

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

**BÁO CÁO THỰC HÀNH**  
**IT3103-744527-2024.1**  
**BÀI THỰC HÀNH - LAB03**

Họ tên SV: Lý Công Tiến  
MSSV: 20225934  
Lớp: Việt-Nhật 04  
GVHD: Lê Thị Hoa  
HTGD: Đặng Mạnh Cường

Hà Nội 9/2024

## Contents

1. Working with method overloading .....	3
1.1. Overloading by differing types of parameter .....	3
1.2. Overloading by differing the number of parameters .....	4
Code .....	4
2. Passing parameter .....	5
Code .....	6
Result.....	6
3. Class Member and Instance Member .....	7
Code .....	8
Result.....	9
4. Open the Cart class .....	9
Code .....	9
Result.....	10
5. Implement the Store class.....	11
Code .....	11
Result.....	12
6. String, StringBuilder and StringBuffer .....	12
Code .....	13
Result.....	13
7. Class Diagram .....	14

## 1. Working with method overloading

### 1.1. Overloading by differing types of parameter

Code

```
public int addDigitalVideoDisc(DigitalVideoDiscTienLC[] dvdlist){
    int addedCount = 0;
    for(DigitalVideoDiscTienLC disc : dvdlist){
        if(qtyOrdered == MAX_NUMBERS_ORDERED){
            System.out.println("The cart is almost full. Can't add more disc");
            break;
        }
        else{
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD " + '\'' + disc.getTitle() + '\'' + " has been added!");
            addedCount++;
        }
    }
    return addedCount;
}
```

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

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

import hust.soict.dsai.aimsproject.disc.DigitalVideoDiscTienLC;
import hust.soict.dsai.aimsproject.cart.CartTienLC;
import hust.soict.dsai.aimsproject.store.StoreTienLC;

public class Aims {
    public static void main(String[] args) {
        CartTienLC anOrder = new CartTienLC();

        //Them dia vao gio hang
        DigitalVideoDiscTienLC dvd1 = new DigitalVideoDiscTienLC("Tom and Jerry", "Cartoon", 20.05f, "Gene Deitch", 120);
        DigitalVideoDiscTienLC dvd2 = new DigitalVideoDiscTienLC("The Lion King", "anime", 15.55f, "Roger Allers", 87);
        DigitalVideoDiscTienLC dvd3 = new DigitalVideoDiscTienLC("Toy Story", "anime", 17.75f, "John Lasseter", 150);

        DigitalVideoDiscTienLC[] dvdList = {dvd1, dvd2, dvd3};
        anOrder.addDigitalVideoDisc(dvdList);
        anOrder.print();
        //Kiem tra khi chua thuc hien ham xoa
        // anOrder.print();
        //Kiem tra khi da thuc hien ham xoa
        // anOrder.removeDigitalVideoDisc(dvd2);
        // anOrder.print();
        //Tim kiem DVD trong gio hang sau khi xoa
        // anOrder.searchById(3);
    }
}
```

Figure 2 Aims Class

## Result

```

run:
The DVD "Tom and Jerry" has been added!
The DVD "The Lion king" has been added!
The DVD "Toy Story" has been added!
*****CART*****
Ordered items:
1.[Tom and Jerry] - [Cartoon] - [Gene Deitch] - [120]: 20.05 $
2.[The Lion king] - [anime] - [Roger Allers] - [87]: 15.55 $
3.[Toy Story] - [anime] - [John Lasseter] - [150]: 17.75 $
total: 53.35 $
*****
BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 3 Result

## 1.2. Overloading by differing the number of parameters

## Code

```

public int addDigitalVideoDisc(DigitalVideoDiscTienLC dvd1, DigitalVideoDiscTienLC dvd2) {
    if (qtyOrdered + 1 >= MAX_NUMBERS_ORDERED) {
        System.out.println("The cart is almost full. Can't add more discs.");
        return 0;
    }
    else{
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered++;
        System.out.println("The DVD " + "\"" + dvd1.getTitle() + "\"" + " has been added.");
        itemsOrdered[qtyOrdered] = dvd2;
        qtyOrdered++;
        System.out.println("The DVD " + "\"" + dvd2.getTitle() + "\"" + " has been added.");
        return 2; //Tra ve so dia DVD da them duoc
    }
}

