    this.id = nbDigigtalVideoDiscs;
}

// Constructor to create a DVD object by category, title, and cost
public DigitalVideoDisc(String _category5791, String _title5791, float _c
    this.category = _category5791;
    this.title = _title5791;
    this.cost = _cost5791;
    nbDigigtalVideoDiscs++;
    this.id = nbDigigtalVideoDiscs;
}

// Constructor to create a DVD object by director, category, title, and c
public DigitalVideoDisc(String _director5791, String _category5791, Strin
    this.director = _director5791;
    this.category = _category5791;
    this.title = _title5791;
    this.cost = _cost5791;
    nbDigigtalVideoDiscs++;
    this.id = nbDigigtalVideoDiscs;
}
```

Figure 7Class member and Instance member

Result

```
2
3  public class Aims {
4      public static void main(String[] args) {
5          // Create a new cart
6          //Cart anOrder5791 = new Cart();
7
8          // Create new dvd objects and add them to the cart
9          DigitalVideoDisc dvd1_5791 = new DigitalVideoDisc("The Lion King","Animation", "Roger Allers", 87, 19.95f);
10         //anOrder5791.addDigitalVideoDisc(dvd1_5791);
11
12         DigitalVideoDisc dvd2_5791 = new DigitalVideoDisc("Star Wars","Science Fiction", "George Lucas", 87, 24.95f)
13         //anOrder5791.addDigitalVideoDisc(dvd2_5791);
14
15         DigitalVideoDisc dvd3_5791 = new DigitalVideoDisc("Aladin","Animation", 18.99f);
16         //anOrder5791.addDigitalVideoDisc(dvd3_5791);
17         //anOrder5791.addDigitalVideoDisc(dvd2_5791, dvd3_5791);
18         // Print total cost of the items in the cart
19         /*System.out.print("Total cost is: ");
20         System.out.println(anOrder5791.totalCost());
21         anOrder5791.removeDigitalVideoDisc(dvd3_5791);
22         System.out.print("Total cost is: ");
23         System.out.println(anOrder5791.totalCost());
24         DigitalVideoDisc[] dvds_5791 = { dvd1_5791,dvd2_5791 ,dvd3_5791 };
25         anOrder5791.addDigitalVideoDisc(dvds_5791);
26         anOrder5791.printCart5791();*/
27         Cart anOrder5791 = new Cart();
28         DigitalVideoDisc disc1 = new DigitalVideoDisc("Movie A", "Action", "Director A", 120, 10.0f);
29         DigitalVideoDisc disc2 = new DigitalVideoDisc("Movie B", "Comedy", "Director B", 90, 12.0f);
30
31         // Add the discs to the cart using the method
32         anOrder5791.addDigitalVideoDisc(disc1, disc2);
33         System.out.println(disc2.getId());
34     }
35 }
36
```

Problems @ Javadoc Declaration Console Terminal

```
<terminated> Aims (1) [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 22, 2024, 6:57:45 PM – 6:57:45 PM) [pid: 22048]
The first disc has been added.
The second disc has been added.
5
```

Figure 8 Result

4. Open the Cart class

Code

```
public void print5791() {
    System.out.println("******************CART******************");
    System.out.println("Ordered Items:");
    float totalCost = 0;
    for (DigitalVideoDisc disc : itemsOrdered5791) {
        if (disc != null) { // Check for null before accessing properties
            System.out.println(disc); // Assumes toString() is properly implemented in DigitalVideoDisc
            totalCost += disc.getCost();
        }
    }
    System.out.println("****************************************");
    System.out.println("Total cost: " + totalCost + " $");
}
```

Figure 9 print

