***,***

Contents

[Use case diagram 3](#_Toc151761334)

[Class diagram 4](#_Toc151761335)

[1.Branch your repository 4](#_Toc151761336)

[2. Working with method overloading 6](#_Toc151761337)

[2.1 Overloading by differing types of parameter 6](#_Toc151761338)

[2.2. Overloading by differing the number of parameters 7](#_Toc151761339)

[3. Passing paramete 7](#_Toc151761340)

[4.Use debug run 8](#_Toc151761341)

[4.2. Example of debug run for the swap method of TestPassingParameter 8](#_Toc151761342)

[4.2.1. Setting, deleting & deactivate breakpoints: 8](#_Toc151761343)

[4.2.2. Run in Debug mode 9](#_Toc151761344)

[5. Classifier Member and Instance Member 9](#_Toc151761345)

[6. Open the Cart class 9](#_Toc151761346)

[7. Implement the Store class 11](#_Toc151761347)

[12](#_Toc151761348)

[8. Re-organize your projects 13](#_Toc151761349)

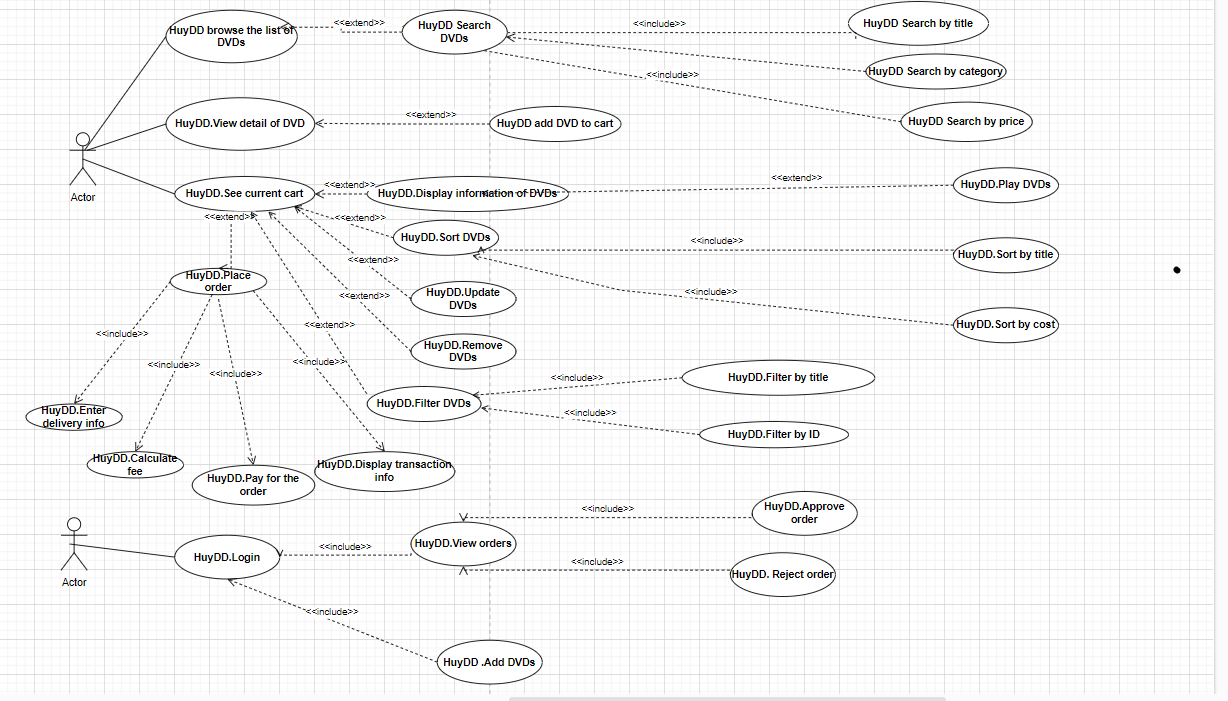
[9. String, StringBuilder and StringBuffer 15](#_Toc151761350)

[10.Release flow demonstration 16](#_Toc151761351)

***BÁO CÁO THỰC HÀNH LAP 3***

***LAP TRÌNH HƯỚNG ĐỐI TƯỢNG***

# Use case diagram



**Mối quan hệ:**

- Include: quan hệ bắt buộc phải có giữa các use case với nhau.

- Extend: quan hệ mở rộng giữa các use case với nhau.

