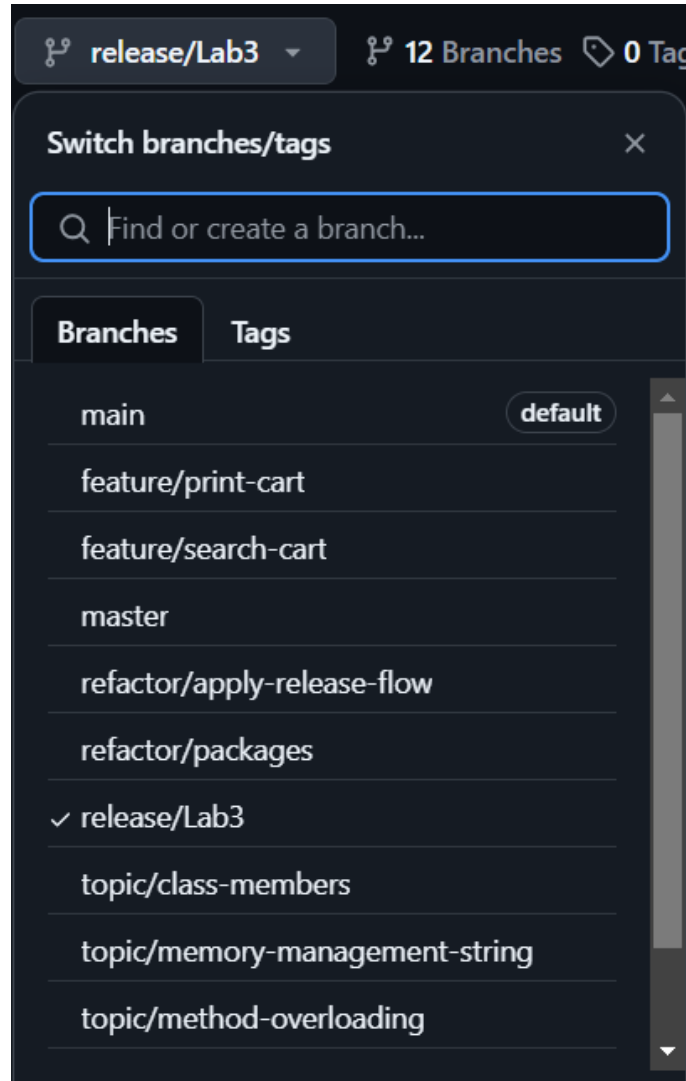


# Object-Oriented Programming

Nguyễn Minh Quang -20226123

## Lab 03: Basic Object-Oriented Techniques

### 1. Branch your repository



### 2. Working with method overloading

#### 2.1 *Overloading by differing types of parameter -*

Code:

File **Cart.java**, phần code đã bổ sung so với Lab 02

```
// // Thêm danh sách DVD (mảng)
public void addDigitalVideoDisc(DigitalVideoDisc[] dvdList) {
    for (DigitalVideoDisc disc : dvdList) {
        if (disc != null) {
            if (qtyOrdered < MAX_NUMBERS_ORDERED) {
                itemOrdered[qtyOrdered] = disc;
                qtyOrdered++;
                System.out.println("The disc \"" + disc.getTitle() + "\" has been added.");
            } else {
                System.out.println(x:"The cart is full. Cannot add more DVDs.");
                break;
            }
        }
    }
}

// Thêm số lượng DVD bất kỳ (varargs)
public void addDigitalVideoDisc(DigitalVideoDisc... dvds) {
    for (DigitalVideoDisc disc : dvds) {
        if (disc != null) {
            if (qtyOrdered < MAX_NUMBERS_ORDERED) {
                itemOrdered[qtyOrdered] = disc;
                qtyOrdered++;
                System.out.println("The disc \"" + disc.getTitle() + "\" has been added.");
            } else {
                System.out.println(x:"The cart is full. Cannot add more DVDs.");
                break;
            }
        }
    }
}
}
```

#### - Giải thích tại sao không chạy được 2 đoạn code 1 lúc

Java không cho phép bạn overload (nạp chồng) phương thức chỉ dựa trên sự khác biệt giữa varargs (DigitalVideoDisc...) và mảng (DigitalVideoDisc[]). Cả hai loại tham số này đều được biên dịch thành mảng khi chương trình chạy, dẫn đến Java không thể phân biệt được giữa hai phương thức.

