Họ và tên: Bùi Thị Xuân

MSSV: 20225957

Báo cáo LAB03

1. Working with method overloading

```java
// them dia DVD vao gio hang
public void addDigitalVideoDisc(DigitalVideoDisc disc) {
    if (qtyOrdered == MAX_NUMBERS_ORDERED) {
        System.out.println(x:"The cart is full. Can't add more disc");
        return;
    }
    else {
        itemsOrdered[qtyOrdered] = disc;
        qtyOrdered++;
        System.out.println("The DVD " + disc.getTitle() + " has been added into cart");
        return;
    }
}
```

```java
// them list dia DVD vao gio hang 2.1
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
    for (DigitalVideoDisc disc : dvdList) {
        if (qtyOrdered == MAX_NUMBERS_ORDERED) {
            System.out.println("The cart is full. Can't add more disc: " + disc.getTitle());
            return;
        }
        else {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD " + disc.getTitle() + " has been added into cart");
        }
    }
}
```

```java
// them so luong tuy y dia DVD 2.1
public void addDigitalVideoDisc(DigitalVideoDisc... discs) {
    for (DigitalVideoDisc disc : discs) {
        if (qtyOrdered == MAX_NUMBERS_ORDERED) {
            System.out.println("The cart is full. Can't add more disc: " + disc.getTitle());
            return;
        }
        else {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The DVD " + disc.getTitle() + " has been added into cart");
        }
    }
}
```

```java
// them 2 dia DVD
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    // them dia 1
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered++;
        System.out.println("The DVD " + dvd1.getTitle() + " has been added into the cart");
    }
    else {
        System.out.println("The cart is full. Can't add more discs. Failed to add: " + dvd1.getTitle());
    }

    // them dia 2
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = dvd2;
        qtyOrdered++;
        System.out.println("The DVD " + dvd2.getTitle() + " has been added into the cart");
    }
    else {
        System.out.println("The cart is full. Can't add more discs. Failed to add: " + dvd2.getTitle());
    }
}
```

2. Passing parameter

```java
public class TestPassingParameter {
    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(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
        DigitalVideoDisc tmp = new DigitalVideoDisc(dvd1.getTitle(), dvd1.getCategory(), dvd1.getDirector(), dvd1.get

        dvd1.setTitle(dvd2.getTitle());
        dvd1.setCategory(dvd2.getCategory());
        dvd1.setDirector(dvd2.getDirector());
        dvd1.setLength(dvd2.getLength());
        dvd1.setCost(dvd2.getCost());

        dvd2.setTitle(tmp.getTitle());
        dvd2.setCategory(tmp.getCategory());
        dvd2.setDirector(tmp.getDirector());
        dvd2.setLength(tmp.getLength());
        dvd2.setCost(tmp.getCost());
    }

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

3. Classifier Member and Instance Member

```java
public class DigitalVideoDisc {
    private String title;
    private String category;
    private String director;
    private int length;
    private float cost;

    private static int nbDigitalVideoDiscs = 0;
    private int id; // Instance attribute ID

    public DigitalVideoDisc(String title) {
        super();
        this.title = title;
        this.id = ++nbDigitalVideoDiscs; // Update class variable and assign id
    }

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

```
    public String getTitle() {
        return title;
    }
    public String getCategory() {
        return category;
    }
    public String getDirector() {
        return director;
    }
    public int getLength() {
        return length;
    }
    public float getCost() {
        return cost;
    }
    public int getId() {return id;} // Ham lay id

    public void setTitle(String title){this.title = title;}
    public void setCategory(String category){this.category = category;}
    public void setDirector(String director){this.director = director;}
    public void setLength(int length){this.length = length;}
    public void setCost(float cost){this.cost = cost;}
}
```

4. Open the **Cart** class

```
    // tinh tong gia tien dia
    public float totalCost() {
        float sum = 0;
        for(int i = 0; i < qtyOrdered; i++) {
            sum += itemsOrdered[i].getCost();
        }
        return sum;
    }

    // hien thong tin dia
    public void displayDigitalVideoDisc() {
        StringBuilder output = new StringBuilder(str:"**********CART********** \nOrdered it
        for(int i = 0; i < qtyOrdered; i++) {
            output.append(i+1 + ". DVD - " + itemsOrdered[i].getTitle() + " - " + itemsOrdered[i].getCategory() + " -
                    + itemsOrdered[i].getDirector() + " - " + itemsOrdered[i].getLength() + ": "
                    + itemsOrdered[i].getCost() + " $\n");
        }
        output.append(str:"Total cost: ").append(totalCost()).append(str:" $\n");
        output.append(str:"***************************************************\n");
        System.out.println(output);
    }
```

```java
// tim kiem boi ID
public void searchById(int i) {
    if(i > qtyOrdered || i <= 0) {
        System.out.println(x:"No match found!\n");
    }
    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");
    }

}

// tim kiem bang title
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(x:"No match found!\n");
}
```

int i - hust.soict.hedspi.aims.cart.Cart.searchByTitle(String)