```
// Search DVDs by ID
public void searchById(int id) {
    boolean found = false;

    for (DigitalVideoDisc disc : itemsOrdered5791) { // Loop through the list/array
        if (disc != null && disc.getId() == id) { // Add a null check before accessing getId()
            System.out.println("Found: " + disc); // Assumes toString() is implemented in DigitalVideoDisc
            found = true;
            break; // Exit loop after finding the match
        }
    }

    if (!found) {
        System.out.println("No DVD found with ID: " + id);
    }
}
```

Figure 10search by id

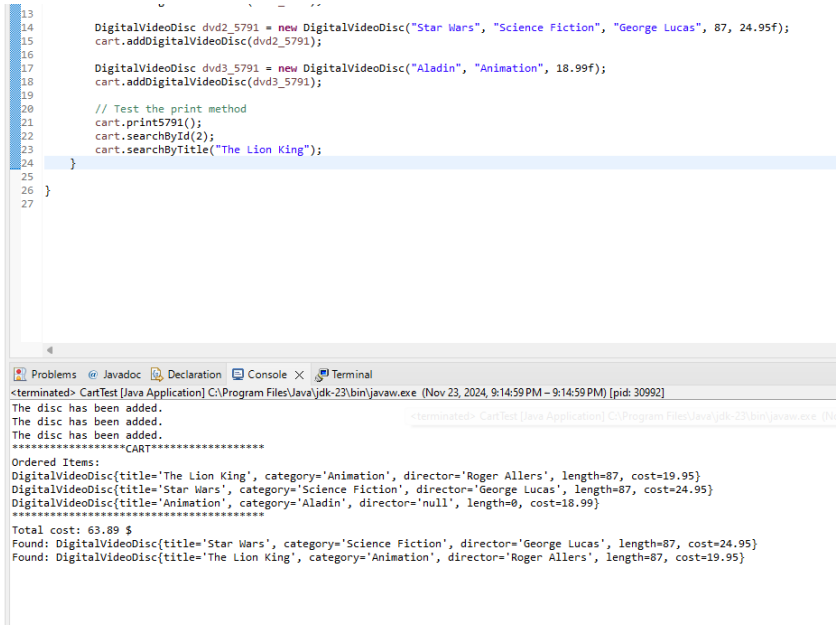
```
public void searchByTitle(String title) {
    boolean found = false;

    for (DigitalVideoDisc disc : itemsOrdered5791) { // Loop through the list/array
        if (disc != null && disc.getTitle().equalsIgnoreCase(title)) { // Check for null and compare titles (case-insensitive)
            System.out.println("Found: " + disc); // Print details of the matched DVD
            found = true;
        }
    }

    if (!found) {
        System.out.println("No DVD found with title: " + title);
    }
}
```

Figure 11 Search by title

Result



```
        DigitalVideoDisc dvd2_5791 = new DigitalVideoDisc("Star Wars", "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addDigitalVideoDisc(dvd2_5791);

        DigitalVideoDisc dvd3_5791 = new DigitalVideoDisc("Aladin", "Animation", 18.99f);
        cart.addDigitalVideoDisc(dvd3_5791);

        // Test the print method
        cart.print5791();
        cart.searchById(2);
        cart.searchByTitle("The Lion King");
    }
}
```

Problems Javadoc Declaration Console Terminal

```
<terminated> CartTest [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 23, 2024, 9:14:59 PM – 9:14:59 PM) [pid: 30992]
The disc has been added.
The disc has been added.
The disc has been added.
**********CART**********
Ordered Items:
DigitalVideoDisc{title='The Lion King', category='Animation', director='Roger Allers', length=87, cost=19.95}
DigitalVideoDisc{title='Star Wars', category='Science Fiction', director='George Lucas', length=87, cost=24.95}
DigitalVideoDisc{title='Animation', category='Aladin', director='null', length=0, cost=18.99}
**********************************
Total cost: 63.89 $
Found: DigitalVideoDisc{title='Star Wars', category='Science Fiction', director='George Lucas', length=87, cost=24.95}
Found: DigitalVideoDisc{title='The Lion King', category='Animation', director='Roger Allers', length=87, cost=19.95}
```

Figure 12 result

5. Implement the Store class

Code

```java
package aimProject;

public class Store {
    private DigitalVideoDisc[] dvdsInStore5791; // Array to store DVDs in the store
    private int currentCount5791; // Tracks the current number of DVDs in the store
    private int maxCapacity5791;  // Maximum capacity of the store

    // Constructor: Initializes the store with the given maximum capacity
    public Store(int maxCapacity) {
        this.maxCapacity5791 = maxCapacity;
        this.dvdsInStore5791 = new DigitalVideoDisc[maxCapacity]; // Initialize the array
        this.currentCount5791 = 0; // Initially, the store is empty
    }

