

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



**BÁO CÁO THỰC HÀNH LAB 03**  
**Học phần: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

Mã học phần: IT3103

Mã lớp: 744530

Giảng viên hướng dẫn:

GV Lê Thị Hoa

Sinh viên thực hiện:

Tạ Hồng Phúc

MSSV:

20225906

Hà Nội, tháng 11 năm 2024

# BÁO CÁO THỰC HÀNH LAB 03

## THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### 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
Result:.....	4
2. Passing parameter.....	5
Code:.....	5
Result: .....	5
Trả lời câu hỏi: .....	5
Hàm Swap hoàn chỉnh:.....	5
Result: .....	6
3. Class Member and Instance Member .....	6
Code.....	6
Result: .....	7
4. Open the Cart class.....	7
5. Implement Store .....	9
6. String, StringBuilder, StringBuffer .....	10
7 Class Diagram.....	12
Table of Figures .....	13

## 1. Working with method overloading

### 1.1. Overloading by differing types of parameter

**Code:** Overloading method by list of parameter

```
//Insert list of DVDs
public void addDigitalVideoDisc_phuc(DigitalVideoDisc_phuc [] disc) {
    for (DigitalVideoDisc_phuc phuc:disc) {
        addDigitalVideoDisc_phuc(phuc);
    }
}
```

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList)

```
//Insert arbitrary numbers of arguments
public void addDigitalVideoDisc_phuc_1(DigitalVideoDisc_phuc ... disc) {
    for (DigitalVideoDisc_phuc phuc:disc) {
        addDigitalVideoDisc_phuc(phuc);
    }
}
```

Figure 2 Method to add arbitrary number of arguments

```
//add 1 dvd
anOrder.addDigitalVideoDisc_phuc(dvd1);
//add a list of dvds
anOrder.addDigitalVideoDisc_phuc(phuc123);
//add arbitrary number of dvds
anOrder.addDigitalVideoDisc_phuc_1(dvd4, dvd5);
```

Figure 3 Aims Class

**Result:** Kết quả khi chạy chương trình

```
"Mất biếc" has been added successfully
"Bố già" has been added successfully
"Despicable Me 3" has been added successfully
"Rush Hour" has been added successfully
"Ngày xưa có một chuyện tình" has been added successfully

Total cost is: 100.0$
```

Figure 4 Result

**Nhận xét:** Do khi truyền không biết trước số lượng đầu vào, Java coi như truyền một mảng giống như khi truyền danh sách DVD nên không thể có 2 phương thức Overloading như Figure 1 và Figure 2 cùng lúc.

## 1.2. Overloading by differing the number of parameters

Code:

```
//Overloading by differing in number of parameters
public void addDigitalVideoDisc_phucth(
    DigitalVideoDisc_phucth dvd1,
    DigitalVideoDisc_phucth dvd2) {
    addDigitalVideoDisc_phucth_1(dvd1, dvd2);
}
```

Figure 5 Method addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2)

```
//add 1 dvd
anOrder.addDigitalVideoDisc_phucth(dvd1);
//add 2 dvd
anOrder.addDigitalVideoDisc_phucth(dvd2, dvd3);
```

Figure 6 Aims Class

Result:

```
"Mắt biếc" has been added successfully
"Bố già" has been added successfully
"Despicable Me 3" has been added successfully

Total cost is: 60.0$
```

Figure 7 Result

## 2. Passing parameter

Code:

```
public class TestPassingParameter {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        DigitalVideoDisc_phuchth jungleDVD= new DigitalVideoDisc_phuchth("Jungle");
        DigitalVideoDisc_phuchth cinderellaDVD= new DigitalVideoDisc_phuchth("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_phuchth dvd, String title) {
        String oldTitle = dvd.getTitle();
        dvd.setTitle(title);
        dvd = new DigitalVideoDisc_phuchth (oldTitle);
    }
}
```

Figure 8 Passing parameter code

Result:

```
jungle DVD title: Jungle
cinderella DVD title: Cinderella
jungle DVD title: Cinderella
```