# Class diagram

A screenshot of a computer

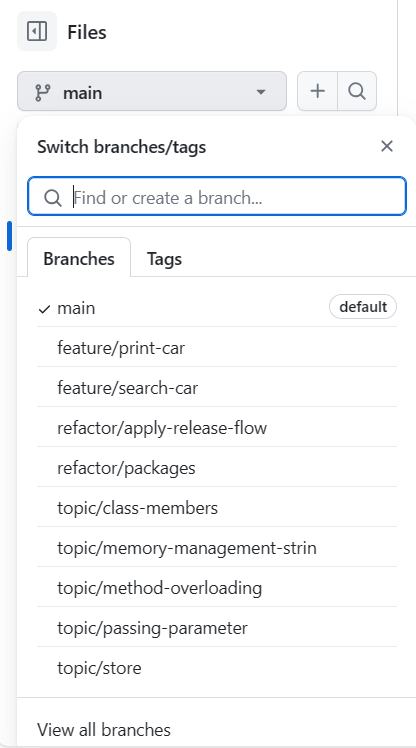
Description automatically generated

**Mối quan hệ:**

Quan hệ kế thừa : Đối tượng cụ thể sẽ kế thừa các thuộc tính và phương thức của đối tượng tổng quát .

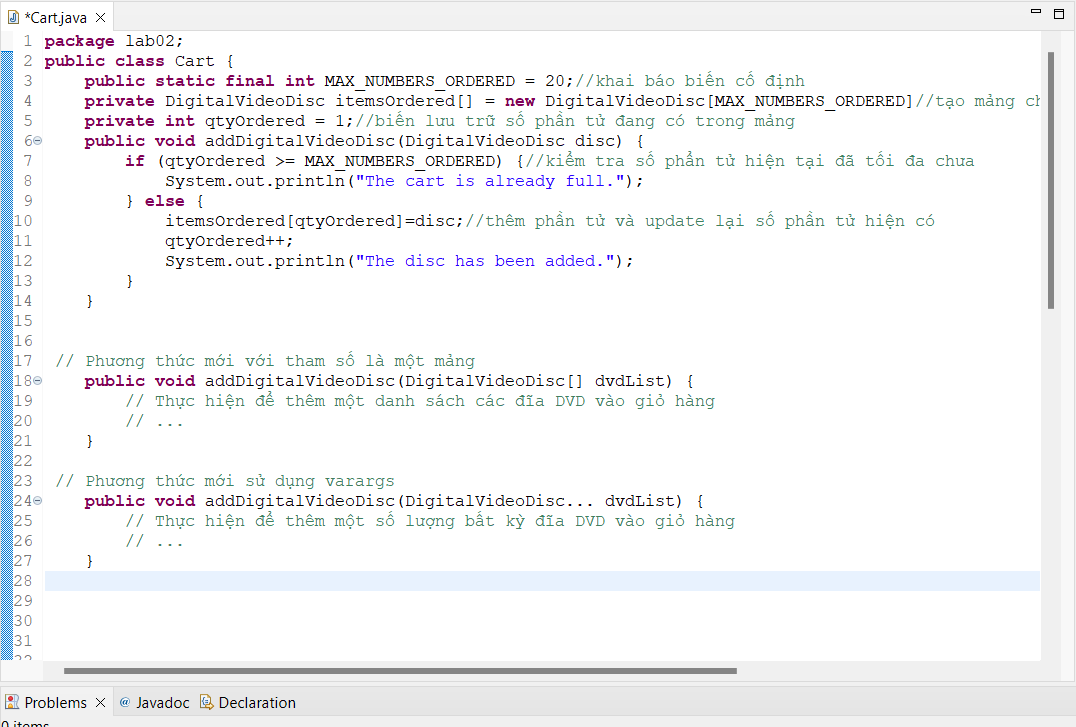
Quan hệ phụ thuộc : quan hệ giữa 2 phần tử trong mô hình mà thay đổi ở phần tử này (phần tử độc lập) có thể gây ra thay đổi ở phần tử kia (phần tử phụ thuộc).

# 1.Branch your repository



# 2. Working with method overloading

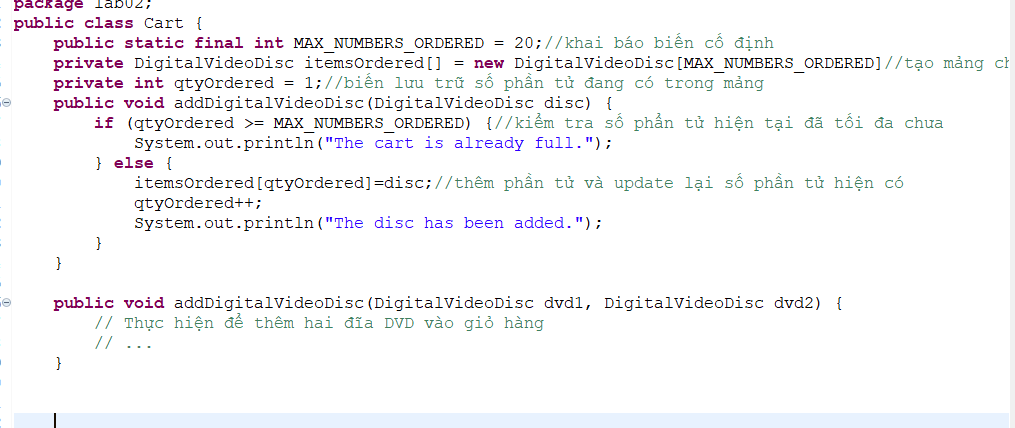
## 2.1 Overloading by differing types of parameter



