

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 và tên sv: **Đỗ Hoàng Đông**

MSSV: **20225807**

Lớp: **Việt Nhật 03 - K67**

GVHD: Lê Thị Hoa

HTGD: Đặng Mạnh Cường

Hà Nội 10/2024

Mục Lục

I. Tạo các branch trong project	4
1. Tạo branch con Lab03, Lab04, Lab05 trong branch cha trong main	4
2. Tạo 9 branch trong Lab03 theo yêu cầu.....	4
3. Trộn branch	4
4. Push lên github.....	4
II. Working with method overloading.....	4
1. Overloading by differing types of parameter.....	4
2. Overloading by differing the number of parameters	5
III. Passing parameter	6
1. Trả lời câu hỏi (Tiếng Việt)	6
2. Giải quyết vấn đề trong đoạn code này	7
a Trả lời 2 câu hỏi (Tiếng Việt)	7
b Sửa lại hàm swap.....	8
IV. Sử dụng Debug	8
V. Classifier Member and Instance Member	9
VI. Open the Cart class	10
1. Sửa code (Thêm hàm print , searchbyID, searchbyTitle trong cart.java)	10

2. Chạy thử và xem kết quả	11
 VII. Implement the Store class	11
1. Tạo lớp Store.....	11
2. Tạo lớp TestStore và kiểm thử.....	12
Tạo lớp TestStore.....	12
Kiểm thử.....	13
 VIII. Re-organize your projects.....	13
 IX. String, StringBuilder and StringBuffer	13
1. Tạo lớp ConcatenationInLoops.....	13
2. Tạo lớp GarbageCreator	13
3. Tạo lớp NoGarbage	14
 X. Release flow demonstration	14
 Cập nhật các Diagram trong project AIMS.....	14
1. Class diagram	14
2. Use-case diagram	15

I. Tạo các branch trong project

1. Tạo branch con Lab03, Lab04, Lab05 trong branch cha trong main
2. Tạo 9 branch trong Lab03 theo yêu cầu
3. Trộn branch
4. Push lên github

Cách push: Mở Git Gui, checkout all branch, sau đó chọn các branch rồi up

II. Working with method overloading

1. Overloading by differing types of parameter

- Phương thức mới đầu tiên addDigitalVideoDisc(DigitalVideoDisc[] dvdList) cho phép thêm một danh sách các đĩa DVD vào giỏ hàng.



```
27     }
28
29     public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
30         for (DigitalVideoDisc dvd : dvdList) {
31             if (dvd != null) { // Kiểm tra nếu DVD hợp lệ
32                 boolean added = false;
33                 for (int i = 0; i < MAX_NUMBERS_ORDERED; i++) {
34                     if (itemsOrdered[i] == null) {
35                         itemsOrdered[i] = dvd;
36                         System.out.println("Added DVD: " + dvd.getTitle());
37                         added = true;
38                         break;
39                     }
40                 }
41                 if (!added) {
42                     System.out.println("The cart is almost full");
43                 }
44             } else {
45                 System.out.println("Invalid DVD found in the list. Skipping...");
46             }
47         }
48     }
49 }
```

- Phương thức mới thứ hai addDigitalVideoDisc(DigitalVideoDisc... dvdArray) sử dụng varargs để cho phép thêm một số lượng tùy ý các đĩa DVD vào giỏ hàng.

```

    }

    public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
        if (dvd1 != null) {
            for (int i = 0; i < MAX_NUMBERS_ORDERED; i++) {
                if (itemsOrdered[i] == null) { // Tìm vị trí trống trong mảng
                    itemsOrdered[i] = dvd1;
                    System.out.println("Added DVD: " + dvd1.getTitle());
                    break;
                }
            }
        }

        if (dvd2 != null) {
            for (int i = 0; i < MAX_NUMBERS_ORDERED; i++) {
                if (itemsOrdered[i] == null) { // Tìm vị trí trống trong mảng
                    itemsOrdered[i] = dvd2;
                    System.out.println("Added DVD: " + dvd2.getTitle());
                    break;
                }
            }
        }
    }
}

```

=> Cách sử dụng varargs linh hoạt hơn vì nó cho phép thêm bất kỳ số lượng đĩa DVD nào mà không cần chỉ định mảng một cách rõ ràng. Điều này làm cho mã nguồn sạch sẽ hơn và thuận tiện hơn khi gọi phương thức. Vì thế em thích phương thức dùng varargs.