Figure 9 Passing parameter result

Trả lời câu hỏi:

- Khi gọi hàm swap, chương trình chỉ truyền giá trị title của 2 Object và thực hiện trao đổi nội bộ trong hàm swap. Ở đây, 2 giá trị title được copy rồi mới truyền vào hàm swap, gọi là **truyền tham trị (pass by value)**.
- Khi gọi hàm changeTitle, chương trình lấy giá trị của cinderellaDVD và gán trực tiếp vào địa chỉ tiêu đề của jungleDVD đang được tham chiếu tới. Vì vậy tiêu đề của jungleDVD thay đổi thành "Cinderella". Đây còn được gọi là **truyền tham chiếu (pass by reference)**.

Hàm Swap hoàn chỉnh:

```
public static void swap_1(DigitalVideoDisc_phuchth dvd0, DigitalVideoDisc_phuchth dvd1) {
    DigitalVideoDisc_phuchth tmp = new DigitalVideoDisc_phuchth("Jungle");
    tmp.setTitle(dvd0.getTitle());
    dvd0.setTitle(dvd1.getTitle());
    dvd1.setTitle(tmp.getTitle());
}
```

Figure 10 New Swap Function

Result:

```
jungle DVD title: Cinderella
cinderella DVD title: Jungle
```

Figure 11 Swap Function Result

- Với các thuộc tính khác, ta làm tương tự với cách swap thuộc tính Title.
- **Kết luận:** Java luôn là ngôn ngữ truyền tham trị (pass by value). Với các kiểu dữ liệu nguyên thủy như int, float, boolean, ... Java sẽ tạo bản sao giá trị rồi truyền vào hàm. Với các Object, tuy thuộc tính có thể thay đổi nhưng việc gán tham chiếu trong hàm sẽ không thay đổi đến bản gốc.

### 3. Class Member and Instance Member

Code

```
private static int nbDVD_phucth = 0;
private int id;
public int getId() {
    return id;
}
```

```
//Create DVD by Title
public DigitalVideoDisc_phucth(String title) {
    super();
    this.title = title;
    nbDVD_phucth++;
    this.id = nbDVD_phucth;
}
//Create DVD by Title, Category, Cost
public DigitalVideoDisc_phucth(String title, String category, float cost) {
    super();
    this.title = title;
    this.category = category;
    this.cost = cost;
    nbDVD_phucth++;
    this.id = nbDVD_phucth;
}
//Create DVD by Title, Category, Director, Cost
public DigitalVideoDisc_phucth(String title, String category, String director, float cost) {
    super();
    this.title = title;
    this.category = category;
    this.director = director;
    this.cost = cost;
    nbDVD_phucth++;
    this.id = nbDVD_phucth;
}
//Create DVD by Title, Category, Director, Length, Cost
public DigitalVideoDisc_phucth(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;
    nbDVD_phucth++;
    this.id = nbDVD_phucth;
}
}
```

Figure 12 Class member and Instance member

- Ta thêm thuộc tính id cùng hàm getter mà không có setter vì id sẽ được gán từ đầu
- Khởi tạo nbDVD\_phucth = 0, tăng 1 đơn vị rồi bắt đầu gán cho các object qua tất cả các phương thức DigitalVideoDisc\_phucth

Result:

```
1 - "Mắt biếc" has been added successfully
2 - "Bố già" has been added successfully
3 - "Despicable Me 3" has been added successfully

Total cost is: 60.0$
```