```

Figure 4 Method addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2)

```

package hust.soict.dsai.aimsproject.store;

import hust.soict.dsai.aimsproject.disc.DigitalVideoDiscTienLC;
import hust.soict.dsai.aimsproject.cart.CartTienLC;
import hust.soict.dsai.aimsproject.store.StoreTienLC;

public class Aims {
    public static void main(String[] args) {
        CartTienLC anOrder = new CartTienLC();

        //Them dia vao gio hang
        DigitalVideoDiscTienLC dvd1 = new DigitalVideoDiscTienLC("Tom and Jerry", "Cartoon", 20.05f, "Gene Deitch", 120);
        DigitalVideoDiscTienLC dvd2 = new DigitalVideoDiscTienLC("The Lion king", "anime", 15.55f, "Roger Allers", 87);
        DigitalVideoDiscTienLC dvd3 = new DigitalVideoDiscTienLC("Toy Story", "anime", 17.75f, "John Lasseter", 150);

        anOrder.addDigitalVideoDisc(dvd1, dvd2);
        // DigitalVideoDiscTienLC[] dvdList = {dvd1, dvd2, dvd3};
        // anOrder.addDigitalVideoDisc(dvdList);
        anOrder.print();
        //Kiem tra khi chua thuc hien ham xoa
        // anOrder.print();
        //Kiem tra khi da thuc hien ham xoa
        // anOrder.removeDigitalVideoDisc(dvd2);
        // anOrder.print();
        //Tim kiem DVD trong gio hang sau khi xoa
        // anOrder.searchById(3);
    }
}

```

Figure 5 Aims Class

## Result

```

run:
The DVD "Tom and Jerry" has been added.
The DVD "The Lion king" has been added.
*****CART*****

Ordered items:
1.[Tom and Jerry] - [Cartoon] - [Gene Deitch] - [120]: 20.05 $
2.[The Lion king] - [anime] - [Roger Allers] - [87]: 15.55 $
total: 35.6 $
*****

BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 6 Result

## 2. Passing parameter

Câu hỏi: Java là ngôn ngữ lập trình "Pass by Value" hay "Pass by Reference"?

Trả lời: Java là ngôn ngữ lập trình "Pass by Value". Trong Java, khi bạn truyền một tham số cho một phương thức, giá trị của tham số được sao chép và truyền vào phương thức. Điều này có nghĩa là nếu bạn thay đổi giá trị của tham số bên trong phương thức, giá trị của biến gọi phương thức không bị ảnh hưởng

## Code

```

2  package hust.soict.dsai.test.disc;
3  import hust.soict.dsai.aimsproject.disc.DigitalVideoDiscTienLC;
4  public class TestPassingParameterTienLC {
5      public static void main(String[] args) {
6          DigitalVideoDiscTienLC jungleDVD = new DigitalVideoDiscTienLC("Jungle");
7          DigitalVideoDiscTienLC cinderellaDVD = new DigitalVideoDiscTienLC("Cinderella");
8          swapTienLC(jungleDVD, cinderellaDVD);
9          System.out.println("jungle dvd title: " + jungleDVD.getTitle());
10         System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());
11         changeTitleTienLC(jungleDVD, cinderellaDVD.getTitle());
12         System.out.println("jungle dvd title: " + jungleDVD.getTitle());
13     }
14     public static void swapTIENLC(Object o1, Object o2){
15         Object tmp = o1;
16         o1 = o2;
17         o2 = tmp;
18     }
19     public static void changeTitletienlc(DigitalVideoDiscTienLC dvd, String title){
20         String oldTitle = dvd.getTitle();
21         dvd.setTitle(title);
22         dvd = new DigitalVideoDiscTienLC(oldTitle);
23     }

```

Figure 7 Passing parameter code

## Result

```

run:
jungle dvd title: Jungle
cinderella dvd title: Cinderella
jungle dvd title: Cinderella
BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 8 Result

## ➔ Có vấn đề với hàm swap

Trả lời câu hỏi:

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

-> Trả lời: Sau khi thực hiện phương thức swap(jungleDVD, cinderellaDVD), tiêu đề của hai đối tượng vẫn giữ nguyên vì trong Java, tham số được truyền vào phương thức là giá trị của đối tượng, không phải là tham chiếu đến đối tượng. Khi ta thay đổi giá trị của tham số bên trong phương thức (như việc đổi chỗ giữa o1 và o2), sự thay đổi này không ảnh hưởng đến giá trị của các đối tượng gốc.