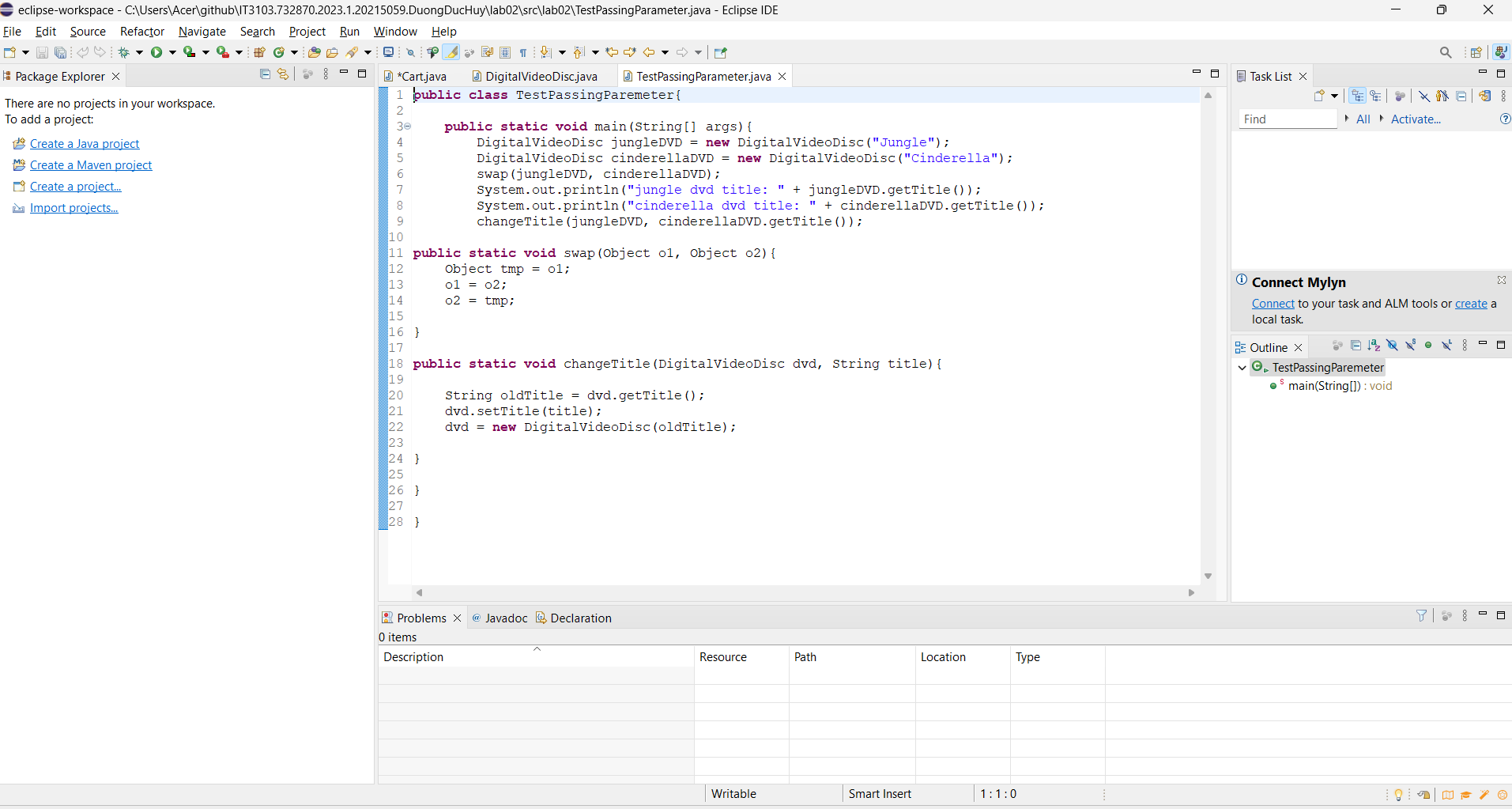
|  |  |  |
| --- | --- | --- |
| So sánh | Varargs | Tham số là Mảng |
| Ưu điểm | Cú pháp sạch sẽ cho người gọi, cho phép họ truyền từng đối số một mà không cần tạo một mảng. | Rõ ràng cho thấy bạn mong đợi một mảng các mục.  Cho phép bạn truyền một mảng đã được định nghĩa trước. |
| Nhược điểm | Cú pháp sạch sẽ cho người gọi, cho phép họ truyền từng đối số một mà không cần tạo một mảng. | Yêu cầu người gọi phải tạo một mảng trước khi truyền vào phương thức. |

Tôi ưu tiên varargs hơn bởi cú pháp sạch sẽ cho người gọi và tính linh hoạt về số lượng đối số

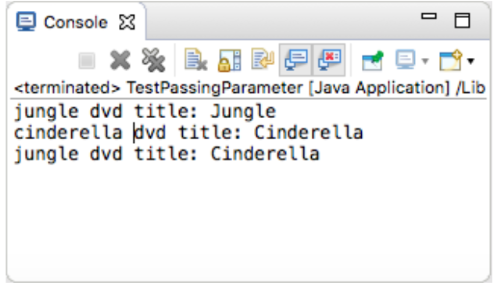
## 2.2. Overloading by differing the number of parameters



