BÁO CÁO OOP LAB03

- ***METHODS OVERLOADING***

Overloading by differing the number of parameters

```java
// Add 2 DVDs to the current cart
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = dvd1;
        qtyOrdered++;
        System.out.println("The disc " + dvd1.getTitle() + " has been added.");
    } else {
        System.out.println("The cart is full. Unable to add the disc " + dvd1.getTitle() + ".");
        return; // Exit if no slots available for the first DVD
    }

    if (qtyOrdered < MAX_NUMBERS_ORDERED) {
        itemsOrdered[qtyOrdered] = dvd2;
        qtyOrdered++;
        System.out.println("The disc " + dvd2.getTitle() + " has been added.");
    } else {
        System.out.println("The cart is full. Unable to add the disc " + dvd2.getTitle() + ".");
    }
}
```

Overloading by differing types of parameter

```java
// Add a list of DVDs to the current cart.
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
    for (DigitalVideoDisc disc : dvdList) {
        if (qtyOrdered < MAX_NUMBERS_ORDERED) {
            itemsOrdered[qtyOrdered] = disc;
            qtyOrdered++;
            System.out.println("The disc " + disc.getTitle() + " has been added.");
        } else {
            System.out.println("The cart is full. Unable to add the disc " + disc.getTitle() + ".");
            break;
        }
    }
}
```

- ***PASSING PARAMETERS***

- Question: Is JAVA a Pass by Value or a Pass by Reference programming language?

Answer: Java là Pass by Value. Java luôn truyền bản sao giá trị vào phương thức

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

Answer: Do jungleDVD và cinderellaDVD không bị thay đổi tham chiếu bên ngoài phương thức, các đối tượng ban đầu không bị hoán đổi. Vì vậy, tiêu đề của chúng vẫn giữ nguyên.

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

Answer: Việc tiêu đề của jungleDVD thay đổi là do tham chiếu được truyền vào phương thức cho phép thay đổi trạng thái của đối tượng mà nó trỏ đến. Tuy nhiên, việc gán một đối tượng mới cho tham chiếu cục bộ dvd không ảnh hưởng đến tham chiếu gốc bên ngoài.

- A swap() method that can correctly swap the two objects:

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

- ***CLASSIFIER MEMBER AND INSTANCE MEMBER***

Sau khi thêm id cho các DVD:

```
public class DigitalVideoDisc {
        private String title;
        private String category;
        private String director;
        private int length;
        private float cost;
        private int id;
        private static int nbDigitalVideoDiscs = 0;

        //Getters
        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;
        }
        public static int getNbDigitalVideoDiscs() {
```

```
        return nbDigitalVideoDiscs;
    }

        // Constructors
    public DigitalVideoDisc(String title) {
        super();
        this.title = title;
        incrementAndSetId();
    }


    public DigitalVideoDisc(String category, String title, float cost) {
        super();
        this.title = title;
        this.category = category;
        this.cost = cost;
        incrementAndSetId();
    }

    public DigitalVideoDisc(String director, String category, String title, float cost){
        super();
        this.title = title;
        this.category = category;
        this.cost = cost;
        this.director = director;
        incrementAndSetId();
    }

    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;
        incrementAndSetId();
    }
    // Private method to increment the number of DVDs and set the unique ID
    private void incrementAndSetId() {
        nbDigitalVideoDiscs++;
        this.id = nbDigitalVideoDiscs;
    }
    //toString method to display DVD
```

```
    @Override
        public String toString() {
            return "DigitalVideoDisc [id=" + id + ", title=" + title + ", category=" +
category + ", director=" + director + ", length="
                    + length + ", cost=" + cost + "]";
        }

    //Setter for title
        public void setTitle(String title) {
     if (title != null && !title.trim().isEmpty()) {
        this.title = title;
        } else {
            System.out.println("Invalid title. Title cannot be null or empty.");
        }
        }
}
```

- ***PRINT THE LIST AND SEARCH DVDS***

Print the list of ordered items of a cart, the price of each item, and the total price

```
// Display the DVDs in the cart
public void displayCart() {
    System.out.println("***************CART****************");
    System.out.println("Ordered Items:");

    for (int i = 0; i < qtyOrdered; i++) {
        DigitalVideoDisc dvd = itemsOrdered[i];
        System.out.printf("%d. DVD - [%s] - [%s] - [%s] - [%d]: [%.2f] $\n",
            i + 1,
            dvd.getTitle(),
            dvd.getCategory() == null ? "Unknown Category" : dvd.getCategory(),
            dvd.getDirector() == null ? "Unknown Director" : dvd.getDirector(),
            dvd.getLength(),
            dvd.getCost()
        );
    }

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

Search for DVDs in the cart by ID