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

-> Trả lời: Sau khi gọi changeTitle(jungleDVD, cinderellaDVD.getTitle()), tiêu đề của jungleDVD bị thay đổi vì trong phương thức changeTitle, ta thực hiện thay đổi trực tiếp trên đối tượng dvd (được truyền vào phương thức) bằng cách gọi dvd.setTitle(title). Điều này ảnh hưởng trực tiếp đến đối tượng gốc được truyền vào phương thức.

Sửa lại hàm **swap**:

```
public static void swapTienLC(DigitalVideoDiscTienLC dvd1, DigitalVideoDiscTienLC dvd2) {  
    String tmpTitle = dvd1.getTitle();  
    dvd1.setTitle(dvd2.getTitle());  
    dvd2.setTitle(tmpTitle);  
  
    String tmpCategory = dvd1.getCategory();  
    dvd1.setCategory(dvd2.getCategory());  
    dvd2.setCategory(tmpCategory);  
  
    String tmpDirector = dvd1.getDirector();  
    dvd1.setDirector(dvd2.getDirector());  
    dvd2.setDirector(tmpDirector);  
  
    int tmpLength = dvd1.getLength();  
    dvd1.setLength(dvd2.getLength());  
    dvd2.setLength(tmpLength);  
  
    float tmpCost = dvd1.getCost();  
    dvd1.setCost(dvd2.getCost());  
    dvd2.setCost(tmpCost);  
}  
  
public static void changeTitleTienLC(DigitalVideoDiscTienLC dvd, String title) {  
    dvd.setTitle(title);  
}  
}
```

Figure 9 Swap

Kết quả sau khi sửa:

```
run:  
jungle dvd title: Cinderella  
cinderella dvd title: Jungle  
jungle dvd title: Jungle  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 10 Result

### 3. Class Member and Instance Member

## Code

```

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

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

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

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

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

    public String getTitle(){ return title; }
    public void setTitle(String title){
        this.title = title;
    }
    public String getCategory(){ return category; }
    public void setCategory(String category){
        this.category = category;
    }
    public float getCost(){ return cost; }
    public void setCost(float cost){
        this.cost = cost;
    }
    public String getDirector(){ return director; }
    public void setDirector(String director){
        this.director = director;
    }
    public int getLength(){ return length; }
    public void setLength(int length){
        this.length = length;
    }
    public int getID(){ return id; }
}

```

Figure 11 Class member and Instance member



## Result

```

run:

The DVD "Tom and Jerry" has been added!
The DVD "The Lion king" has been added!
The DVD "Toy Story" has been added!
*****CART*****

Ordered items:
1.[Tom and Jerry] - [Cartoon] - [Gene Deitch] - [120]: 20.05 $
2.[The Lion king] - [anime] - [Roger Allers] - [87]: 15.55 $
3.[Toy Story] - [anime] - [John Lasseter] - [150]: 17.75 $
total: 53.35 $
*****

BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 12 Result

## 4. Open the Cart class

### Code

- Sửa code (Thêm hàm print trong cart.java)

```

public void print(){
    StringBuilder output = new StringBuilder("*****CART***** \nOrdered items: \n");
    for(int i = 0; i < qtyOrdered; i++){
        output.append(i+1 + ".[" + itemsOrdered[i].getTitle() + "] - [" + itemsOrdered[i].getCategory() + "] - [" +
            itemsOrdered[i].getDirector() + "] - [" + itemsOrdered[i].getLength() + "]: " +
            itemsOrdered[i].getCost() + " $\n");
    }
    output.append("total: ").append(totalCost()).append(" $\n");
    output.append("*****\n");
    System.out.println(output);
}

```

Figure 13 Code method to print the content in Cart

```

public void searchById(int i){
    if(i > qtyOrdered){
        System.out.println("No match found.");
    }
    else{
        System.out.println("Result: " + "[" + itemsOrdered[i-1].getTitle() +
            "] - [" + itemsOrdered[i-1].getCategory() + "] - [" +
            itemsOrdered[i-1].getDirector() + "] - [" +
            itemsOrdered[i-1].getLength() + "]: " + itemsOrdered[i-1].getCost() + " $\n");
    }
}