You, 42 minutes ago | 1 author (You)

```java
public class CartTest {
    Run | Debug
    public static void main(String[] args) {
        Cart cart = new Cart();
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King",category:"Animation",
                director:"Roger Allers",length:87,cost:19.95f);
        cart.addDigitalVideoDisc(dvd1);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star wars",category:"Science Fiction",
                director:"Geogre Lucas",length:87,cost:24.95f);
        cart.addDigitalVideoDisc(dvd2);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin",category:"Animation",cost:18.99f);
        cart.addDigitalVideoDisc(dvd3);

        cart.displayDigitalVideoDisc();

        //Test search by ID method
        cart.searchById(i:3);
        cart.searchById(i:4);

        //Test search by Title method
        cart.searchByTitle(title:"The Lion King");
        cart.searchByTitle(title:"Alan Walker");
    }
}
```

5. Implement the **Store** class

```java
public class Store {
   private DigitalVideoDisc[] itemsInStore = new DigitalVideoDisc[1000];
   private int numItems = 0;
   private static final int MAX_CAPACITY = 1000;

   public Store() {
   }

   public boolean checkDVD(DigitalVideoDisc disc) {
      DigitalVideoDisc[] var5;
      int var4 = (var5 = this.itemsInStore).length;

      for(int var3 = 0; var3 < var4; ++var3) {
         DigitalVideoDisc digitalVideoDisc = var5[var3];
         if (digitalVideoDisc != null && digitalVideoDisc.equals(disc)) {
            return true;
         }
      }

      return false;
   }
```

```java
public void removeDVD(DigitalVideoDisc disc) {
   if (!this.checkDVD(disc)) {
      System.out.println("There is no " + disc.getTitle() + " in the store!");
   } else {
      for(int i = 0; i < this.numItems; ++i) {
         if (this.itemsInStore[i].getTitle().equals(disc.getTitle())) {
            System.out.println("DVD \"" + this.itemsInStore[i].getTitle() + "\" has been removed from the store.")

            for(int j = i; j < this.numItems - 1; ++j) {
               this.itemsInStore[j] = this.itemsInStore[j + 1];
            }

            this.itemsInStore[this.numItems - 1] = null;
            --this.numItems;
            return;
         }
      }

   }
}
```

```java
public void addDVD(DigitalVideoDisc disc) {
    if (!this.checkDVD(disc)) {
        this.itemsInStore[this.numItems] = disc;
        ++this.numItems;
        System.out.println(disc.getTitle() + " has been added to the store!");
    } else {
        System.out.println(disc.getTitle() + " already exists in the store!");
    }

}

public String toString() {
    StringBuilder string = new StringBuilder("***************STORE**************\nItems in the store: \n");
    if (this.numItems == 0) {
        string.append("There is no dvd in the store!\n");
    } else {
        DigitalVideoDisc[] var5;
        int var4 = (var5 = this.itemsInStore).length;

        for(int var3 = 0; var3 < var4; ++var3) {
            DigitalVideoDisc dvd = var5[var3];
            if (dvd != null) {
                string.append(dvd.getTitle() + " - " + dvd.getCost() + " $\n");
            }
        }
    }
```

6.

```java
package hust.soict.hedspi.garbage;      You, 27 minutes ago |

import java.util.Random;

You, 27 minutes ago | 1 author (You)
public class ConcatenationInLoops {
    Run | Debug
    public static void main(String[] args) {
        Random r = new Random(seed:123);
        long start = System.currentTimeMillis();
        String s = "";
        for (int i = 0; i < 65536; i++) {
            s += r.nextInt(bound:2);
        }
        System.out.println(System.currentTimeMillis() - start); // This prints roughly 4500.

        r = new Random(seed:123);
        start = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 65536; i++) {
            sb.append(r.nextInt(bound:2));
        }
        s = sb.toString();
        System.out.println(System.currentTimeMillis() - start); // This prints 5.
    }
}
```

```java
You, 30 minutes ago | 1 author (You)
package hust.soict.hedspi.garbage;      You, 30 minutes ago • UPDATE LAB03

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

You, 30 minutes ago | 1 author (You)
public class GarbageCreator {
    Run | Debug
    public static void main(String[] args) throws IOException {
        String filename = "D:\\OOP_LAB\\OtherProjects\\test.txt";
        byte[] inputBytes = { 0 };
        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);
    }
}
```