# 3. Passing paramete



Kết quả



**Java là ngôn ngữ lập trình "pass by value" và không hỗ trợ trực tiếp "pass by reference".**

***Sau lời gọi swap(jungleDVD, cinderellaDVD), tại sao title của hai đối tượng vẫn giữ nguyên?***

*Khi gọi phương thức swap(jungleDVD, cinderellaDVD), các giá trị của hai đối tượng jungleDVD và cinderellaDVD được truyền vào phương thức swap. Tuy nhiên, vì Java truyền tham số theo giá trị, nghĩa là một bản sao của giá trị tham số được tạo và truyền vào phương thức, nên bất kỳ sự thay đổi nào được thực hiện trên các bản sao này không ảnh hưởng đến đối tượng gốc. Do đó, sau khi gọi swap, title của jungleDVD và cinderellaDVD vẫn giữ nguyên vì swap chỉ hoạt động trên các bản sao của chúng*.

***Sau lời gọi changeTitle(jungleDVD, cinderellaDVD.getTitle()), tại sao title của JungleDVD bị thay đổi?***

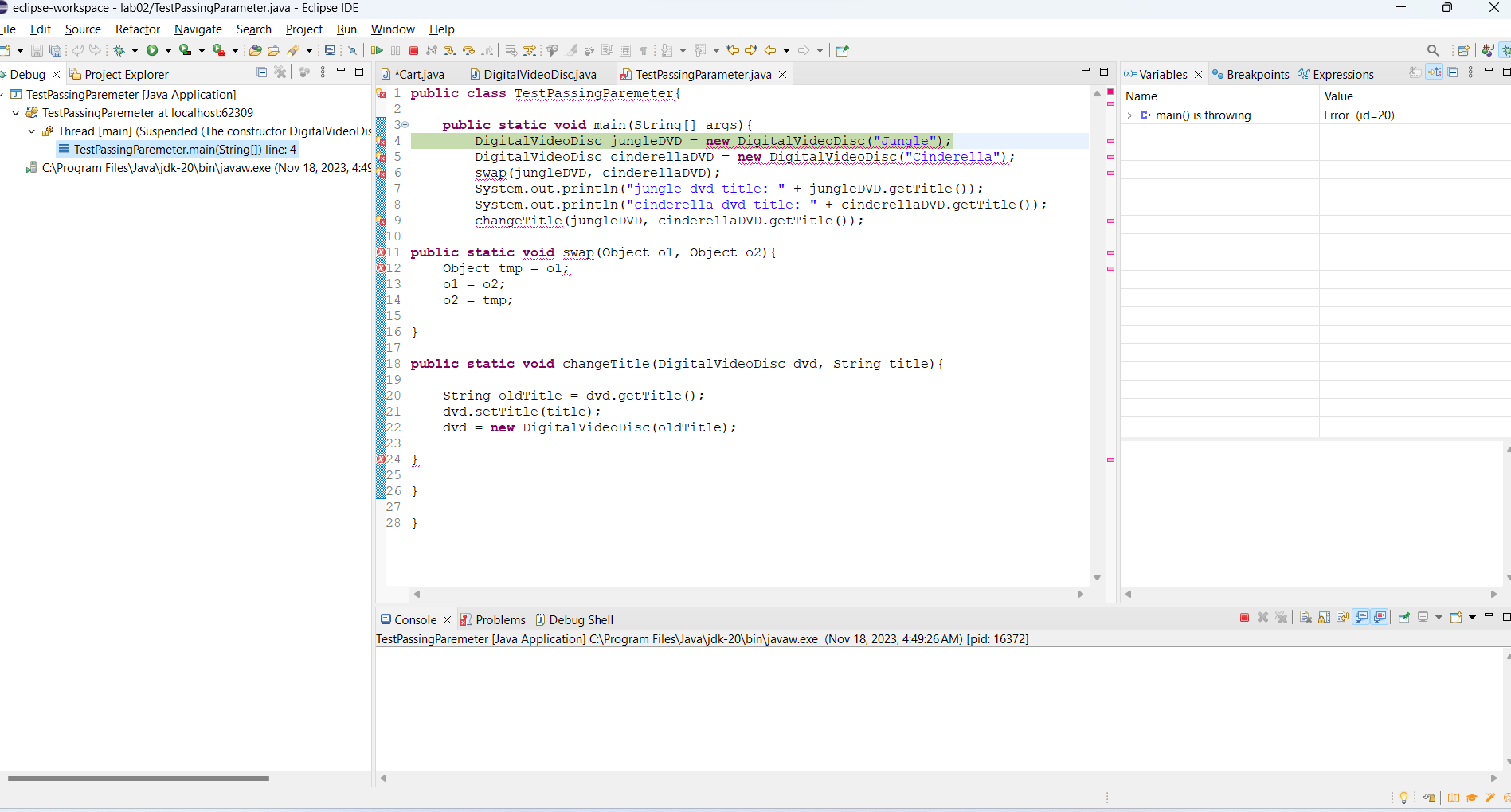
*Trong lời gọi này, thay vì truyền cả đối tượng cinderellaDVD vào phương thức changeTitle, chỉ title của đối tượng đó được truyền vào. Phương thức changeTitle có thể thực hiện thay đổi trực tiếp lên title của đối tượng được tham chiếu bởi jungleDVD. Vì vậy, khi gọi changeTitle, nó ảnh hưởng trực tiếp đến title của đối tượng jungleDVD.*

