**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**\*\*\*\*\***

A red and white sign with a yellow and red stripe and a red and white flag with a yellow and red and white flag with a red and white flag with a yellow and red flag with a

Description automatically generated

**BÁO CÁO THỰC HÀNH LAB 1**

**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Giảng viên hướng dẫn**: Trịnh Tuấn Đạt

**Sinh viên thực hiện**: Phạm Công Hào

**MSSV**: 20215045

**Mã học phần**: IT3103

**Mã lớp học:** 744521

Mục lục

[1. Tạo lớp Aims 2](#_Toc180910992)

[2. Tạo lớp DigitalVideoDisc và các thuộc tính 3](#_Toc180910993)

[3. Tạo các phương thức set và get cho lớp DigitalVideoDisc 4](#_Toc180910994)

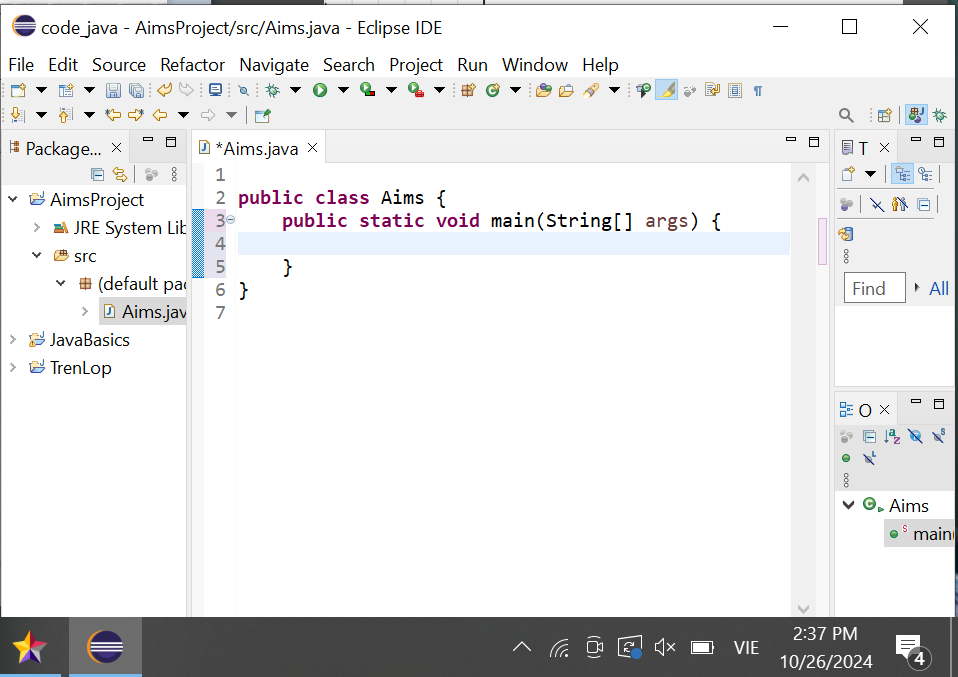
[4. Tạo lớp Cart để làm việc với DigitalVideoDisc 6](#_Toc180910995)

[5. Tạo giỏ hàng cho các DigitalVideoDisc 8](#_Toc180910996)

[6. Xóa DVD khỏi giỏ hàng 10](#_Toc180910997)