☐ Để chương trình hoạt động, giữ lại một trong hai phương thức và comment hoặc xóa phương thức còn lại.

#### - Câu hỏi: Compare to an array parameter. What do you prefer in this case?

Sử dụng **varargs** (DigitalVideoDisc...) vì linh hoạt hơn so với mảng. Varargs cho phép truyền cả nhiều tham số riêng lẻ lẫn mảng mà không cần chuẩn bị trước, đơn giản hóa mã nguồn và dễ sử dụng hơn. Hiệu suất của varargs và mảng gần như tương đương, nên ưu tiên varargs trong trường hợp này.

## 2.2 Overloading by differing the number of parameters

```
// Thêm hai DVD
public void addDigitalVideoDisc(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    addDigitalVideoDisc(dvd1);
    addDigitalVideoDisc(dvd2);
}
}
```

## 3. Passing parameter

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

+ Java là ngôn ngữ **Pass by Value**. Khi truyền một biến vào một phương thức, Java thực sự truyền một bản sao giá trị của biến đó, không phải bản thân biến.

+ **Pass by Reference:** Java không hỗ trợ Pass by Reference, nhưng khi truyền một đối tượng (reference type), Java truyền bản sao địa chỉ của đối tượng. Điều này cho phép phương thức thay đổi trạng thái của đối tượng, nhưng không thể thay đổi tham chiếu gốc.

```
public class TestPassingParameter {
    Run | Debug
    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"Cinderella");
        swap(jungleDVD,cinderellaDVD);

        System.out.println("jungle title: " + jungleDVD.getTitle());
        System.out.println("cinderella title: " + cinderellaDVD.getTitle());

        changeTitle(jungleDVD,cinderellaDVD.getTitle());
        System.out.println("jungle 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);
    }
}

public class TestPassingParameter {
    Run | Debug
    public static void main(String[] args) {
        DigitalVideoDisc jungleDVD = new DigitalVideoDisc(title:"Jungle");
        DigitalVideoDisc cinderellaDVD = new DigitalVideoDisc(title:"Cinderella");
        swap(jungleDVD,cinderellaDVD);

        System.out.println("jungle title: " + jungleDVD.getTitle());
        System.out.println("cinderella title: " + cinderellaDVD.getTitle());

        changeTitle(jungleDVD,cinderellaDVD.getTitle());
        System.out.println("jungle 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);
    }
}
```

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

+ Sau khi gọi hàm, hai tiêu đề của hai đối tượng không thay đổi vì Java là Pass by Value. Khi gọi phương thức, Java chỉ truyền bản sao tham chiếu của jungleDVD và cinderellaDVD vào swap.

+ Bất kì thay đổi nào trên o1 và o2 (tham số của swap) chỉ ảnh hưởng tới bản sao tham số, không làm thay đổi tham chiếu gốc bên ngoài. Vì vậy đối tượng gốc vẫn giữ nguyên.

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

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

Tiêu đề của jungleDVD thay đổi vì trong Java, khi truyền tham số kiểu tham chiếu, bản sao địa chỉ của đối tượng được truyền vào. Phương thức `changeTitle` gọi `setTitle`, thay đổi trực tiếp thuộc tính của đối tượng mà `jungleDVD` trỏ tới. Do đó, tiêu đề của đối tượng bị cập nhật.