2. Overloading by differing the number of parameters

```

    }

    public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
        if (dvd1 != null) {
            for (int i = 0; i < MAX_NUMBERS_ORDERED; i++) {
                if (itemsOrdered[i] == null) { // Tìm vị trí trống trong mảng
                    itemsOrdered[i] = dvd1;
                    System.out.println("Added DVD: " + dvd1.getTitle());
                    break;
                }
            }
        }

        if (dvd2 != null) {
            for (int i = 0; i < MAX_NUMBERS_ORDERED; i++) {
                if (itemsOrdered[i] == null) { // Tìm vị trí trống trong mảng
                    itemsOrdered[i] = dvd2;
                    System.out.println("Added DVD: " + dvd2.getTitle());
                    break;
                }
            }
        }
    }
}

```

III. Passing parameter

1. Trả lời câu hỏi (Tiếng Việt)

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.

- *Tạo class TestPassingParameter*

```

    *
    * @author HP
    */
public class TestPassingParameter {
    public static void main(String[] args) {
        // Tạo các đối tượng DigitalVideoDisc
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title: "Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title: "Cinderella");

        // Gọi phương thức swap
        swap(o1: jungleDVD, o2: cinderellaDVD);
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
        System.out.println("cinderella dvd title: " + cinderellaDVD.getTitle());

        // Gọi phương thức changeTitle
        changeTitle(dvd: jungleDVD, title: cinderellaDVD.getTitle());
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
    }
    // Phương thức đổi chỗ hai đối tượng
    public static void swap(Object o1, Object o2) {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }
    // Phương thức thay đổi tiêu đề
}
```

```

        changeTitle(dvd:jungleDVD, title: cinderellaDVD.getTitle());
        System.out.println("jungle dvd title: " + jungleDVD.getTitle());
    }

    // Phương thức đổi chỗ hai đối tượng
    public static void swap(Object o1, Object o2) {
        Object tmp = o1;
        o1 = o2;
        o2 = tmp;
    }

    // Phương thức thay đổi tiêu đề
    public static void changeTitle(DigitalVideoDisc dvd, String title) {
        String oldTitle = dvd.getTitle();
        dvd.setTitle(titleSet: title);
        dvd = new DigitalVideoDisc(title: oldTitle);
        System.out.println("x: " + dvd.getTitle());
    }

    // public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    //     String temp = dvd1.getTitle();
    //     dvd1.setTitle(dvd2.getTitle());
    //     dvd2.setTitle(temp);
    // }
}

```

Kết quả khi chạy:

```

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

```

⇒ Có vấn đề với hàm swap!

2. Giải quyết vấn đề trong đoạn code này

a Trả lời 2 câu hỏi (Tiếng Việt)

- 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.

b Sửa lại hàm swap

```
public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {  
    String temp = dvd1.getTitle();  
    dvd1.setTitle(titleSet: dvd2.getTitle());  
    dvd2.setTitle(titleSet: temp);  
}
```

Kết quả sau khi sửa:

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

IV. Sử dụng Debug

Em sử dụng Netbeans để lập trình Java, trong IDE này nó tích hợp sẵn công cụ Debug tự động nên có hơi khác với Eclipse, và em đã biết sử dụng công cụ Debug thủ công trong IDE này rồi.

V. Classifier Member and Instance Member

```

    */
    public class DigitalVideoDisc {
        private static int nbDigitalVideoDiscs = 0; // Class attribute can khai bao
        private String title;
        private String category;
        private String director;
        private int length;
        private float cost;
        private int id; // Instance attribute ID

        //Phương thức khởi dựng các thông tin của đĩa DVD
        //Các phương thức khởi dựng trên nạp chồng
        //Constructor by title
        public DigitalVideoDisc(String title) {
            super();
            this.title = title;
            this.id = ++nbDigitalVideoDiscs; // Update class variable and assign id
        }
        //Constructor by category, title and cost
        public DigitalVideoDisc(String title, String category, float cost) {
            this.title = title;
            this.category = category;
            this.cost = cost;
            this.id = ++nbDigitalVideoDiscs; // Update class variable and assign id
        }

        //Constructor by category, title and cost
        public DigitalVideoDisc(String title, String category, float cost) {
            this.title = title;
            this.category = category;
            this.cost = cost;
            this.id = ++nbDigitalVideoDiscs; // Update class variable and assign id
        }

        //Constructor by title, category , director, cost
        public DigitalVideoDisc(String title, String category, String director, float cost) {
            this.title = title;
            this.category = category;
            this.director = director;
            this.cost = cost;
            this.id = ++nbDigitalVideoDiscs; // Update class variable and assign id
        }

        // Constructor by all attributes
        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;
            this.id = ++nbDigitalVideoDiscs; // Update class variable and assign id
        }

        //Các phương thức lấy dữ liệu của đĩa DVD
    }