Figure 13 Result

## 4. Open the Cart class

```
// print the list of ordered items of a cart
public void printItemsInCart() {
    System.out.println("*****CART*****");
    System.out.println("Ordered Items:");
    for(int i=0; i<qtyOrdered; i++) {
        DigitalVideoDisc_phuc dvd = itemsOrdered[i];
        System.out.printf("%d. DVD - %s - %s - %s - %d: %.2f$ \n", i+1, dvd.getTitle(), dvd.getCategory(), dvd.getDirector(), dvd.getLength(), dvd.getCost() );
    }

    float totalCost = totalCost();
    System.out.printf("Total cost: %.2f $ \n", totalCost);
    System.out.println("*****");
}
```

```
1 - "Mắt biếc" has been added successfully
2 - "Bố già" has been added successfully
3 - "Despicable Me 3" has been added successfully
*****CART*****
Ordered Items:
1. DVD - Mắt biếc - Tình cảm - Victor Vũ - 120: 20.50$
2. DVD - Bố già - Hà - Thành Trần - 87: 24.15$
3. DVD - Despicable Me 3 - Animation - null - 0: 15.35$
Total cost: 60.00 $
*****
Total cost is: 60.0$
```

Figure 14 Code method to print the content in Cart

```
// search by ID
public boolean searchById(int id) {
    int n = qtyOrdered;
    for(int i=0; i<n; i++) {
        if(itemsOrdered[i].getId() == id) return true;
    }
    return false;
}
```

Figure 15 Code method search by id

```
//search by title
public void searchByTitle(String title) {
    int n = qtyOrdered;
    boolean check = false;
    for(int i=0; i<n; i++) {
        if(itemsOrdered[i].isMatch(title)) {
            System.out.println("Your cart has a DVD as title name is: " + title);
            check = true;
        }
    }
    if(!check) {
        System.out.println("No data is matching");
    }
}
```

Figure 16 Code method search by title

```
// Check Title
public boolean isMatch(String title) {
    if(this.title == null) {
        return false;
    }
    return this.title.equalsIgnoreCase(title);
}
```

Figure 17 IsMatch() in DigitalVideoDisc class

Result:

```
public class CartTest_phuchth {
    public static void main(String[] args) {
        Cart_phuchth ordered = new Cart_phuchth();

        //create new DVD and add to cart
        DigitalVideoDisc_phuchth dvd1 =
            new DigitalVideoDisc_phuchth("Mắt biếc", "Tình cảm", "Victor Vũ", 120, 20.5f);
        DigitalVideoDisc_phuchth dvd2 =
            new DigitalVideoDisc_phuchth("Bố già", "Hài", "Thánh Trần", 87, 24.15f);
        DigitalVideoDisc_phuchth dvd3 =
            new DigitalVideoDisc_phuchth("Despicable Me 3", "Animation", 15.35f);
        DigitalVideoDisc_phuchth dvd4 =
            new DigitalVideoDisc_phuchth("Rush Hour", "Acation", 20f);
        DigitalVideoDisc_phuchth dvd5 =
            new DigitalVideoDisc_phuchth("Ngày xưa có một chuyện tình", "Tình cảm", 20f);

        ordered.addDigitalVideoDisc_phuchth_1(dvd1, dvd2, dvd3, dvd4, dvd5);

        // Tìm kiếm theo ID
        if(ordered.searchById(2)) {
            System.out.println("Items has id = 2 is ordered");
        } else {
            System.out.println("No data");
        }
        // tìm kiếm DVD theo tên
        ordered.searchByTitle("Rush Hour");
    }
}
```

Figure 18 Cart Test

```
1 - "Mắt biếc" has been added successfully
2 - "Bố già" has been added successfully
3 - "Despicable Me 3" has been added successfully
4 - "Rush Hour" has been added successfully
5 - "Ngày xưa có một chuyện tình" has been added successfully
Items has id = 2 is ordered
Your cart has a DVD as title name is: Rush Hour
```