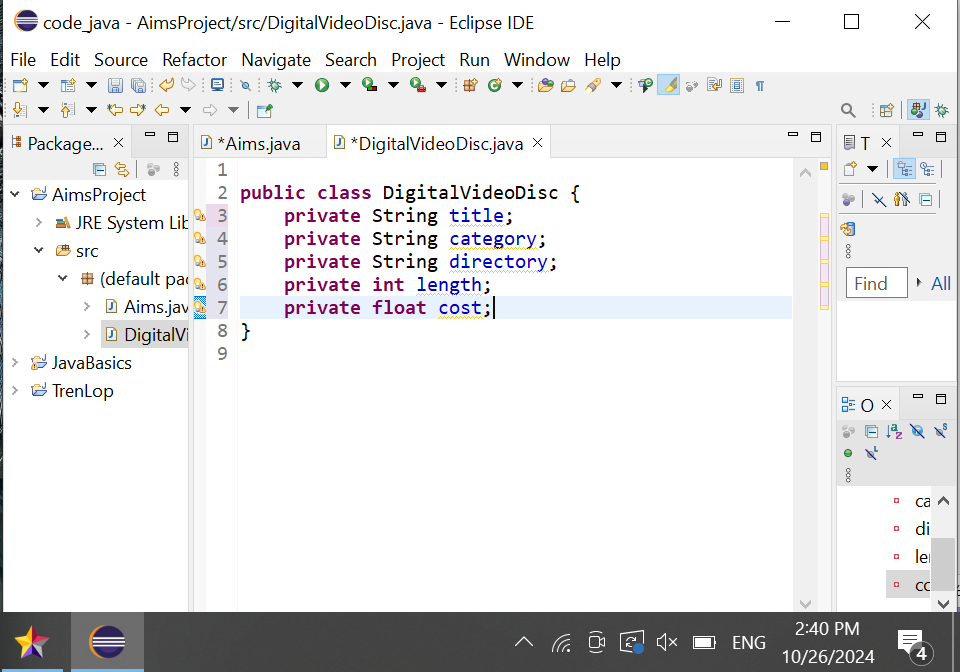
[7. Trả lời các câu hỏi 10](#_Toc180910998)

1. Tạo lớp Aims

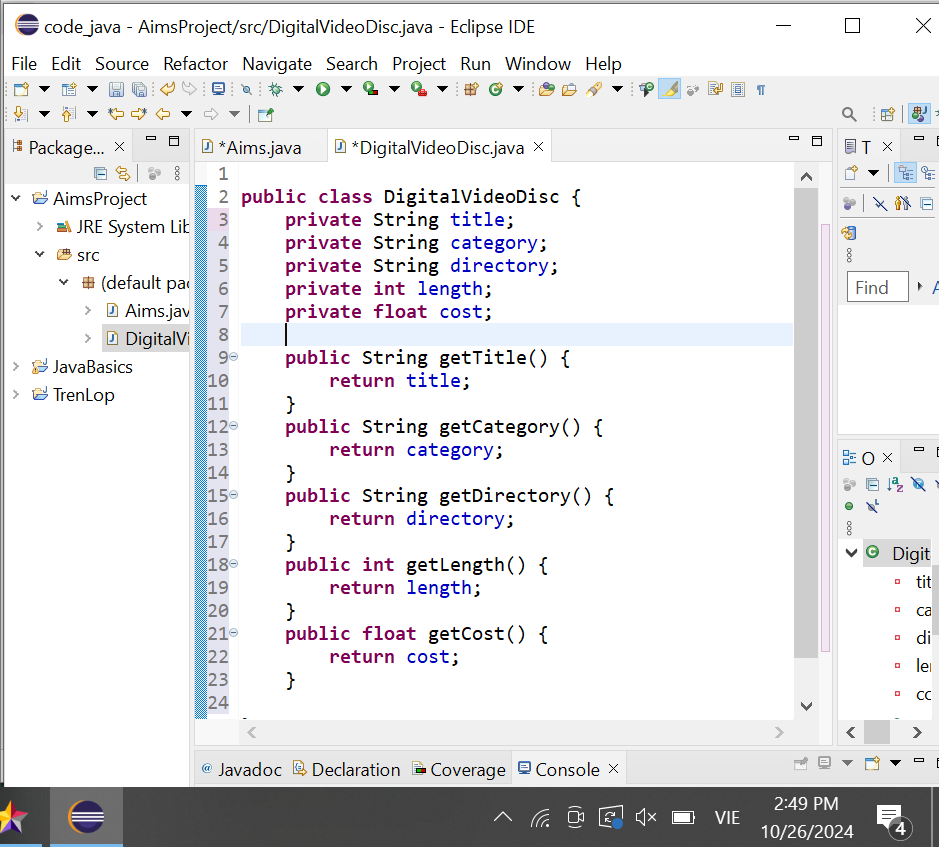
* Tạo lớp Aims trên Eclipse
* Thêm hàm main()  
  