# 4.Use debug run

4.2. Example of debug run for the swap method of TestPassingParameter

4.2.1. Setting, deleting & deactivate breakpoints:

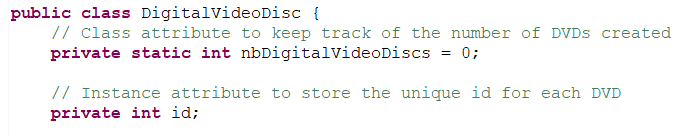
### 4.2.2. Run in Debug mode



# 5. Classifier Member and Instance Member

Tạo một thuộc tính lớp có tên là "nbDigitalVideoDiscs" trong lớp DigitalVideoDis

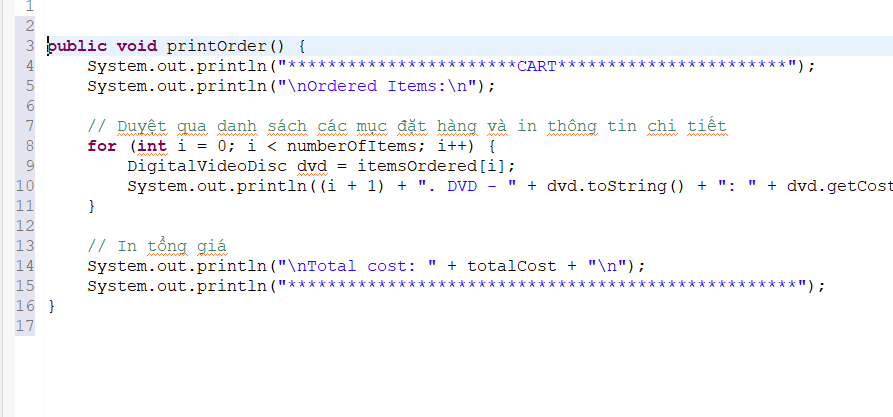
Tạo một thuộc tính thể hiện có tên "id" trong lớp DigitalVideoDisc



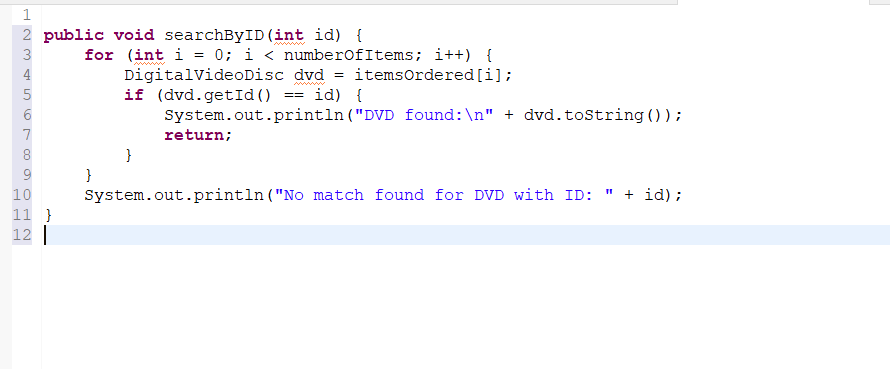
# 6. Open the Cart class

Viết các phương thức mới để thực hiện các chức năng sau:

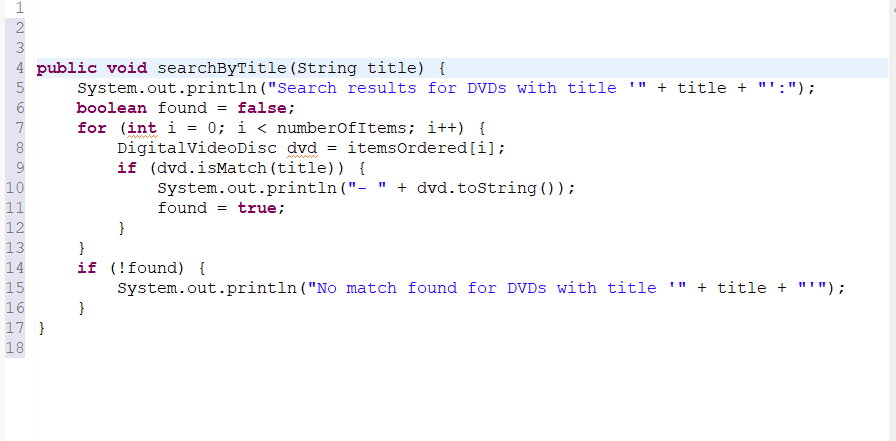
- Tạo phương thức mới để in danh sách các mặt hàng đã đặt của giỏ hàng, giá từng mặt hàng và tổng giá. Định dạng phác thảo như sau