Figure 19 Result



## 5. Implement Store

### Code

```
public class Store_phuchth {
    public static final int MAX_DVD = 50;
    private DigitalVideoDisc_phuchth itemsInStore[] = new DigitalVideoDisc_phuchth[MAX_DVD];
    private int qtyOrdered = 0;

    public int getQtyOrdered() {
        return qtyOrdered;
    }

    public void setQtyOrdered(int n) {
        qtyOrdered += n;
    }

    public void addDVDStore(DigitalVideoDisc_phuchth disc) {
        if(this.getQtyOrdered() <= 50) {
            itemsInStore[this.getQtyOrdered()] = disc;
            setQtyOrdered(1);
            System.out.println("DVD " + "'" + disc.getTitle() + "'" + " has been added in phuchth");
        } else {
            System.out.println("The phuchth store is almost full");
        }
    }

    public void removeDVDStore(DigitalVideoDisc_phuchth disc) {
        for(int i=0; i<qtyOrdered; i++) {
            if(itemsInStore[i].getTitle() == disc.getTitle()) {
                for(int j=i; j<qtyOrdered; j++) {
                    itemsInStore[j] = itemsInStore[j+1];
                }
                i--;
                setQtyOrdered(-1);
                System.out.println("The DVD " + disc.getTitle() + " has been removed in phuchth ");
            }
        }
    }
}
```

Figure 20 Store Class

```
public class StoreTest_phuchth {
    public static void main(String[] args) {
        Store_phuchth myStore_phuchth = new Store_phuchth();

        DigitalVideoDisc_phuchth dvd1 =
            new DigitalVideoDisc_phuchth("Mắt biếc", "Tình cảm", "Victor Vũ", 120, 20.5f);
        DigitalVideoDisc_phuchth dvd2 =
            new DigitalVideoDisc_phuchth("Bố già", "Hài", "Thánh Trần", 87, 24.15f);
        DigitalVideoDisc_phuchth dvd3 =
            new DigitalVideoDisc_phuchth("Despicable Me 3", "Animation", 15.35f);
        DigitalVideoDisc_phuchth dvd4 =
            new DigitalVideoDisc_phuchth("Rush Hour", "Acation", 20f);
        DigitalVideoDisc_phuchth dvd5 =
            new DigitalVideoDisc_phuchth("Ngày xưa có một chuyện tình", "Tình cảm", 20f);

        myStore_phuchth.addDVDStore(dvd1);
        myStore_phuchth.addDVDStore(dvd2);
        myStore_phuchth.addDVDStore(dvd3);
        myStore_phuchth.addDVDStore(dvd4);
        myStore_phuchth.addDVDStore(dvd5);
        myStore_phuchth.removeDVDStore(dvd3);
    }
}
```

Figure 21 StoreTest Class

## Result

```
DVD "Mất biếc" has been added in phucth
DVD "Bố già" has been added in phucth
DVD "Despicable Me 3" has been added in phucth
DVD "Rush Hour" has been added in phucth
DVD "Ngày xưa có một chuyện tình" has been added in phucth
The DVD Despicable Me 3 has been removed in phucth
```

Figure 22 Result

## 6. String, StringBuilder, StringBuffer

Ba lớp **ConcatenationInLoops**, **GarbageCreator**, và **NoGarbage** được xây dựng nhằm thử nghiệm hiệu suất của việc nối chuỗi bằng toán tử '+' so với việc sử dụng **StringBuilder**. Mỗi lớp có nhiệm vụ riêng và minh họa hiệu suất theo các cách khác nhau:

- **ConcatenationInLoops**: Lớp này tạo chuỗi bằng cách nối các số ngẫu nhiên 0 và 1. Phần đầu tiên sử dụng toán tử + để nối chuỗi trong vòng lặp và đo thời gian thực thi. Phần thứ hai dùng **StringBuilder** để nối chuỗi và sau đó chuyển kết quả về kiểu **String**. Kết quả thử nghiệm cho thấy việc sử dụng **StringBuilder** nhanh hơn nhiều so với toán tử + khi nối chuỗi trong vòng lặp.

```
package lab02;
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 = "/Users/andrew_ta/Andrew-Code/IT3103.744530.2024.1.20225906.TaHongPhuc/lab03/AimsProject_phucth_2/bin/lab02/test.txt";
        byte[] inputBytes = {};
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
            startTime = System.currentTimeMillis();
            StringBuilder sb = new StringBuilder();
            for (byte b : inputBytes) {
                sb.append((char) b);
            }
            String outputString = sb.toString();
            endTime = System.currentTimeMillis();
            System.out.println(endTime - startTime);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
206
2
```