```

Figure 14 Code method search by id

```

public void searchByTitle(String title){
    for(int i = 0; i < qtyOrdered; i++){
        if(itemsOrdered[i].getTitle().equals(title)){
            System.out.println("Result: " + "[" + itemsOrdered[i].getTitle() +
                "]" - "[" + itemsOrdered[i].getCategory() + "]" - "["
                + itemsOrdered[i].getDirector() + "]" - "["
                + itemsOrdered[i].getLength() + "]: " + itemsOrdered[i].getCost() + " $" + "\n");
            return;
        }
    }
    System.out.println("No match found.");
}
}

```

Figure 15 Code method search by title

```

package hust.soict.dsai.test.cart;
import hust.soict.dsai.aimsproject.disc.DigitalVideoDiscTienLC;
import hust.soict.dsai.aimsproject.cart.CartTienLC;
public class TestCart {
    public static void main(String[] args) {
        CartTienLC anOrder = new CartTienLC();

        //Them dia vao gio hang
        DigitalVideoDiscTienLC dvd1 = new DigitalVideoDiscTienLC("Tom and Jerry", "Cartoon", 20.05f, "Gene Deitch", 120);
        anOrder.addDigitalVideoDisc(dvd1);
        DigitalVideoDiscTienLC dvd2 = new DigitalVideoDiscTienLC("The Lion king", "anime", 15.55f, "Roger Allers", 87);
        anOrder.addDigitalVideoDisc(dvd2);
        DigitalVideoDiscTienLC dvd3 = new DigitalVideoDiscTienLC("Toy Story", "anime", 17.75f, "John Lasseter", 150);
        anOrder.addDigitalVideoDisc(dvd3);
        anOrder.print();

        //Test search by ID method
        anOrder.searchById(3);
        anOrder.searchById(1);

        //Test search by Title method
        anOrder.searchByTitle("The Lion king");
    }
}

```

Figure 16 CartTest class

## Result

```

run:
The DVD "Tom and Jerry" have been added.
The DVD "The Lion king" have been added.
The DVD "Toy Story" have been added.
*****CART*****
Ordered items:
1.[Tom and Jerry] - [Cartoon] - [Gene Deitch] - [120]: 20.05 $
2.[The Lion king] - [anime] - [Roger Allers] - [87]: 15.55 $
3.[Toy Story] - [anime] - [John Lasseter] - [150]: 17.75 $
total: 53.35 $
*****

Result: [Toy Story] - [anime] - [John Lasseter] - [150]: 17.75 $

Result: [Tom and Jerry] - [Cartoon] - [Gene Deitch] - [120]: 20.05 $

Result: [The Lion king] - [anime] - [Roger Allers] - [87]: 15.55 $

BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 17 Result

## 5. Implement the Store class

### Code

```

2 package hust.soict.dsai.aimsproject.store;
3 import hust.soict.dsai.aimsproject.disc.DigitalVideoDiscTienLC;
4 import java.util.LinkedList;
5
6 public class StoreTienLC {
7     private LinkedList<DigitalVideoDiscTienLC> itemsInStore = new LinkedList<DigitalVideoDiscTienLC>();
8
9     private boolean checkDVD(DigitalVideoDiscTienLC disc){
10         for(DigitalVideoDiscTienLC DigitalVideoDisc : itemsInStore){
11             if (DigitalVideoDisc.equals(disc)){
12                 return true;
13             }
14         }
15         return false;
16     }
17
18     public void addDVD(DigitalVideoDiscTienLC disc){
19         if(!checkDVD(disc)){
20             itemsInStore.add(disc);
21             System.out.println( disc.getTitle() + " 've been added to the store !");
22         }
23         else{
24             System.out.println(disc.getTitle() + " 'already exists in the store.");
25         }
26     }
27
28     public void removeDVD(DigitalVideoDiscTienLC disc){
29         if(checkDVD(disc)){
30             itemsInStore.remove(disc);
31             System.out.println(disc.getTitle() + " 've been deleted from the store.");
32         }
33         else{
34             System.out.println("There is no "+ disc.getTitle() + " in the store.");
35         }
36     }
37
38     public void printStore(){
39         System.out.println("\n--- Store ---");
40         if(itemsInStore.isEmpty()){
41             System.out.println("No Dvds in store.");
42         }
43         else{
44             itemsInStore.forEach(dvd -> System.out.println(dvd));
45         }
46         System.out.println("-----");
47     }
48 }

```

Figure 18 Store Class