```java
// Search for a DVD by ID
public void searchById(int id) {
    boolean found = false; // Flag to track if the DVD is found
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].getId() == id) {
            System.out.println("DVD Found: " + itemsOrdered[i].toString());
            found = true;
            break;
        }
    }
    if (!found) {
        System.out.println("No DVD found with ID: " + id);
    }
}
```

Search for DVDs in the cart by title

```java
// Search for a DVD by title
public void searchByTitle(String title) {
    boolean found = false; // Flag to track if a match is found
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemsOrdered[i].getTitle().equalsIgnoreCase(title)) { // Case-insensitive comparison
            System.out.println("DVD Found: " + itemsOrdered[i].toString());
            found = true;
        }
    }
    if (!found) {
        System.out.println("No DVD found with the title: " + title);
    }
```

- ***IMPLEMENT THE STORE CLASS***

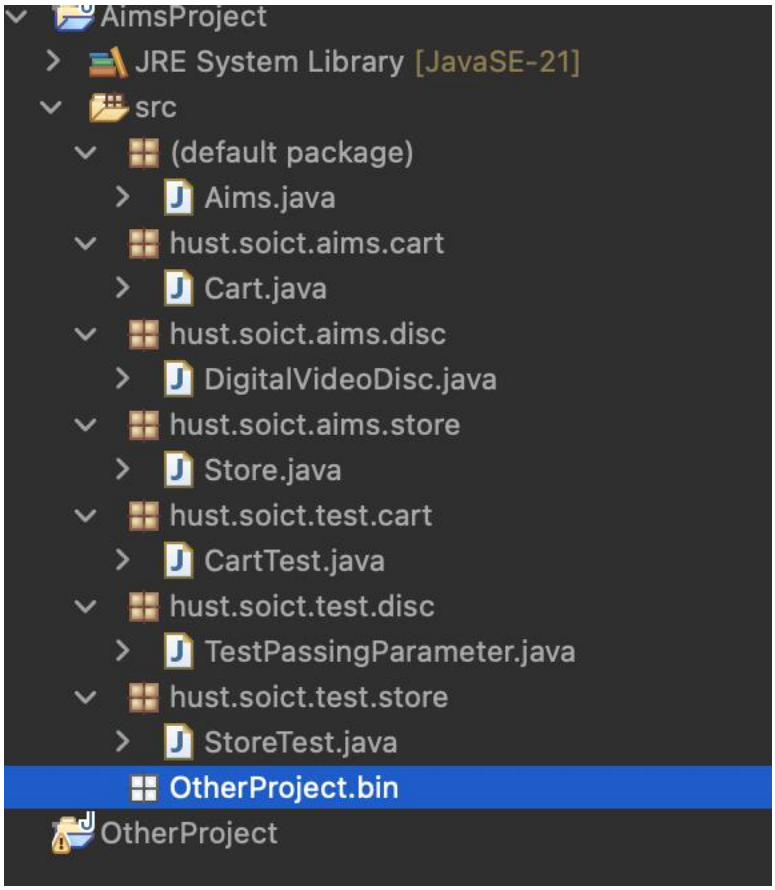
```java
public class Store {
    public static final int MAX_NUMBERS_IN_STORE = 50;
    private DigitalVideoDisc itemsInStore[] = new DigitalVideoDisc[MAX_NUMBERS_IN_STORE];
    private int qtyInStore = 0;

    // Add a DVD to the store
    public void addDVD(DigitalVideoDisc disc) {
        if (qtyInStore < MAX_NUMBERS_IN_STORE) {
            itemsInStore[qtyInStore] = disc;
            qtyInStore++;
            System.out.println("The DVD " + disc.getTitle() + " has been added to the store.");
        } else {
            System.out.println("The store is full. Cannot add more DVDs.");
        }
    }

    // Remove a DVD from the store
    public void removeDVD(DigitalVideoDisc disc) {
```

```
        for (int i = 0; i < qtyInStore; i++) {
            if (itemsInStore[i] == disc) {
                for (int j = i; j < qtyInStore - 1; j++) {
                    itemsInStore[j] = itemsInStore[j + 1];
                }
                itemsInStore[qtyInStore - 1] = null;
                qtyInStore--;
                System.out.println("The DVD " + disc.getTitle() + " has been removed from the store.");
                return;
            }
        }
        System.out.println("The DVD " + disc.getTitle() + " is not found in the store.");
    }
}
```

- **RE-ORGANIZED THE PROJECT**



- ***STRING, STRINGBUILDER AND STRINGBUFFER***

```java
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 = "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
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

public class NoGarbage {
    public static void main(String[] args) throws IOException {
        String filename = "test.txt";
        byte[] inputBytes = { 0 };
        long startTime, endTime;

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