Figure 23 So sánh String và StringBuilder

- **GarbageCreator:** Lớp này tạo chuỗi từ nội dung của một tệp. Bạn đọc toàn bộ nội dung từ tệp "test.txt" và nối từng byte thành một chuỗi bằng toán tử '+'. Sau đó, đo thời gian thực hiện để đánh giá hiệu suất của việc nối chuỗi với toán tử '+'.

```
package lab02;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

public class GarbageCreator {
    public static void main(String[] args) throws IOException {
        String filename = "/Users/andrew_ta/Andrew-Code/IT3103.744530.2024.1.20225906.TaHongPhuc/lab03/AimsProject_phuc_th_2/bin/lab02/test.txt";
        byte[] inputBytes = {};
        long startTime, endTime;

        inputBytes = Files.readAllBytes(Paths.get(filename));
        startTime = System.currentTimeMillis();
        String outputString = "";
        for (byte b : inputBytes) {
            outputString += (char) b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}
```

Figure 24 Garbage Creator Class

6

Figure 25 Result: Program hangs

- **NoGarbage:** Đây là phiên bản cải tiến của GarbageCreator, được thiết kế để tránh việc tạo ra rác bộ nhớ. Bạn đọc nội dung từ tệp "test.txt" và nối từng byte thành chuỗi bằng StringBuilder, rồi chuyển kết quả thành kiểu String. Kết quả thử nghiệm cho thấy việc sử dụng StringBuilder không tạo ra rác bộ nhớ thừa và giúp cải thiện hiệu suất.

```
package lab02;
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 = "/Users/andrew_ta/Andrew-Code/IT3103.744530.2024.1.20225906.TaHongPhuc/lab03/AimsProject_phuc_th_2/bin/lab02/test.txt";
        byte[] inputBytes = {};
        long startTime, endTime;

        try {
            inputBytes = Files.readAllBytes(Paths.get(filename));
            startTime = System.currentTimeMillis();
            StringBuilder sb = new StringBuilder();
            for (byte b : inputBytes) {
                sb.append((char) b);