```

2 package hust.soict.dsai.test.store;
3
4 import hust.soict.dsai.aimsproject.disc.DigitalVideoDiscTienLC;
5 import hust.soict.dsai.aimsproject.store.StoreTienLC;
6 public class TestStore {
7     public static void main(String[] args) {
8         StoreTienLC store = new StoreTienLC();
9         DigitalVideoDiscTienLC dvd1 = new DigitalVideoDiscTienLC("Tom and Jerry", "Cartoon", 20.05f, "Gene Deitch", 120);
10        DigitalVideoDiscTienLC dvd2 = new DigitalVideoDiscTienLC("The Lion king", "anime", 15.55f, "Roger Allers", 87);
11        DigitalVideoDiscTienLC dvd3 = new DigitalVideoDiscTienLC("Toy Story", "anime", 17.75f, "John Lasseter", 150);
12
13        store.addDVD(dvd1);
14        store.addDVD(dvd2);
15        store.addDVD(dvd3);
16
17        System.out.println(store.toString());
18
19        store.removeDVD(dvd1);
20        store.removeDVD(dvd2);
21
22        System.out.println(store.toString());
23    }
24 }

```

Figure 19 StoreTest class

## Result

```

run:
Tom and Jerry 've been added to the store !
The Lion king 've been added to the store !
Toy Story 've been added to the store !
*****STORE*****
items in the store:
Tom and Jerry - 20.05 $
The Lion king - 15.55 $
Toy Story - 17.75 $
*****
Tom and Jerry 've been deleted from the store.
The Lion king 've been deleted from the store.
*****STORE*****
items in the store:
Toy Story - 17.75 $
*****

```

Figure 20 Result

## 6. String, StringBuilder and StringBuffer

## Code

```

2 package hust.soict.dsai.garbage;
3 import java.util.Random;
4 public class ConcatenationInLoops {
5     public static void main(String[] args) {
6         Random rTienLC = new Random(123);
7         // Phép nối chuỗi bằng String
8         long start = System.currentTimeMillis();
9         String s = "";
10        for (int i = 0; i < 65536; i++) {
11            s += rTienLC.nextInt(2);
12        }
13        System.out.println("Time taken with String concatenation: " + (System.currentTimeMillis() - start) + "ms");
14
15        System.out.println("Final String length: " + s.length());
16
17        // Phép nối chuỗi bằng StringBuilder
18        rTienLC = new Random(123); // Khởi tạo lại Random để kết quả giống nhau
19        start = System.currentTimeMillis();
20        StringBuilder sb = new StringBuilder();
21        for (int i = 0; i < 65536; i++) {
22            sb.append(rTienLC.nextInt(2)); // Thêm giá trị vào StringBuilder
23        }
24        String finalString = sb.toString();
25        System.out.println("Time taken with StringBuilder: " + (System.currentTimeMillis() - start) + "ms");
26
27        System.out.println("Final String length (StringBuilder): " + finalString.length());
28    }
29 }

```

Figure 21 So sánh String và StringBuilder

## Result

```

1 package hust.soict.dsai.garbage;
2 import java.io.IOException;
3 import java.nio.file.Files;
4 import java.nio.file.Paths;
5
6 public class GarbageCreator {
7     public static void main(String[] args) {
8         try {
9             String filename = "C:\\Users\\ASUS\\Desktop\\LyCongTien-LabOOP\\Lab03\\OtherProject\\aims\\src\\hust\\soict\\dsai\\garbage\\file.txt";
10
11             byte[] inputBytes = Files.readAllBytes(Paths.get(filename));
12             long startTime = System.currentTimeMillis();
13             // Xử lý dữ liệu từ file
14             StringBuilder outputString = new StringBuilder();
15             for (byte b : inputBytes) {
16                 outputString.append((char) b);
17             }
18             long endTime = System.currentTimeMillis();
19             // In thời gian xử lý
20             System.out.println("Processing time: " + (endTime - startTime) + " ms");
21         } catch (IOException e) {
22             // Thông báo lỗi đọc file
23             System.err.println("Error reading file: " + e.getMessage());
24         } catch (Exception e) {
25             // Thông báo các lỗi khác
26             System.err.println("An unexpected error occurred: " + e.getMessage());
27         }
28     }
29 }

```

Figure 22 GarbageCreator class

```

package hust.soict.dsai.garbage;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

public class NoGarbage {
    public static void main(String[] args) {
        String filename = "C:\\Users\\ASUS\\Desktop\\LyCongTien-Lab00P\\Lab03\\OtherProject\\aims\\src\\hust\\soict\\dsai\\garbage\\file.txt";
        byte[] inputBytes;
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
            // Bắt đầu đo thời gian thực hiện
            startTime = System.currentTimeMillis();

