**BÁO CÁO THỰC HÀNH LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG – LAB 03**

**Họ và tên: Trần Đại Hiệp**

**MSSV: 20226081**

# 1. Branch your repository

# A screenshot of a computer Description automatically generated

# 2. Working with method overloading

# 2.1. Overloading by differing types of parameter

# You will overload the method addDigitalVideoDisc you created last time

# A screen shot of a computer code Description automatically generated

# Try to add a method addDigitalVideoDisc which allows an arbitrary number of arguments for dvd

# A screen shot of a computer Description automatically generated

# What do you prefer in this case?

# Answers: I’d prefer using the latter variant because

# It can pass an arbitrary number of arguments (in this case it’s dvd object) into the method

# This is more flexible since we don’t have to pass an entire list like the former implementation, making the code cleaner, easier to write for programmers and software developers.

# 2.2. Overloading by differing types of parameter

# 

# 3. Passing parameter

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

## - Answer: Java is strictly pass-by-value. For primitive types, the actual value is passed, so changes in the method don’t affect the original variable. For objects, the reference (a copy of the memory address) is passed, so while you can modify the object's fields, you cannot reassign the original reference.

## 3.1. Create class TestPassingParameter

## 

## Result:

## A screen shot of a computer Description automatically generated

## Questions:

## After the call of swap(jungleDVD, cinderellaDVD) why does the title of these two objects still remain?

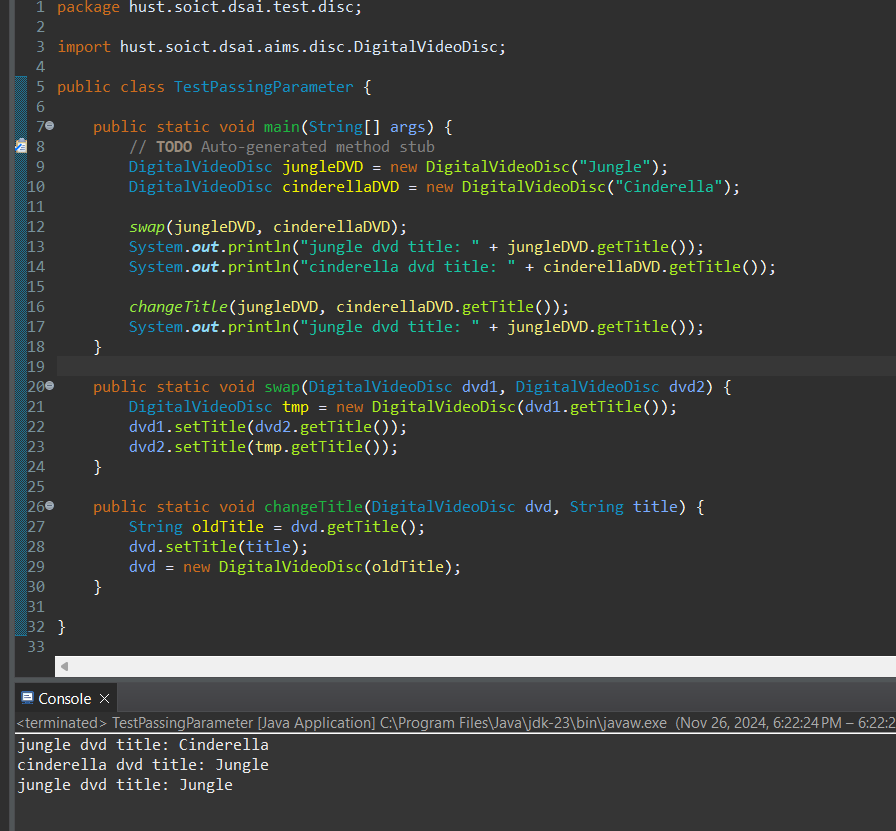
## After the call of changeTitle(jungleDVD, cinderellaDVD.getTitle()) why is the title of the JungleDVD changed?

## Answer:

## In Java, the parameters passed to a method are the values of the references to the objects, not the actual references themselves. When you change the value of a parameter inside the method, it does not affect the original object references outside the method.