            }
            String outputString = sb.toString();
            endTime = System.currentTimeMillis();
            System.out.println(endTime - startTime);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Figure 26 NoGarbage Class

0

Figure 27 Result: Finishes quickly

Tóm lại, các lớp này thực hiện các thử nghiệm để chứng minh rằng việc sử dụng StringBuilder nhanh hơn so với toán tử + trong việc nối chuỗi, đặc biệt là khi thao tác với chuỗi trong vòng lặp. Lợi ích của StringBuilder đến từ khả năng thay đổi nội dung chuỗi mà không cần tạo nhiều bản sao của chuỗi, giúp tiết kiệm bộ nhớ và nâng cao hiệu suất.

## 7 Class Diagram

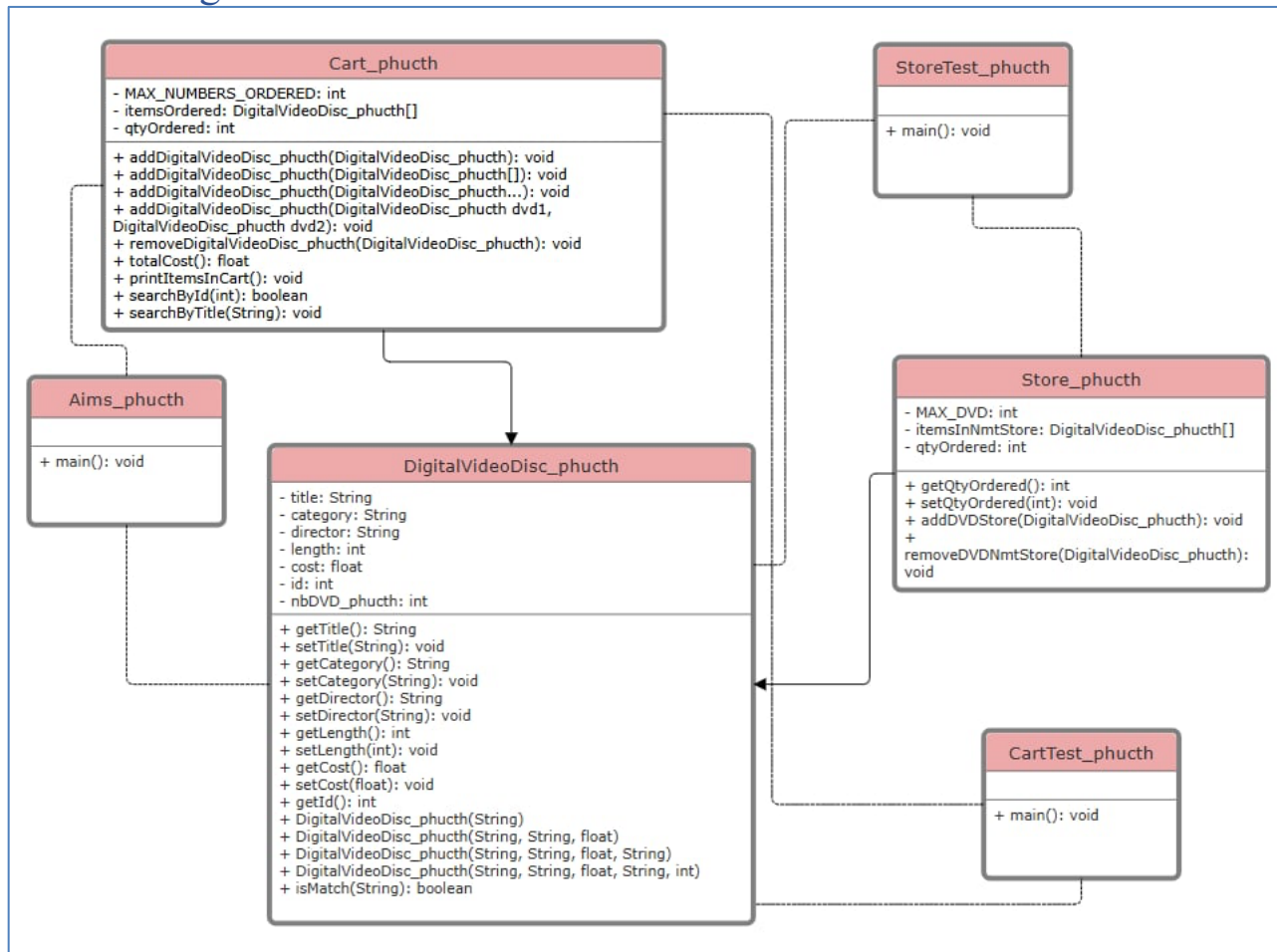


Figure 28 Class Diagram

## Table of Figures

Figure 1 Method addDigitalVideoDisc(DigitalVideoDisc [] dvdList).....	3
Figure 2 Method to add arbitrary number of arguments.....	3
Figure 3 Aims Class .....	3
Figure 4 Result .....	3
Figure 5 Method addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2).....	4
Figure 6 Aims Class .....	4
Figure 7 Result .....	4
Figure 8 Passing parameter code .....	5
Figure 9 Passing parameter result.....	5
Figure 10 New Swap Function.....	5
Figure 11 Swap Function Result .....	6
Figure 12 Class member and Instance member .....	6
Figure 13 Result .....	7
Figure 14 Code method to print the content in Cart .....	7
Figure 15 Code method search by id .....	7
Figure 16 Code method search by title .....	7
Figure 17 IsMatch() in DigitalVideoDisc class.....	8
Figure 18 Cart Test.....	8
Figure 19 Result .....	8
Figure 20 Store Class .....	9
Figure 21 StoreTest Class .....	9
Figure 22 Result .....	10
Figure 23 So sánh String và StringBuilder.....	10
Figure 24 Garbage Creator Class .....	11
Figure 25 Result: Program hangs .....	11
Figure 26 NoGarbage Class .....	11
Figure 27 Result: Finishes quickly.....	11
Figure 28 Class Diagram .....	12