- Viết hàm `swap()` để thực hiện đúng mục đích

```
public static void swap(DigitalVideoDisc dvd1, DigitalVideoDisc dvd2) {
    DigitalVideoDisc tmp = new DigitalVideoDisc(dvd1.getTitle(), dvd1.getCategory(), dvd1.getDirector(), dvd1.getLength(), dvd1.getCost());

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

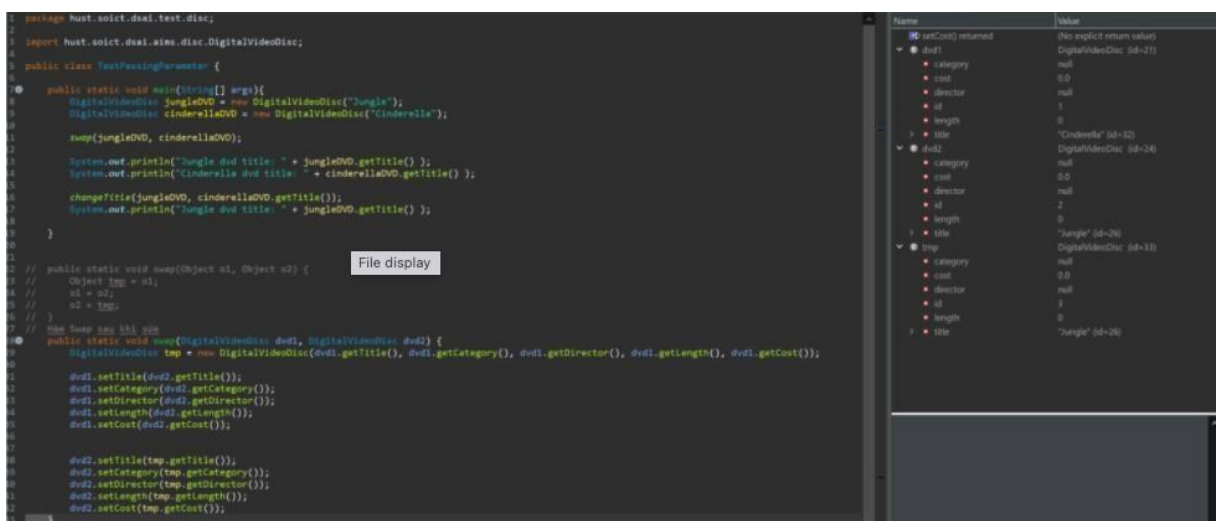
    dvd2.setTitle(tmp.getTitle());
    dvd2.setCategory(tmp.getCategory());
    dvd2.setDirector(tmp.getDirector());
    dvd2.setLength(tmp.getLength());
    dvd2.setCost(tmp.getCost());
}
```

#### 4. Debug

- Debug với hàm `swap` cũ



- Debug với hàm `swap` mới



#### 5. Classifier Member and Instance Member

- Thêm thuộc tính "id" và thực hiện gán giá trị cho DVD theo code. Code được bổ sung vào `DigitalVideoDisc` class:

```
//Khai báo thuộc tính cho class
private int id;

//Thêm thuộc tính cho class
private static int nbDigitalVideoDiscs = 0;

public DigitalVideoDisc(String title, String category, String director, int length, float cost) {
    // Tăng biến đếm của lớp
    nbDigitalVideoDiscs++;
    // Gán id cho đối tượng
    this.id = nbDigitalVideoDiscs;

    this.title = title;
    this.category = category;
    this.director = director;
    this.length = length;
    this.cost = cost;
}

// Getter cho nbDigitalVideoDiscs
public static int getNbDigitalVideoDiscs() {
    return nbDigitalVideoDiscs;
}
```

## 6. Open the Cart class

- Create a new method to print the list of ordered items of a cart, the price of each item, and the total price.

```
// List item, tính tổng chi phí
public float totalCost() {
    float total = 0;
    System.out.println("Items in cart:");
    for (int i = 0; i < qtyOrdered; i++) {
        if (itemOrdered[i] != null) {
            System.out.println((i + 1) + ". " + itemOrdered[i].getTitle() + " - $" + itemOrdered[i].getCost());
            total += itemOrdered[i].getCost();
        }
    }
    System.out.println("Total cost: $" + total);
    return total;
}
```

- Kết quả:

```
The disc "The Lion" has been added.
The disc "Star Wars" has been added.
The disc "Aladin" has been added.
Items in cart:
1. The Lion - $19.95
2. Star Wars - $24.95
3. Aladin - $18.99
Total cost: $63.89
The disc "Star Wars" has been removed.
The disc "Aladin" has been removed.
Items in cart:
1. The Lion - $19.95
Total cost: $19.95
```

**Suggestion:** Write a **toString()** method for the **DigitalVideoDisc** class. What should be the return type of this method?

