

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



BÁO CÁO THỰC HÀNH – LAB03

IT3103-744528-2024.1

BÀI THỰC HÀNH - LAB03

Họ và tên sv : Nguyễn Trọng Hình

Mssv : 20225842

Lớp : Việt Nhật 04 – K67

GVHD : Lê Thị Hoa

HTGD : Đặng Mạnh Cường

Content

II . Working with method overloading	3
1 Overloading by differing types of parameter	3
2. Overloading by differing the number of parameters.....	3
III. Passing parameter	3
1 .Câu hỏi:	3
2. Tạo class TestPassingParameter	4
3. Giải quyết vấn đề trong đoạn code này	4
3.1 Trả lời câu hỏi.....	4
3.2 Sửa lại hàm swap.....	5
V. Classifier Member and Instance Member.....	5
VI . Open the Cart class	6
1. Sửa code (Thêm hàm print trong cart.java).....	6
2.Kết quả khi chạy	6
VII . Implement the Store class	7
1.Tạo lớp Store.....	7
2. Tạo lớp TestStore và kiểm tra kết quả	7
2.1 Tạo lớp TestStore	7
2.2 Kiểm tra kết quả	8
IX . String, StringBuilder and StringBuffer	8
1. Tạo lớp ConcatenationInLoops	8
2. Tạo lớp GarbageCreator	9
3. Tạo lớp NoGarbageCreator.....	9
X . Cập nhật các Diagram trong project AIMS.....	10
1. Class diagram.....	10

II . Working with method overloading

1 Overloading by differing types of parameter

Phương thức mới thứ hai addDigitalVideoDisc(DigitalVideoDisc... dvdArray)

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

2. Overloading by differing the number of parameters

```
//Ham them 2 dia DVD
public int addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    if (qtyOrdered + 1 >= MAX_NUMBERS_ORDERED) {
        System.out.println(x:"The cart is almost full. Can't add more discs");
        return 0;
    } else {
        dvds[qtyOrdered] = dvd1;
        qtyOrdered++;
        System.out.println("The DVD " + "\"" + dvd1.getTitle() + "\"" + " has been added!");

        dvds[qtyOrdered] = dvd2;
        qtyOrdered++;
        System.out.println("The DVD " + "\"" + dvd2.getTitle() + "\"" + " has been added!");

        return 2; //Tra ve so dia DVD da them duoc
    }
}
```

III. Passing parameter

1 .Câu hỏi:

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

Java là ngôn ngữ lập trình **"Pass by Value"** (truyền tham trị) .

Tuy nhiên, khi truyền đối tượng, giá trị truyền vào là **tham chiếu** đến đối tượng, không phải chính đối tượng đó. Vì vậy, bạn có thể thay đổi nội dung của đối tượng, nhưng không thể thay đổi tham chiếu của đối tượng (tức là không thể thay đổi đối tượng gốc trong phương thức).

2. Tạo class TestPassingParameter

```
package hust.soict.hedspi.test.disc;  
  
import hust.soict.hedspi.aims.disc.DigitalVideoDisc;  
  
public class TestPassingParameter {  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");  
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"Cinderella");  
        swap(jungleDVD, cinderellaDVD);  
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());  
        System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());  
        changeTitle(jungleDVD, cinderellaDVD.getTitle());  
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());  
    }  
  
    public static void swap(Object o1, Object o2) {  
        Object tmp = o1;  
        o1 = o2;  
        o2 = tmp;  
    }  
  
    public static void changeTitle(DigitalVideoDisc dvd, String title) {  
        String oldTitle = dvd.getTitle();  
        dvd.setTitle(title);  
        dvd = new DigitalVideoDisc(oldTitle);  
    }  
}
```

Kết quả khi chạy :

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

3. Giải quyết vấn đề trong đoạn code này

3.1 Trả lời câu hỏi

Đoạn code có một vấn đề lớn trong hàm swap, đó là việc swap (hoán đổi) các đối tượng o1 và o2 không có tác dụng bên ngoài hàm vì Java sử dụng **pass-by-value** cho tham số.