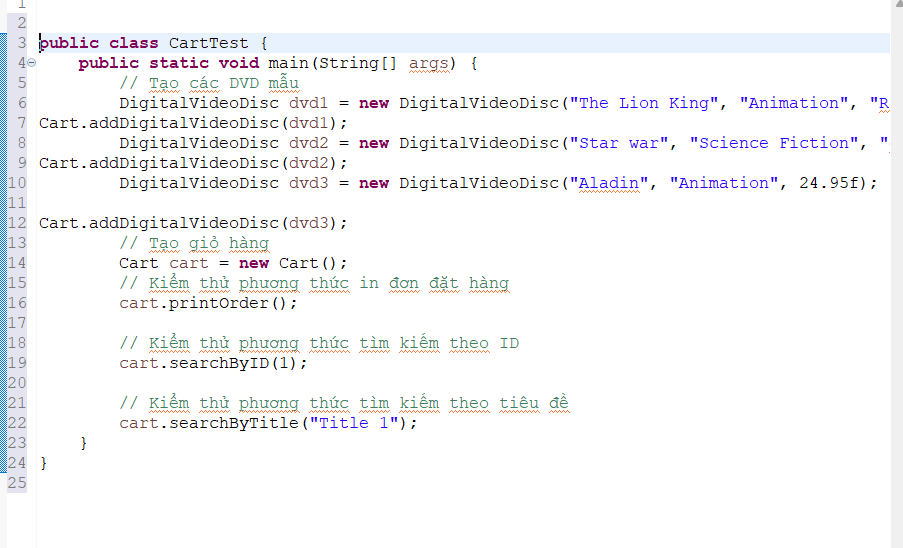
b. Tìm kiếm DVD trong giỏ hàng bằng ID và hiển thị kết quả tìm kiếm. Đảm bảo thông báo cho người dùng nếu không tìm thấy kết quả.



c. Tìm kiếm DVD trong giỏ hàng bằng tiêu đề và in kết quả tìm kiếm. Đảm bảo thông báo cho người dùng nếu không tìm thấy kết quả



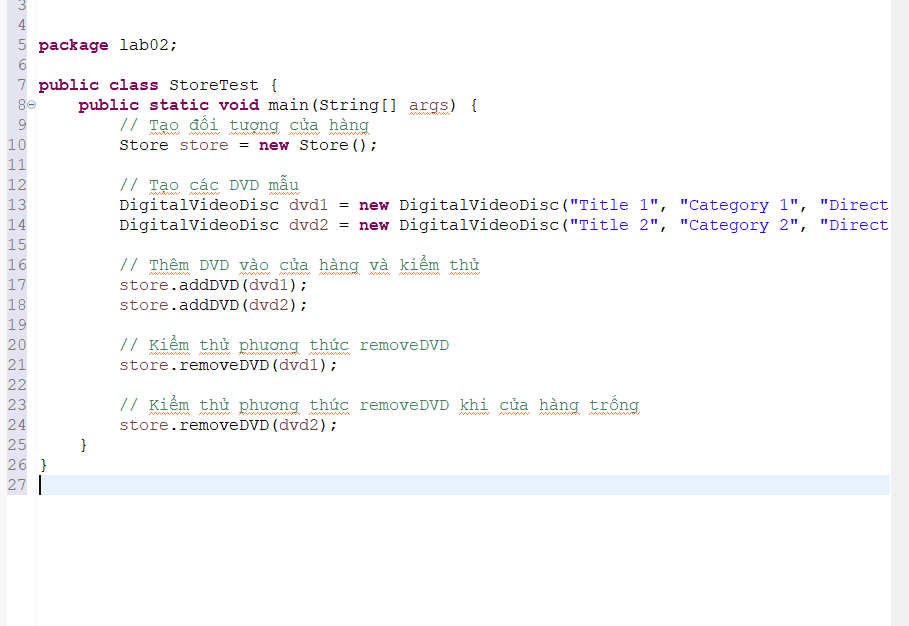
d. Trong lớp CartTest, viết mã kiểm thử cho tất cả các phương thức đã viết trong bài tập này.