- Search for DVDs in the cart by ID and display the search results. Make sure to notify the user if no match is found.
- Search for DVDs in the cart by title and print the results. Make sure to notify the user if no match is found. **Refer to problem statement in Lab02 for the matching rule.**

**Suggestion:** write a **boolean isMatch(String title)** method in the **DigitalVideoDisc** which finds out if the corresponding disk is a match given the title.



```

// Tìm kiếm DVD theo ID
public DigitalVideoDisc searchById(int id) {
    for (DigitalVideoDisc disc : itemOrdered) {
        if (disc != null && disc.getId() == id) {
            System.out.println("Found: " + disc);
            return disc;
        }
    }

    System.out.println("No match found for ID: " + id);
    return null;
}

// Tìm kiếm DVD theo tiêu đề
public DigitalVideoDisc searchByTitle(String title) {
    for (DigitalVideoDisc disc : itemOrdered) {
        if (disc != null && disc.isMatch(title)) {
            System.out.println("Found: " + disc);
            return disc;
        }
    }

    System.out.println("No match found for title: \"" + title + "\"");
    return null;
}

```

- Kết quả:

```

Items in cart:
1. The Lion - $19.95
2. Star Wars - $24.95
3. Aladin - $18.99
Total cost: $63.89
The disc "Star Wars" has been removed.
The disc "Aladin" has been removed.
Items in cart:
1. The Lion - $19.95
Total cost: $19.95
No match found for ID: 2
No match found for title: "Aladdin"
No match found for title: "Frozen"

```

- **CartTest** class:

```

3_OOP > AimsProject > src > J CartTest.java > CartTest > main(String[])
1
2 public class CartTest {
3
4     Run | Debug
5     public static void main(String[] args) {
6         DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion", category:"Animation",director:"Roger Allers",length:87,cost:19.95f);
7         DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars",category:"Science Fiction",director:"George Lucas",length:87,cost:24.95f);
8         DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladin",category:"Animation",cost:18.99f);
9         Cart anOrder = new Cart();
10
11         anOrder.addDigitalVideoDisc(dvd1);
12         anOrder.addDigitalVideoDisc(dvd2);
13         anOrder.addDigitalVideoDisc(dvd3);
14         anOrder.totalCost();
15         anOrder.removeDigitalVideoDisc(dvd2);
16
17         anOrder.removeDigitalVideoDisc(dvd3);
18         anOrder.totalCost();
19
20         // Tìm kiếm DVD theo ID
21         anOrder.searchById(id:2); // ID của "Star Wars"
22
23         // Tìm kiếm DVD theo tiêu đề
24         anOrder.searchByTitle(title:"Aladdin");
25
26         // Tìm kiếm không thành công
27         anOrder.searchByTitle(title:"Frozen");
28
29         // Hiển thị giỏ hàng và tính tổng chi phí
30         anOrder.totalCost();
31
32         // Xóa DVD
33         anOrder.removeDigitalVideoDisc(dvd2);
34
35         // Hiển thị lại giỏ hàng
36         anOrder.totalCost();
37     }
38 }