```

```

//Các phương thức lấy dữ liệu của đĩa DVD

//Getter for title
public String getTitle() { return title; }
//Getter for category
public String getCategory() { return category; }
//Getter for director
public String getDirector() { return director; }
//Getter for length
public int getLength() { return length; }
//Getter for cost
public float getCost() { return cost; }
public int getId() { return id; } // Hàm lấy id

//Cac phuong thuc chinh sua thuoc tinh cua du lieu
//Setting for title
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(float costSet){ cost = costSet; }
}

```

VI. Open the Cart class

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

```

public void print() {
    StringBuilder output = new StringBuilder("*****CART***** \nOrdered items: \n");
    for (int i = 0; i < MAX_NUMBERS_ORDERED; i++) {
        if (itemsOrdered[i] != null) {
            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);
}

```

```

public void searchById(int i) {
    if (i > MAX_NUMBERS_ORDERED || itemsOrdered[i] == null) {
        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");
    }
}

public void searchByTitle(String title) {
    for (int i = 0; i < MAX_NUMBERS_ORDERED; i++) {
        if (itemsOrdered[i] != null) {
            if (itemsOrdered[i].getTitle().toLowerCase().contains(title.toLowerCase())) {
                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 !");
}

```

2. Chạy thử và xem kết quả

- Phương thức print() được gọi ở lớp TestCart

```

    cart.print();
    cart.searchByTitle(title: "be");
    cart.searchById(i: 2);
}
}

```

IT3103 (run) × Javadoc

```

Added DVD: Aladdin
Added DVD: Beauty and the Beast
*****CART*****
Ordered items:
1.[The Lion King] - [Animation] - [Roger Allers] - [89]: 19.99 $
2.[Aladdin] - [Animation] - [Ron Clements] - [90]: 24.99 $
3.[Beauty and the Beast] - [Animation] - [Gary Trousdale] - [84]: 21.99 $
total: 66.97 $
*****

Result: [Beauty and the Beast] - [Animation] - [Gary Trousdale] - [84]: 21.99 $

Result: [Aladdin] - [Animation] - [Ron Clements] - [90]: 24.99 $

```

VII. Implement the Store class

1. Tạo lớp Store

```

class Store {
    // Danh sách các DVD trong cửa hàng
    private ArrayList<DigitalVideoDisc> itemsInStore;

    // Constructor
    public Store() {
        this.itemsInStore = new ArrayList<>();
    }

    // Phương thức thêm DVD vào cửa hàng
    public void addDVD(DigitalVideoDisc dvd) {
        itemsInStore.add(dvd);
        System.out.println("Added DVD: " + dvd.getTitle());
    }

    // Phương thức xóa DVD khỏi cửa hàng
    public void removeDVD(String title) {
        boolean removed = false;
        for (DigitalVideoDisc dvd : itemsInStore) {
            if (dvd.getTitle().equalsIgnoreCase(title)) {
                itemsInStore.remove(dvd);
                System.out.println("Removed DVD: " + dvd.getTitle());
                removed = true;
                break;
            }
        }
    }
}

```

```

        removed = true;
        break;
    }
}
if (!removed) {
    System.out.println("DVD with title '" + title + "' not found.");
}
}

// Phương thức hiển thị danh sách DVD
public void displayItems() {
    if (itemsInStore.isEmpty()) {
        System.out.println(x: "The store is empty.");
    } else {
        System.out.println(x: "DVDs available in the store:");
        for (DigitalVideoDisc dvd : itemsInStore) {
            System.out.println("- " + dvd.getTitle());
        }
    }
}
}

```

2. Tạo lớp TestStore và kiểm thử

Tạo lớp TestStore

```

public class StoreTest {
    public static void main(String[] args) {
        // Tạo cửa hàng
        Store store = new Store();

        // Tạo một số DVD
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title: "The Lion King");
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title: "Star Wars");
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title: "Aladdin");

        // Thêm DVD vào cửa hàng
        store.addDVD(dvd:dvd1);
        store.addDVD(dvd:dvd2);
        store.addDVD(dvd:dvd3);

        // Hiển thị các DVD hiện có trong cửa hàng
        store.displayItems();

        // Xóa một DVD khỏi cửa hàng
        store.removeDVD(title: "Star Wars");

        // Hiển thị lại các DVD trong cửa hàng
        store.displayItems();

        // Thử xóa một DVD không tồn tại
        store.removeDVD(title: "Harry Potter");
    }
}

```

Kiểm thử

```
run:
Added DVD: The Lion King
Added DVD: Star Wars
Added DVD: Aladdin
DVDs available in the store:
- The Lion King
- Star Wars
- Aladdin
Removed DVD: Star Wars
DVDs available in the store:
- The Lion King
- Aladdin
DVD with title 'Harry Potter' not found.
BUILD SUCCESSFUL (total time: 0 seconds)
```