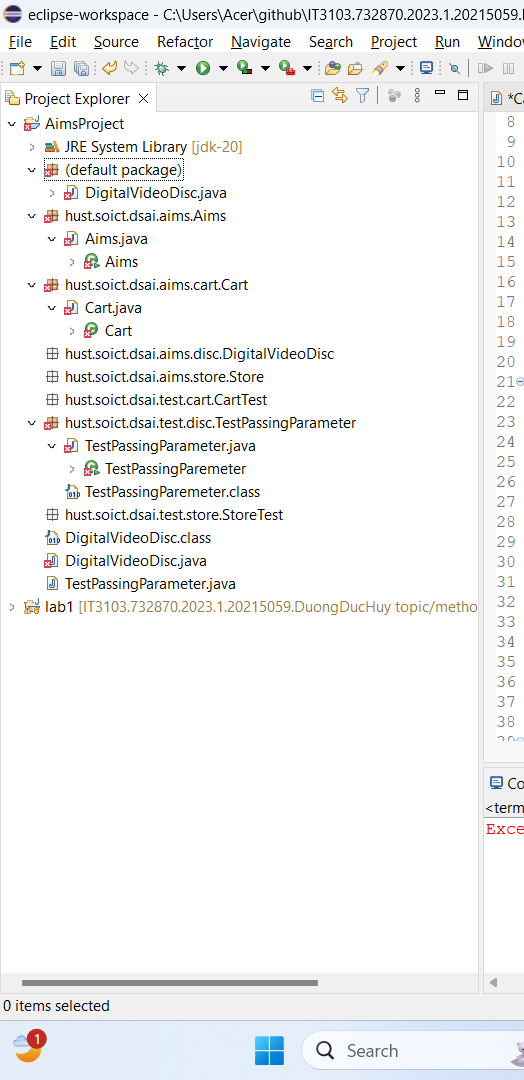
# 7. Implement the Store class

**Lớp Store**

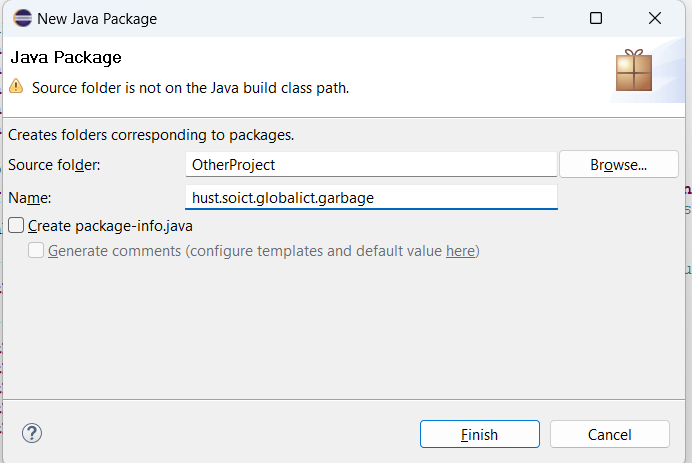
### 

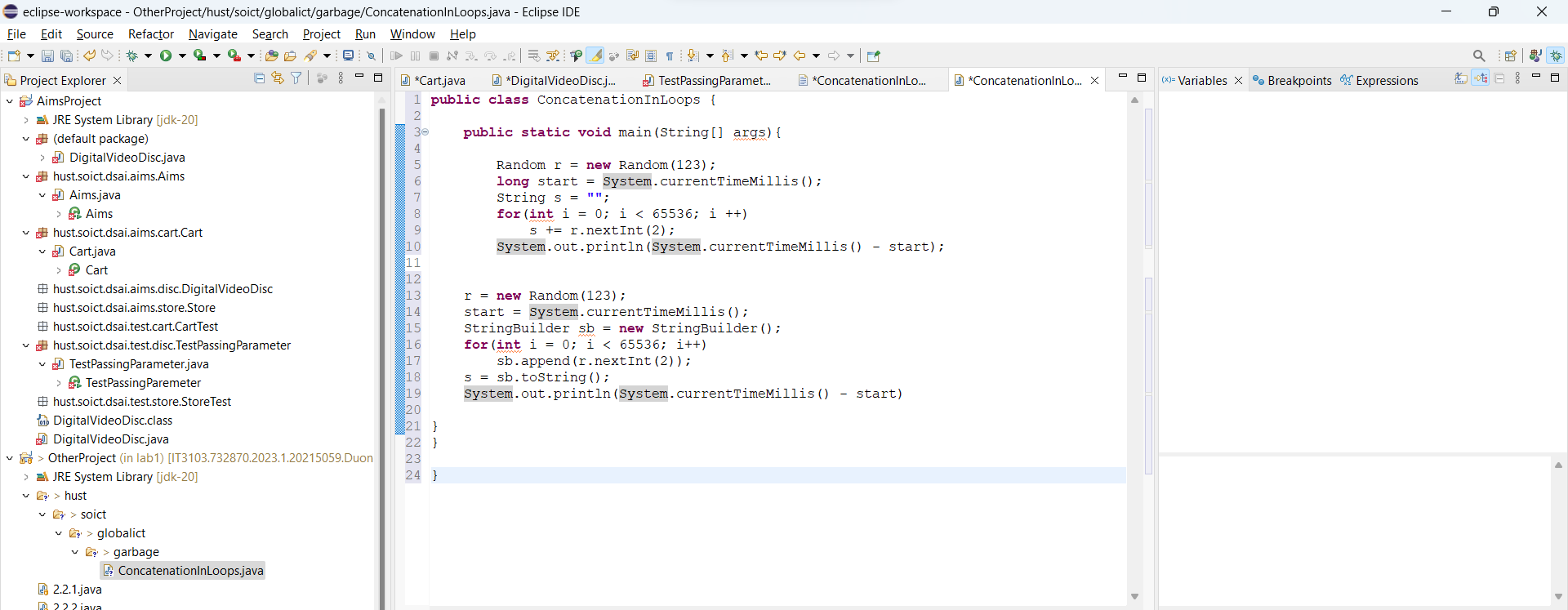


# 8. Re-organize your projects

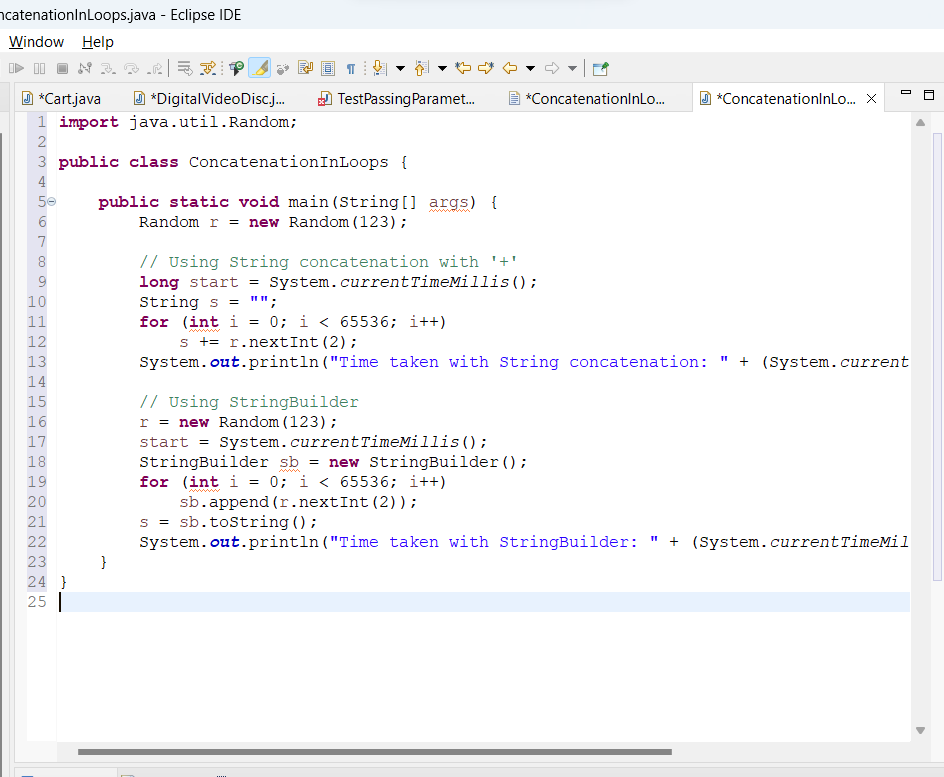


# 9. String, StringBuilder and StringBuffer



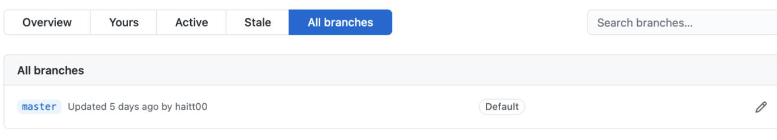