            StringBuilder outputStringBuilder = new StringBuilder();
            for (byte b : inputBytes) {
                outputStringBuilder.append((char) b);
            }
            // Kết thúc đo thời gian thực hiện
            endTime = System.currentTimeMillis();
            System.out.println("Execution time: " + (endTime - startTime) + " ms");
            // Xử lý ngoại lệ nếu có lỗi xảy ra khi đọc file
        } catch (IOException e) {
            System.err.println("Error reading file: " + filename);
            System.err.println("Reason: " + e.getMessage());
        }
    }
}

```

Figure 23 NoGarbage class

```

run:
Execution time: 0 ms
BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 24 Result: Program finishes quickly

## 7. Class Diagram

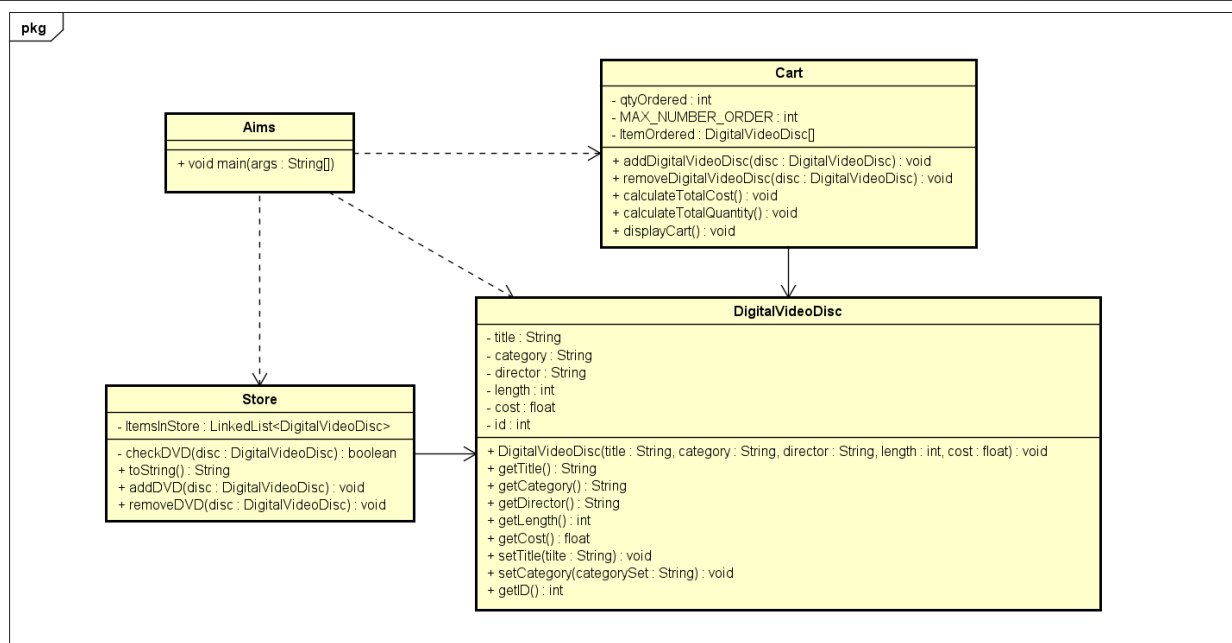


Figure 25 Class Diagram

## Table of Figures

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList).....	3
Figure 2 Aims Class.....	3
Figure 3 Result.....	4
Figure 4 Method addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) .....	4
Figure 5 Aims Class.....	5
Figure 6 Result.....	5
Figure 7 Passing parameter code.....	6
Figure 8 Result.....	6
Figure 9 Swap .....	7
Figure 10 Result.....	7
Figure 11 Class member and Instance member.....	8
Figure 12 Result.....	9
Figure 13 Code method to print the content in Cart .....	9
Figure 14 Code method search by id.....	9
Figure 15 Code method search by title.....	10
Figure 16 CartTest class.....	10
Figure 17 Result.....	10
Figure 18 Store Class .....	11
Figure 19 StoreTest class .....	12
Figure 20 Result.....	12
Figure 21 So sánh String và StringBuilder .....	13
Figure 22 GarbageCreator class .....	13
Figure 23 NoGarbage class .....	14
Figure 24 Result: Program finishes quickly.....	14
Figure 25 Class Diagram.....	14