```

## 7. Implement the Store class -

File Store.java

```
public class Store {
    public static final int MAX_ITEMS_IN_STORE = 100; // Giới hạn số lượng DVD trong cửa hàng
    private DigitalVideoDisc[] itemsInStore; // Mảng lưu trữ DVD
    private int qtyInStore; // Số lượng DVD hiện tại

    // Constructor
    public Store() {
        itemsInStore = new DigitalVideoDisc[MAX_ITEMS_IN_STORE];
        qtyInStore = 0;
    }

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

    // Xóa một DVD khỏi cửa hàng
    public void removeDVD(DigitalVideoDisc disc) {
        for (int i = 0; i < qtyInStore; i++) {
            if (itemsInStore[i] != null && itemsInStore[i].getTitle().equals(disc.getTitle())) {
                // Xóa DVD
                itemsInStore[i] = null;
                // Dồn các phần tử phía sau lên trước để giữ cấu trúc mảng
                for (int j = i; j < qtyInStore - 1; j++) {
                    itemsInStore[j] = itemsInStore[j + 1];
                }
                itemsInStore[qtyInStore - 1] = null; // Giải phóng phần tử cuối
                qtyInStore--; // Giảm số lượng
                System.out.println("The disc \"" + disc.getTitle() + "\" has been removed from the store.");
                return;
            }
        }
        System.out.println("The disc \"" + disc.getTitle() + "\" was not found in the store.");
    }

    // Hiển thị danh sách DVD trong cửa hàng
    public void displayStore() {
        System.out.println("Items in the store:");
        for (int i = 0; i < qtyInStore; i++) {
            if (itemsInStore[i] != null) {
                System.out.println((i + 1) + ". " + itemsInStore[i].toString());
            }
        }
        System.out.println("Total items in store: " + qtyInStore);
    }
}
```

- File Store.test:

```

public class StoreTest {
    Run | Debug
    public static void main(String[] args) {
        Store store = new Store();

        // Tạo các DVD mẫu
        DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"The Lion King", category:"Animation", director:"Roger Allers", length:88, cost:19.99f);
        DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Star Wars", category:"Sci-Fi", director:"George Lucas", length:124, cost:24.99f);
        DigitalVideoDisc dvd3 = new DigitalVideoDisc(title:"Aladdin", category:"Animation", director:"Ron Clements", length:90, cost:18.99f);

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

        // Hiển thị danh sách DVD trong cửa hàng
        store.displayStore();

        // Xóa DVD khỏi cửa hàng
        store.removeDVD(dvd2);

        // Hiển thị lại danh sách DVD sau khi xóa
        store.displayStore();

        // Thử xóa DVD không tồn tại
        DigitalVideoDisc dvdNotExist = new DigitalVideoDisc(title:"Frozen");
        store.removeDVD(dvdNotExist);
    }
}

```

- Kết quả chạy:

```

The disc "The Lion King" has been added to the store.
The disc "Star Wars" has been added to the store.
The disc "Aladdin" has been added to the store.
Items in the store:
1. DigitalVideoDisc [ID=1, Title=The Lion King, Category=Animation, Director=Roger Allers, Length=88 mins, Cost=$19.99]
2. DigitalVideoDisc [ID=2, Title=Star Wars, Category=Sci-Fi, Director=George Lucas, Length=124 mins, Cost=$24.99]
3. DigitalVideoDisc [ID=3, Title=Aladdin, Category=Animation, Director=Ron Clements, Length=90 mins, Cost=$18.99]
Total items in store: 3
The disc "Star Wars" has been removed from the store.
Items in the store:
1. DigitalVideoDisc [ID=1, Title=The Lion King, Category=Animation, Director=Roger Allers, Length=88 mins, Cost=$19.99]
2. DigitalVideoDisc [ID=3, Title=Aladdin, Category=Animation, Director=Ron Clements, Length=90 mins, Cost=$18.99]
Total items in store: 2

```

## 8. Re-organize your projects

- AimsProject

hust.soict.ep.aims.disc.DigitalVideoDisc

hust.soict.ep.aims.cart.Cart hust.soict.