    // Add a DVD to the store
    public void addDVD(DigitalVideoDisc dvd) {
        if (currentCount5791 < maxCapacity5791) {
            dvdsInStore5791[currentCount5791] = dvd;
            currentCount5791++;
            System.out.println("DVD added to the store: " + dvd.getTitle());
        } else {
            System.out.println("The store is full. Cannot add more DVDs.");
        }
    }

    // Remove a DVD from the store by its title
    public void removeDVD(String title5791) {
        boolean removed = false;
        for (int i = 0; i < currentCount5791; i++) {
            if (dvdsInStore5791[i] != null && dvdsInStore5791[i].getTitle().equalsIgnoreCase(title5791)) {
                // Shift the remaining DVDs to fill the gap
                for (int j = i; j < currentCount5791 - 1; j++) {
                    dvdsInStore5791[j] = dvdsInStore5791[j + 1];
                }
                dvdsInStore5791[currentCount5791 - 1] = null; // Clear the last slot
                currentCount5791--;
                System.out.println("DVD removed from the store: " + title5791);
```

```java
package aimProject;

public class Store {
    private DigitalVideoDisc[] dvdsInStore5791; // Array to store DVDs in the store
    private int currentCount5791; // Tracks the current number of DVDs in the store
    private int maxCapacity5791;  // Maximum capacity of the store

    // Constructor: Initializes the store with the given maximum capacity
    public Store(int maxCapacity) {
        this.maxCapacity5791 = maxCapacity;
        this.dvdsInStore5791 = new DigitalVideoDisc[maxCapacity]; // Initialize the array
        this.currentCount5791 = 0; // Initially, the store is empty
    }

    // Add a DVD to the store
    public void addDVD(DigitalVideoDisc dvd) {
        if (currentCount5791 < maxCapacity5791) {
            dvdsInStore5791[currentCount5791] = dvd;
            currentCount5791++;
            System.out.println("DVD added to the store: " + dvd.getTitle());
        } else {
            System.out.println("The store is full. Cannot add more DVDs.");
        }
    }

    // Remove a DVD from the store by its title
    public void removeDVD(String title5791) {
        boolean removed = false;
        for (int i = 0; i < currentCount5791; i++) {
            if (dvdsInStore5791[i] != null && dvdsInStore5791[i].getTitle().equalsIgnoreCase(title5791)) {
                // Shift the remaining DVDs to fill the gap
                for (int j = i; j < currentCount5791 - 1; j++) {
                    dvdsInStore5791[j] = dvdsInStore5791[j + 1];
                }
                dvdsInStore5791[currentCount5791 - 1] = null; // Clear the last slot
                currentCount5791--;
                System.out.println("DVD removed from the store: " + title5791);
```


```java
    // Remove a DVD from the store by its title
    public void removeDVD(String title5791) {
        boolean removed = false;
        for (int i = 0; i < currentCount5791; i++) {
            if (dvdsInStore5791[i] != null && dvdsInStore5791[i].getTitle().equalsIgnoreCase(title5791)) {
                // Shift the remaining DVDs to fill the gap
                for (int j = i; j < currentCount5791 - 1; j++) {
                    dvdsInStore5791[j] = dvdsInStore5791[j + 1];
                }
                dvdsInStore5791[currentCount5791 - 1] = null; // Clear the last slot
                currentCount5791--;
                System.out.println("DVD removed from the store: " + title5791);
                removed = true;
                break;
            }
        }
        if (!removed) {
            System.out.println("No DVD found with title: " + title5791);
        }
    }

    // Display all DVDs in the store
    public void displayStore() {
        System.out.println("******************STORE******************");
        if (currentCount5791 == 0) {
            System.out.println("The store is empty.");
        } else {
            for (int i = 0; i < currentCount5791; i++) {
                if (dvdsInStore5791[i] != null) {
                    System.out.println((i + 1) + ". " + dvdsInStore5791[i]); // Assumes toString() is implemented in DigitalVideoDisc
                }
            }
        }
        System.out.println("*****************************************");
    }
```


Figure 13 Store class

Result