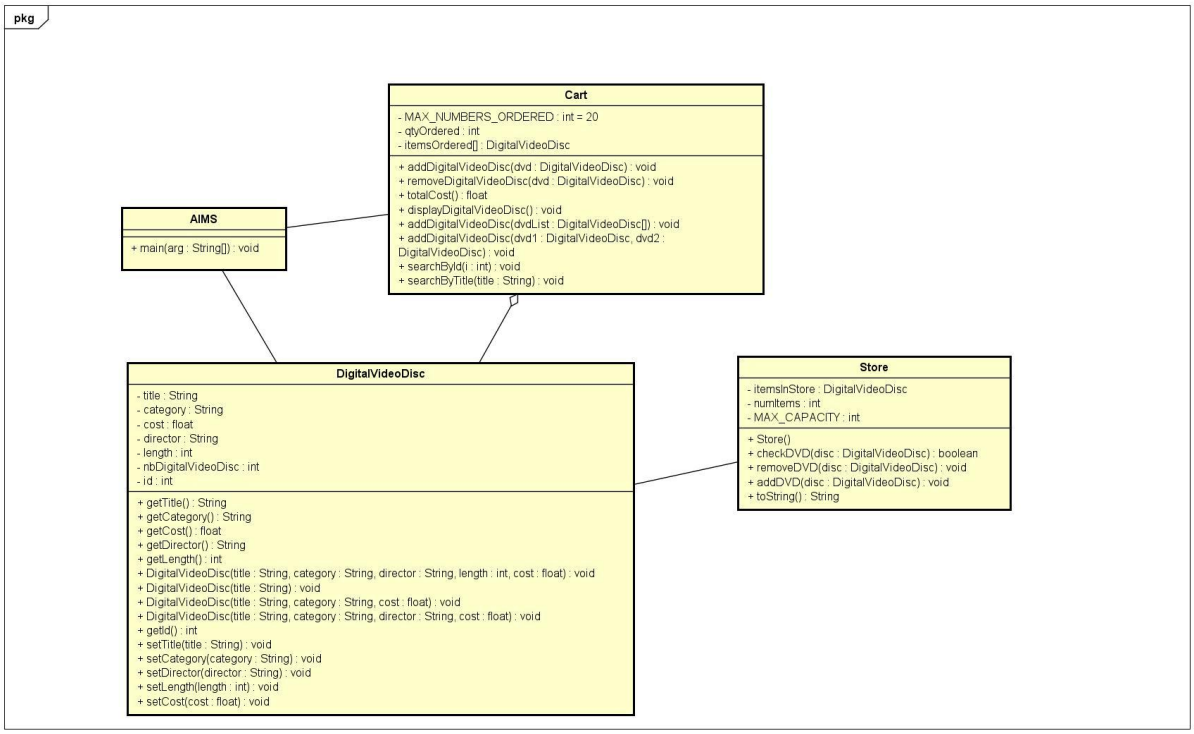
```java
You, 32 minutes ago | 1 author (You)
package hust.soict.hedspi.garbage;      You, 32 minutes ago • UPDATE LAB03

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

You, 32 minutes ago | 1 author (You)
public class NoGarbage {
    Run | Debug
    public static void main(String[] args) throws IOException {
        String filename = "D:\\OOP_LAB\\OtherProjects\\test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

        inputBytes = Files.readAllBytes(Paths.get(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);
    }
}
```

7. Images of the uploaded use-case diagram and class diagram




8. Trả lời câu hỏi

**Điểm giống:** Cả hai đều cần sử dụng vòng lặp để duyệt qua và xử lý các phần tử.

**Điểm khác:**

- **Tham số mảng:** Yêu cầu phải tạo một mảng trước, làm giảm tính linh hoạt vì cần thêm một bước chuẩn bị.
- **Tham số số lượng đối số tùy ý:** Cho phép truyền nhiều đối số trực tiếp mà không cần qua bước trung gian, gọn gàng hơn khi gọi phương thức và dễ dàng mở rộng.

**Ý kiến cá nhân:** Em ưu tiên sử dụng tham số số lượng đối số tùy ý vì cách này đơn giản hơn, tiết kiệm thời gian và linh hoạt khi áp dụng trong các tình huống khác nhau.

## a. Sau khi gọi swap(jungleDVD, cinderellaDVD) tại sao tiêu đề của hai đối tượng vẫn giữ nguyên?

Khi gọi swap(jungleDVD, cinderellaDVD), tiêu đề của hai đối tượng không thay đổi vì trong Java, các tham số được truyền bằng giá trị (pass by value).

Cụ thể, khi truyền các đối tượng vào phương thức swap, các tham số như o1 và o2 chỉ là bản sao của các tham chiếu gốc (references). Việc hoán đổi giá trị của o1 và o2 chỉ làm thay đổi các tham chiếu bên trong phương thức, nhưng không ảnh hưởng đến các tham chiếu ban đầu bên ngoài. Do đó, các đối tượng jungleDVD và cinderellaDVD vẫn giữ nguyên giá trị ban đầu.

**b. Sau khi gọi changeTitle(jungleDVD, cinderellaDVD.getTitle()), tại sao tiêu đề của JungleDVD lại thay đổi?**

Phương thức changeTitle sử dụng tham chiếu thực sự của đối tượng jungleDVD. Tham chiếu này cho phép truy cập trực tiếp vào thuộc tính title của đối tượng. Khi phương thức thực hiện lời gọi dvd.setTitle(title), nó không chỉ thay đổi thuộc tính title cục bộ mà còn cập nhật thuộc tính trong đối tượng gốc mà tham chiếu jungleDVD trỏ tới. Do đó, sự thay đổi thông qua phương thức này được phản ánh ngay trên đối tượng ban đầu.