3.2 Sửa lại hàm swap

```
public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc 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);

    double tmpCost = dvd1.getCost();
    dvd1.setCost(dvd2.getCost());
    dvd2.setCost(tmpCost);
}

public static void changeTitle(DigitalVideoDisc dvd, String title) {
    dvd.setTitle(title);
}
}
```

Kết quả khi sửa chữa :

```
Jungle dvd title: Cinderella
Cinderella dvd title: Jungle
Jungle dvd title: Jungle
```

V. Classifier Member and Instance Member

```
public class DigitalVideoDisc {
    private static int nbDigitalVideoDiscs = 0;
    private String title;
    private String category;
    private String director;
    private int length = 0;
    private double cost;
    private int id; // Instance attribute for ID

    public String getTitle() {return title;}
    public String getCategory() {return category;}
    public String getDirector() {return director;}
    public int getLength() {return length;}
    public double getCost() {return cost;}
    public int getId() {return id;}

    // Constructor with title, category, and cost parameters
    public DigitalVideoDisc(String title, String category, double cost) {
        this(title, category, director:null, length:0, cost);
    }

    // Constructor with only title
    public DigitalVideoDisc(String title) {
        this(title, category:null, director:null, length:0, cost:0.0);
    }

    // Constructor with title, category, director, and cost parameters
    public DigitalVideoDisc(String title, String category, String director, double cost) {
        this(title, category, director, length:0, cost);
    }

    // Constructor with all parameters
    public DigitalVideoDisc(String title, String category, String director, int length, double cost) {
        this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
        this.id = ++nbDigitalVideoDiscs;
    }
}
```

```
// Setter methods
public void setTitle(String titleSet) {title = titleSet; }
public void setCategory(String categorySet) {category = categorySet; }
public void setDirector(String directorSet) {director = directorSet; }
public void setLength(int lengthSet) {length = lengthSet; }
public void setCost(double costSet) {cost = costSet; }

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append(str:"DVD Title: ").append(title).append(str:"\n")
      .append(str:"Category: ").append(category != null ? category : "N/A").append(str:"\n")
      .append(str:"Director: ").append(director != null ? director : "N/A").append(str:"\n")
      .append(str:"Length: ").append(length > 0 ? length + " minutes" : "N/A").append(str:"\n")
      .append(str:"Cost: $").append(cost).append(str:"\n")
      .append(str:"ID: ").append(id);

    return sb.toString();
}
}
```

VI. Open the Cart class

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

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

2. Kết quả khi chạy

Phương thức print() được gọi ở lớp TestCart

```
NTH-The DVD "One Piece" has been successfully added to the cart.
NTH-The DVD "Songoku" has been successfully added to the cart.
NTH-The DVD "Naruto" has been successfully added to the cart.
*****CART*****
Ordered items:
1. [One Piece] - [Anime] - [Oda] - [148]: 29.5 $
2. [Songoku] - [Action] - [Toriyama] - [136]: 19.0 $
3. [Naruto] - [Anime] - [Kishimoto] - [169]: 24.5 $
total: 73.0 $
*****
```

VII . Implement the Store class

1.Tạo lớp Store

```
AimsProject > src > hust > soict > dsai > aims > store > J Store.java > ...
1 // nguyentronghinh - 20225842
2 // class "Store"
3
4 package hust.soict.dsai.aims.store;
5
6 import hust.soict.dsai.aims.disc.DigitalVideoDisc;
7 import java.util.ArrayList;
8
9 public class Store {
10     private final ArrayList<DigitalVideoDisc> itemsInStore = new ArrayList<>();
11
12     public void addDigitalVideoDisc(DigitalVideoDisc dvd) {
13         itemsInStore.add(dvd);
14         System.out.println("The DVD \"" + dvd.getTitle() + "\" has been added.");
15     }
16
17     public void removeDigitalVideoDisc(DigitalVideoDisc dvd) {
18         if (itemsInStore.remove(dvd)) {
19             System.out.println("The DVD \"" + dvd.getTitle() + "\" has been removed.");
20         } else {
21             System.out.println("DVD \"" + dvd.getTitle() + "\" is not in the store.");
22         }
23     }
24
25     public void printStorelist() {
26         System.out.println(x: "\n--- Store ---");
27         if (itemsInStore.isEmpty()) {
28             System.out.println(x: "No DVDs in store.");
29         } else {
30             itemsInStore.forEach(dvd -> System.out.println(dvd));
31         }
32         System.out.println(x: "-----");
33     }
34 }
35
```