```java
        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Titanic", "Romance", "James Cameron", 195, 14.99f);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Avatar", "Sci-Fi", "James Cameron", 162, 22.99f);
        DigitalVideoDisc dvd4 = new DigitalVideoDisc("The Lion King", "Animation", "Jon Favreau", 118, 12.99f);

        // Add DVDs to the store
        store.addDVD(dvd1);
        store.addDVD(dvd2);
        store.addDVD(dvd3);
        store.addDVD(dvd4);

        // Display all DVDs in the store
        System.out.println("\nDisplaying the store:");
        store.displayStore();

        // Remove a DVD by its title
        System.out.println("\nRemoving 'Avatar' from the store:");
        store.removeDVD("Avatar");

        // Display the store again to verify removal
        System.out.println("\nDisplaying the store after removal:");
        store.displayStore();

        // Try removing a DVD that doesn't exist
        System.out.println("\nTrying to remove a non-existent DVD:");
        store.removeDVD("NonExistentMovie");

        // Add a new DVD to check if the store works correctly after removal
        DigitalVideoDisc dvd5 = new DigitalVideoDisc("Frozen", "Animation", "Chris Buck", 102, 9.99f);
        System.out.println("\nAdding 'Frozen' to the store:");
        store.addDVD(dvd5);

        // Display the store again
        System.out.println("\nDisplaying the store after adding 'Frozen':");
        store.displayStore();
    }
}
```

```
Problems  Javadoc  Declaration  Console  Terminal
rminated> Store_Test [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 23, 2024, 9:21:16 PM – 9:21:16 PM) [pid: 12068]
) added to the store: Inception
) added to the store: Titanic
) added to the store: Avatar
) added to the store: The Lion King

splaying the store:
******************STORE******************
DigitalVideoDisc{title='Inception', category='Sci-Fi', director='Christopher Nolan', length=148, cost=19.99}
DigitalVideoDisc{title='Titanic', category='Romance', director='James Cameron', length=195, cost=14.99}
DigitalVideoDisc{title='Avatar', category='Sci-Fi', director='James Cameron', length=162, cost=22.99}
DigitalVideoDisc{title='The Lion King', category='Animation', director='Jon Favreau', length=118, cost=12.99}
*****************************************

moving 'Avatar' from the store:
) removed from the store: Avatar

splaying the store after removal:
```

Figure 14 Store_test

6. String, StringBuilder and StringBuffer

Code

```java
package garbage;

import java.util.Random;

public class ConcatenationInLoops {

    public static void main(String[] args) {
        // Initialize Random object
        Random r5791 = new Random(123);

        // Measure time for string concatenation (inefficient)
        long start = System.currentTimeMillis();
        String s5791 = "";
        for (int i = 0; i < 65536; i++) {
            s5791 += r5791.nextInt(2);
        }
        System.out.println("Time taken with String concatenation: " + (System.currentTimeMillis() - start) + " ms");

        // Reset Random object
        r5791 = new Random(123);

        // Measure time for StringBuilder (efficient)
        start = System.currentTimeMillis();
        StringBuilder sb5791 = new StringBuilder();
        for (int i = 0; i < 65536; i++) {
            sb5791.append(r5791.nextInt(2));
        }
        String result5791 = sb5791.toString();
        System.out.println("Time taken with StringBuilder: " + (System.currentTimeMillis() - start) + " ms");
    }
}
```

Figure 15 CIL

```java
package garbage;
import java.io.IOException;
public class GarbageCreator {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         String filename = "C:\\Users\\manht\\eclipse-workspace\\garbage\\src\\garbage\\test.txt";
            byte[] inputBytes5791 = { 0 };
            long startTime5791, endTime5791;

            try {
                inputBytes5791 = Files.readAllBytes(Paths.get(filename));
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            startTime5791 = System.currentTimeMillis();
            String outputString5791 = "";
            for (byte b : inputBytes5791) {
                outputString5791 += (char)b;
            }
            endTime5791 = System.currentTimeMillis();
            System.out.println(endTime5791 - startTime5791);
    }
}
```

Figure 16 GC