VIII. Re-organize your projects

Trong code và trong project em đã làm rồi. (vì làm trong Netbeans, việc chia nhánh không giống như yêu cầu được, em đã hiểu việc chia nhánh và tự sắp xếp ở trong thư mục nên phạm import trong mỗi phần code sẽ không khớp)

IX. String, StringBuilder and StringBuffer

1. Tạo lớp ConcatenationInLoops

```
public class ConcatenationInLoops {
    public static void main(String[] args) {
        // Sử dụng String để nối chuỗi
        Random r = new Random(seed:123); // Khởi tạo bộ sinh số ngẫu nhiên với seed = 123
        long start = System.currentTimeMillis(); // Lấy thời gian bắt đầu
        String s = "";
        for (int i = 0; i < 65536; i++) {
            s += r.nextInt(bound: 2); // Thêm số ngẫu nhiên vào chuỗi
        }
        System.out.println(System.currentTimeMillis() - start); // In thời gian thực thi
        // Ghi chú: Dòng này in khoảng 4500 ms (tùy vào cấu hình máy).

        // Sử dụng StringBuilder để nối chuỗi
        r = new Random(seed:123); // Tạo lại bộ sinh số ngẫu nhiên
        start = System.currentTimeMillis(); // Lấy thời gian bắt đầu
        StringBuilder sb = new StringBuilder(); // Sử dụng StringBuilder để tối ưu
        for (int i = 0; i < 65536; i++) {
            sb.append(i: r.nextInt(bound: 2)); // Thêm số ngẫu nhiên vào StringBuilder
        }
        s = sb.toString(); // Chuyển đổi StringBuilder thành chuỗi
        System.out.println(System.currentTimeMillis() - start); // In thời gian thực thi
        // Ghi chú: Dòng này in khoảng 5 ms (tùy vào cấu hình máy).
    }
}
```

2. Tạo lớp GarbageCreator

```

public class GarbageCreator {
    public static void main(String[] args) throws IOException {
        String filename = "G:\\OOPLab\\IT3103.732871.2023.1.20215060.DoGiaHuy\\Lab02_03_04_05\\Other";
        byte[] inputBytes = {0};
        long startTime, endTime;
        inputBytes = Files.readAllBytes(path: Paths.get(first: filename));
        startTime = System.currentTimeMillis();
        String outputString = "";
        for(byte b : inputBytes) {
            outputString += (char)b;
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}

```

3. Tạo lớp NoGarbage

```

public class NoGarbage {
    public static void main(String[] args) throws IOException {
        String filename = "G:\\OOPLab\\IT3103.732871.2023.1.20215060.DoGiaHuy\\Lab02_03_04_05\\OtherProject\\src\\hust\\soict";
        byte[] inputBytes = {0};
        long startTime, endTime;

        inputBytes = Files.readAllBytes(path: Paths.get(first: filename));
        startTime = System.currentTimeMillis();
        StringBuilder outputStringBuilder = new StringBuilder(str: "");
        for(byte b : inputBytes) {
            outputStringBuilder.append((char)b);
        }
        endTime = System.currentTimeMillis();
        System.out.println(endTime - startTime);
    }
}

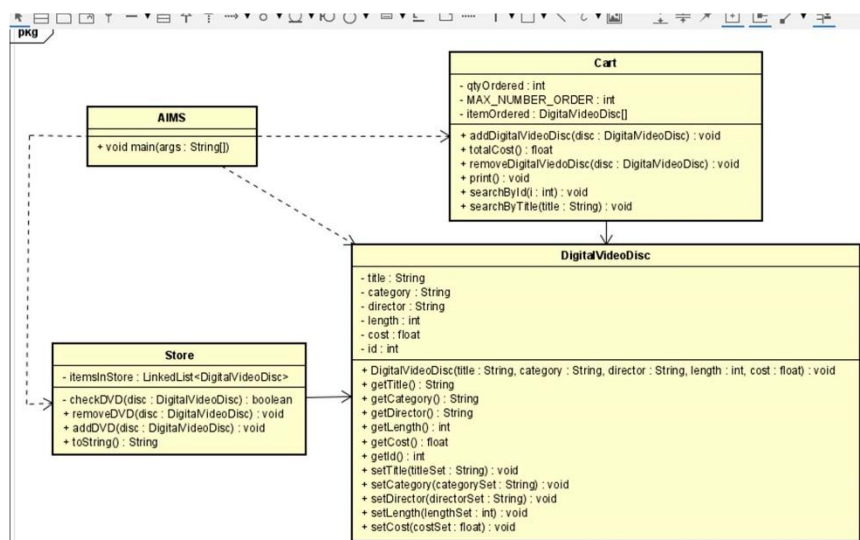
```

X. Release flow demonstration

Em đã thử làm cái này rồi

XI. Cập nhật các Diagram trong project AIMS

1. Class diagram



2. Use-case diagram