tạo "garbage" và giải quyết vấn đề bằng cách sử dụng **StringBuilder**:



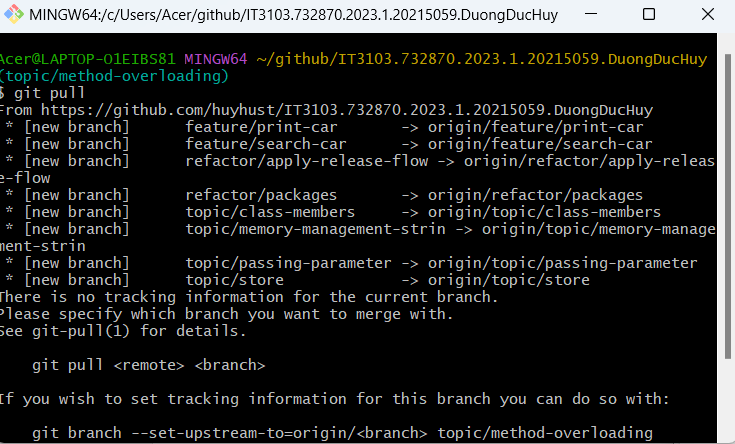
# 10.Release flow demonstration

10.1. Hypothesis

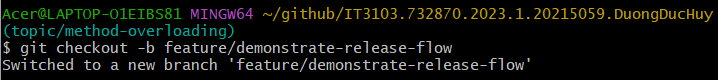


10.2. Demonstration

**Step 1. Update local repository**



**Step 2. Create and switch to a new branch in the local repository.**



**Step 4. Commit the change in the local repository.**

