Object-Oriented Programming

Lecturer:

Teaching Assistant:

**Lab 03: Basic Object-Oriented Techniques**

## 2.1 Overloading by differing types of parameter

**+** Add a list of DVDs to the current cart.

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

+ Pass an arbitrary number of arguments for dvd

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

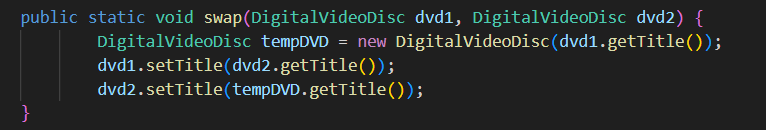
Mô tả được tạo tự động

**Prefer varargs methods → flexibility and ease of use in this case**

## 2.2. Overloading by differing the nu **DigitalVideoDisc** mber of parameters

# Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ Mô tả được tạo tự động

3. Passing parameter



Question:

* After the call of **swap(jungleDVD, cinderellaDVD)** why does the title of these two objects still remain?
* Java is a pass-by-value programming language. When objects are passed as parameters to a method:

The value of the reference (memory address) is copied into the method’s parameter.

Inside the method, the parameter points to the same object as the original reference, but modifying the parameter does not affect the original reference itself.

* After the call of **changeTitle(jungleDVD, cinderellaDVD.getTitle())** why is the title of the JungleDVD changed?
* Although Java is pass-by-value, the value passed for an object reference allows access to the original object. When one modifies the attributes of the object using this reference, the changes persist because the reference points to the same memory location.

**Please write a swap() method that can correctly swap the two objects.**

1. Use debug run:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, máy tính

Mô tả được tạo tự động

1. Classifier Member and Instance Member

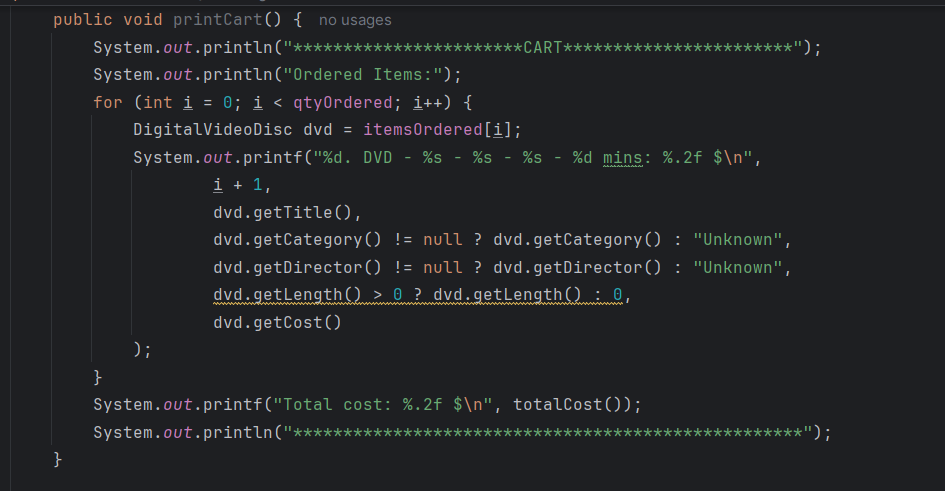
Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

1. Open the **Cart** class



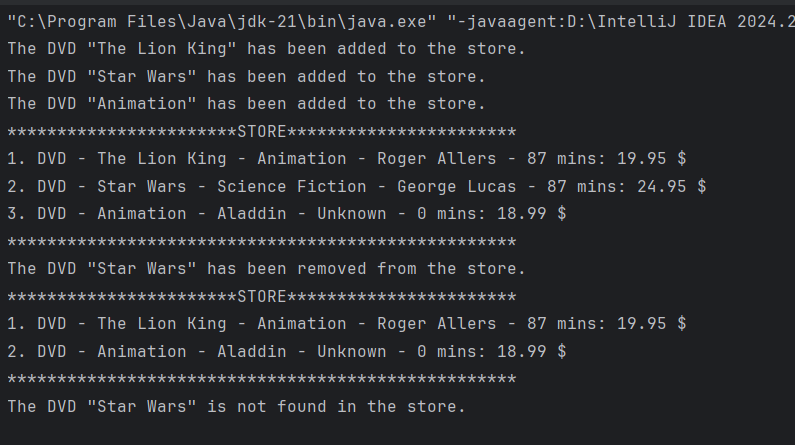
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

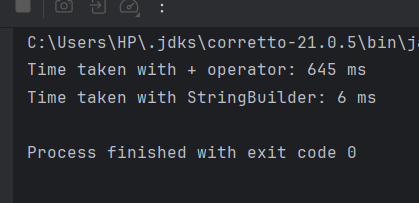
Mô tả được tạo tự động

Implement the **Store** class



Re-organize your projects

*String, StringBuilder and StringBuffer*

 Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

*Figure 22. ConcatenationInLoops*

Release flow demonstration

*10.1. Hypothesis*

We hypothesis that the Figure 25 shows the branches of our current remote repository.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

*Figure 25. Branches of Remote Repository*

Now we add a new topic or a new feature to our application. The next section shows us how to apply Release Flow in this hypothesis.

*10.2. Demonstration*