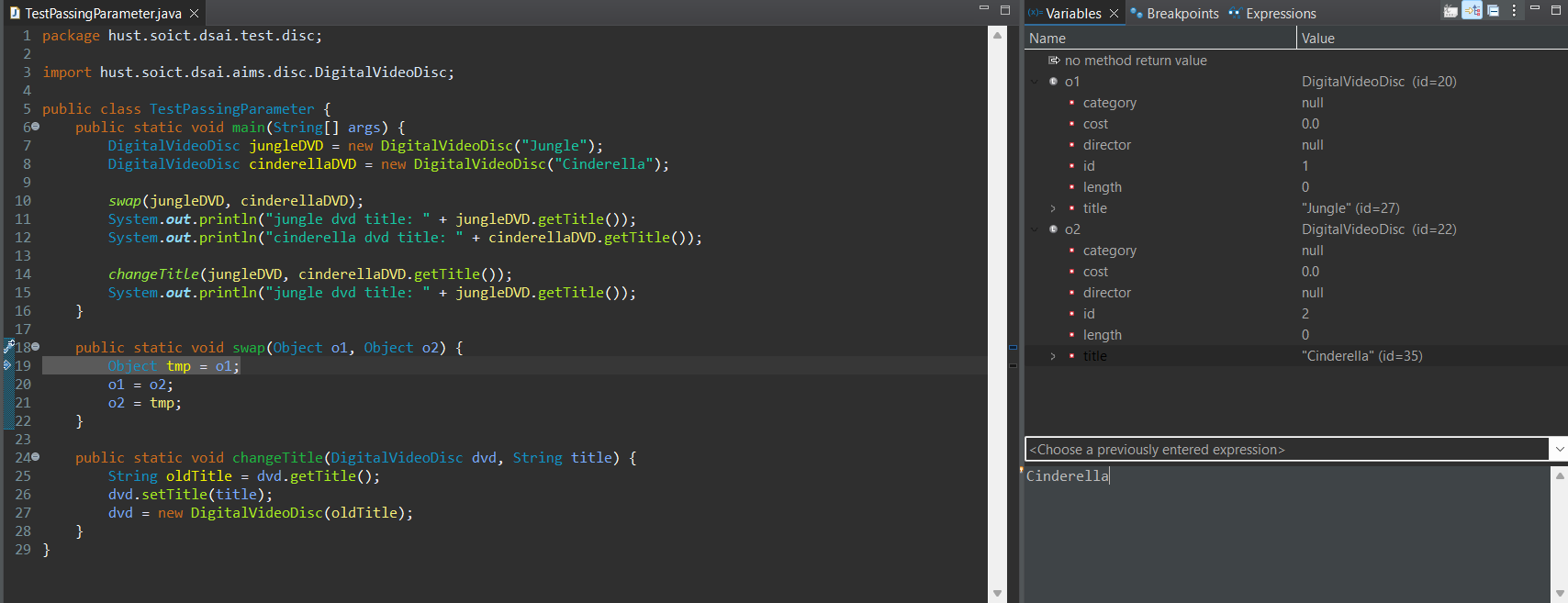
## In the changeTitle method, the title of the object dvd is directly modified by calling dvd.setTitle(title). This operation updates the original object because the method is working on the object that the reference points to.

## 3.2 Fix swap



# 4. Use Debug run

# 4.1 Investigate value of variables



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# 4.2 Change value of variables

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# 5. Classifier Member and Instance Member

# Create a class attribute named “nbDigitalVideoDiscs” in “DigitalVideoDisc” class

# 

# Each time an instance of the DigitalVideoDisc class is created, the “nbDigitalVideoDisc” should be updated. Therefore, you should update the value for this class variable inside the constructor method and assign the appropriate value for the id

# 

# 6. Open the Cart class

# Search for DVDSs 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

# 

# \* Additional implementation inside DigitalVideoDisc class

# 

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

# ToString method in DigitalVideoDisc

# 

# 

# Results in CartTest.java:

# 

# 7. Implement the Store class

# Create a Store class, which contains the one attribute itemsInStore[] - an array of DVDs available in the store

# To add and remove DVDs from the store, implement two methods called addDVD and removeDVD

# 

# 

# Test these two methods in the StoreTest class:

# 

# 8. Re-organize your projects

# 

# 9. String, StringBuilder and StringBuffer

# Create a new class ConcatenationInLoops to test the processing time to construct String using + operator, StringBuffer and StringBuilder

# 

# Results:

# 

# Create a new class GarbageCreator. Create “garbage” as much as possible and observe when you run a program (it should let the program hang or even stop working when there is too much “garbage”). Write another class NoGarbage to solve the problem

# 

# NoGarbage Class implementation:

# 

# 10. Release flow

# 11. Update Diagrams

## 1. Class Diagram

## 

## 2. Usecase Diagram