2. Tạo lớp TestStore và kiểm tra kết quả

2.1 Tạo lớp TestStore

```
// nguyentronghinh - 20225842
// class "TestStore"

package hust.soict.dsai.test.store;

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

public class TestStore {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {

        Store store = new Store();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"One Piece", category:"Anime", director:"Oda", length:148, cost:29.5f);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Songoku", category:"Action", director:"Toriyama", length:136, cost:19.0f);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Naruto", category:"Anime", director:"Kishimoto", length:169, cost:24.5f);

        // Thêm DVD
        store.addDigitalVideoDisc(dvd1);
        store.addDigitalVideoDisc(dvd2);
        store.addDigitalVideoDisc(dvd3);

        // In danh sách DVD
        store.printStorelist();

        // Xóa DVD
        store.removeDigitalVideoDisc(dvd1);
        store.removeDigitalVideoDisc(dvd2);
        store.removeDigitalVideoDisc(dvd3);

        // In lại danh sách DVD sau khi xóa
        store.printStorelist();
    }
}

```


2.2 Kiểm tra kết quả

```

The DVD "Songoku" has been added.
The DVD "Naruto" has been added.

--- Store ---
DVD Title: One Piece
Category: Anime
Director: Oda
Length: 148 minutes
Cost: $29.5
ID: 1
DVD Title: Songoku
Category: Action
Director: Toriyama
Length: 136 minutes
Cost: $19.0
ID: 2
DVD Title: Naruto
Category: Anime
Director: Kishimoto
Length: 169 minutes
Cost: $24.5
ID: 3
-----
The DVD "One Piece" has been removed.
The DVD "Songoku" has been removed.
The DVD "Naruto" has been removed.

--- Store ---
No DVDs in store.
-----

```

IX . String, StringBuilder and StringBuffer

1. Tạo lớp ConcatenationInLoops

```

package hust.soict.dsai.garbage;

import java.util.Random;

public class ConcatenationInLoops {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Random r = new Random(seed:123);

        // Phép nối chuỗi bằng String
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i++) {
            s += r.nextInt(bound:2);
        }
        System.out.println("Time taken with String concatenation: " + (System.currentTimeMillis() - start) + "ms");

        System.out.println("Final String length: " + s.length());

        // Phép nối chuỗi bằng StringBuilder
        r = new Random(seed:123); // Khởi tạo lại Random để kết quả giống nhau
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 65536; i++) {
            sb.append(r.nextInt(bound:2)); // Thêm giá trị vào StringBuilder
        }
        String finalString = sb.toString();
        System.out.println("Time taken with StringBuilder: " + (System.currentTimeMillis() - start) + "ms");

        System.out.println("Final String length (StringBuilder): " + finalString.length());
    }
}

```


2. Tạo lớp GarbageCreator

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

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

public class GarbageCreator {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        try {
            // Đường dẫn tới file
            String filename = "C:\\Users\\hinhnh\\\\.vscode\\LTHDT_Java\\Lab02_03_04_05_20225842_NguyenTrongHinh\\OtherProject\\src\\hust\\soict\\dsai\\garbage\\file.exe";

            // Đọc file thành mảng byte
            byte[] inputBytes = Files.readAllBytes(Paths.get(filename));

            long startTime = System.currentTimeMillis();

            // Xử lý dữ liệu từ file
            StringBuilder outputString = new StringBuilder();
            for (byte b : inputBytes) {
                outputString.append((char) b);
            }

            long endTime = System.currentTimeMillis();

            // In thời gian xử lý
            System.out.println("Processing time: " + (endTime - startTime) + " ms");
        } catch (IOException e) {
            // Thông báo lỗi đọc file
            System.err.println("Error reading file: " + e.getMessage());
        } catch (Exception e) {
            // Thông báo các lỗi khác
            System.err.println("An unexpected error occurred: " + e.getMessage());
        }
    }
}
```

3. Tạo lớp NoGarbageCreator

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

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

public class NoGarbage {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        String filename = "C:\\Users\\hinhnh\\\\.vscode\\LTHDT_Java\\Lab02_03_04_05_20225842_NguyenTrongHinh\\OtherProject\\src\\hust\\soict\\dsai\\garbage\\file.exe";
        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());
        }
    }
}
```

X . Cập nhật các Diagram trong project AIMS

1. Class diagram