```java
package garbage;
import java.io.IOException;
public class NoGarbage {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String filename = "C:\\Users\\manht\\eclipse-workspace\\garbage\\src\\garbage\\test.txt";
        byte[] inputBytes5791 = { 0 };
        long startTime5791, endTime5791;

        try {
            inputBytes5791 = Files.readAllBytes(Paths.get(filename));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        startTime5791 = System.currentTimeMillis();
        StringBuilder outpStringBuilder = new StringBuilder();
        for (byte b : inputBytes5791) {
            outpStringBuilder.append((char)b);
        }
        endTime5791 = System.currentTimeMillis();
        System.out.println(endTime5791 - startTime5791);
    }

}
```

Figure 17 NG

Result

```
Time taken with String concatenation: 462 ms
Time taken with StringBuilder: 2 ms
```

Figure 18 CIL result

```
19
```

Figure 19 GC

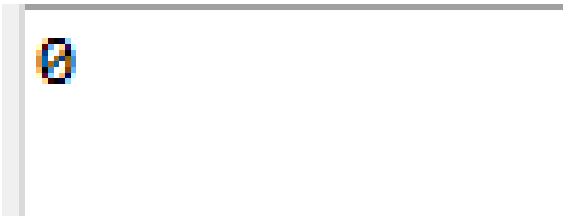


Figure 20 NG

7 Class Diagram

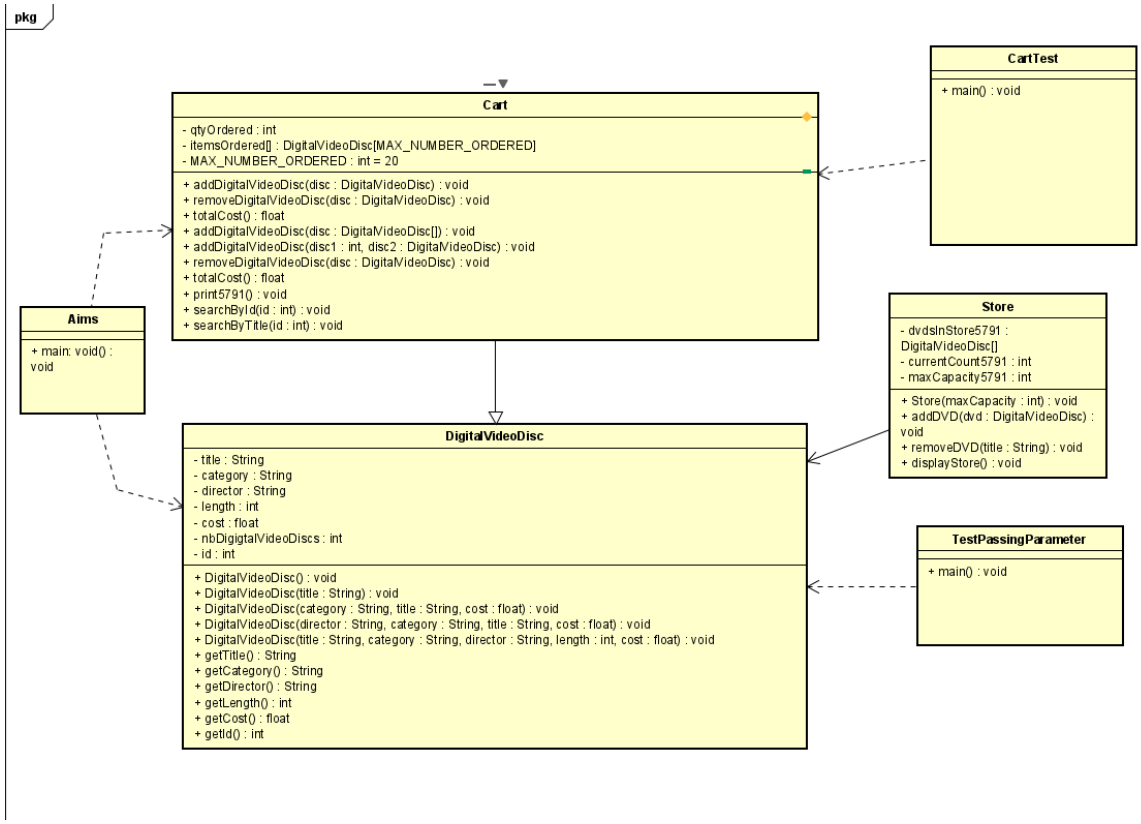


Figure 21 class diagram