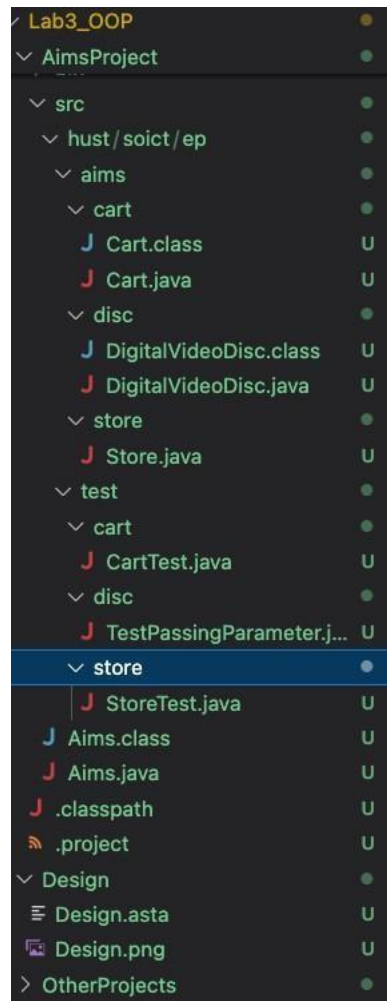
ep.aims.store.Store hust.soict.ep.aims.Aims

hust.soict.ep.test.cart.CartTest

hust.soict.ep.test.store.StoreTest

hust.soict.ep.test.disc.TestPassingParamete r





+ OtherProjects hust.soict.ep.lab01



## 9. String, StringBuilder and StringBuffer

- Tạo lớp ConcatenationInLoops

```

1 package hust.soict.ep.garbage;
2 import java.util.Random;
3 import static java.lang.System.currentTimeMillis; //Lam cho code ngan gon hon thay vi goi nguyen phuong thuc ay
4
5 public class ConcatenationInLoops {
6     public static void main(String[] args) {
7         Random r = new Random(seed:123);
8         long start = currentTimeMillis();
9         String s = "";
10        for (int i = 0; i < 65536; i++) s += r.nextInt(bound:2);
11        System.out.println(currentTimeMillis() - start);
12
13        r = new Random(seed:123);
14        start = System.currentTimeMillis();
15        StringBuilder sb = new StringBuilder();
16        for(int i = 0; i < 65536; i++)
17            sb.append(r.nextInt(bound:2));
18        s += sb.toString();
19        System.out.println(System.currentTimeMillis() - start);
20    }
}

```

- Tạo lớp GarbageCreator

```

1 package hust.soict.ep.garbage;
2 import java.io.IOException;
3 import java.nio.file.Files;
4 import java.nio.file.Paths;
5
6 public class GarbageCreator {
7     public static void main(String[] args) throws IOException {
8         String filename = "/Users/macbook/Library/CloudStorage/OneDrive-HanoiUniversityofScienceandTechnology/TAI
9         byte[] inputBytes = { 0 };
10        long startTime, endTime;
11        inputBytes = Files.readAllBytes(Paths.get(filename));
12        startTime = System.currentTimeMillis();
13        String outputString = "";
14        for(byte b : inputBytes) {
15            outputString += (char)b;
16        }
17        endTime = System.currentTimeMillis();
18        System.out.println(endTime - startTime);
19    }
20 }

```

- Tạo lớp NoGarbage

```

1 package hust.soict.ep.garbage;
2 import java.io.IOException;
3 import java.nio.file.Files;
4 import java.nio.file.Paths;
5
6 public class NoGarbage {
7     public static void main(String[] args) throws IOException {
8         String filename = "G:\\00PLab\\IT3103.732871.2023.1.20215060.DoGiaHuy\\Lab02_03_04_05\\OtherProject\\src\\
9         byte[] inputBytes = { 0 };
10        long startTime, endTime;
11
12        inputBytes = Files.readAllBytes(Paths.get(filename));
13        startTime = System.currentTimeMillis();
14        StringBuilder outputStringBuilder = new StringBuilder(str:"");
15        for(byte b : inputBytes) {
16            outputStringBuilder.append((char)b);
17        }
18        endTime = System.currentTimeMillis();
19        System.out.println(endTime - startTime);
20    }
}

```

## 10. Cập nhật các Diagram