1. **Tạo lớp DigitalVideoDisc và các thuộc tính**

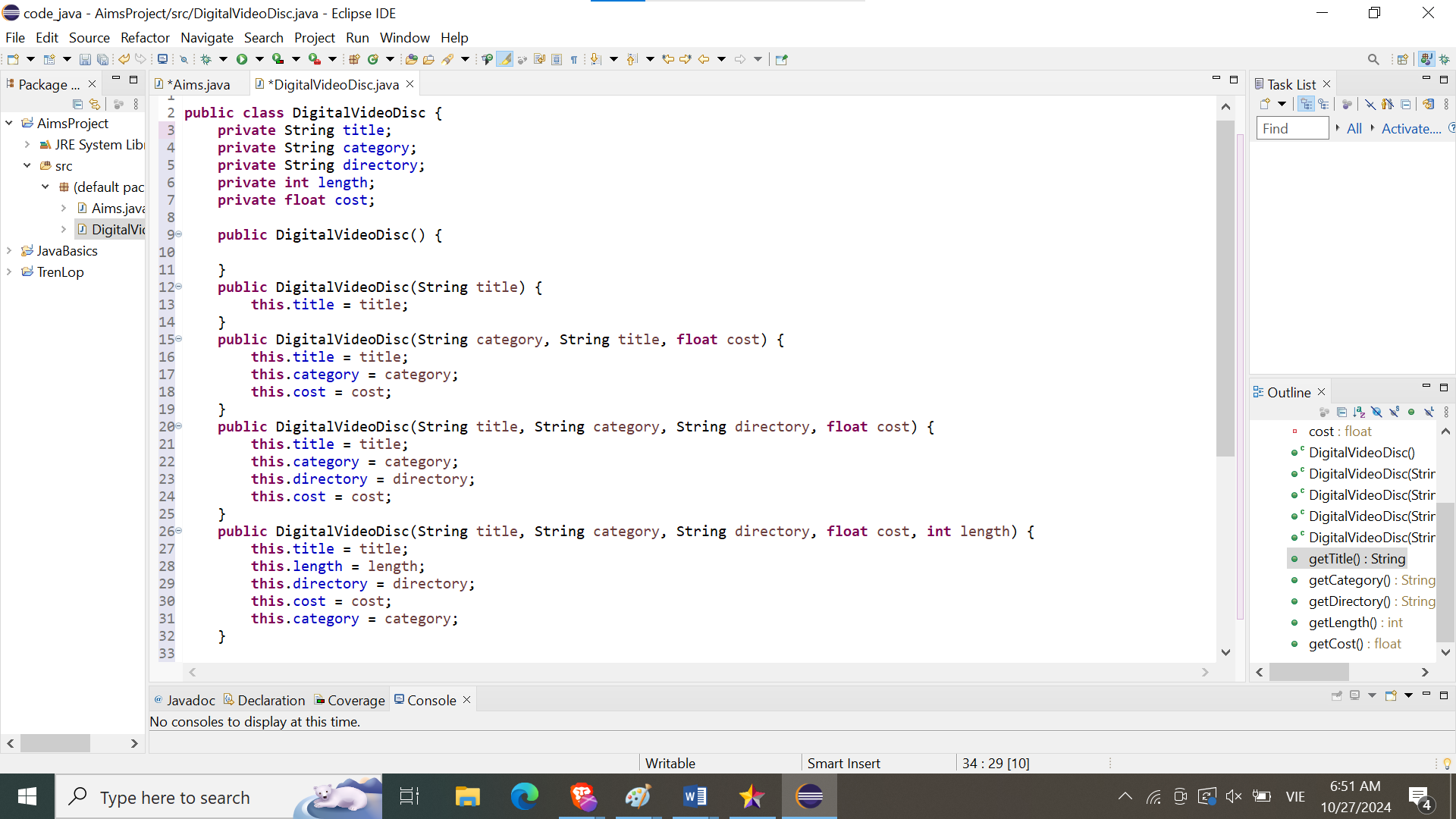
* Thêm các thuộc tính title, category, directory, length, cost cho lớp DigitalVideoDisc.

****

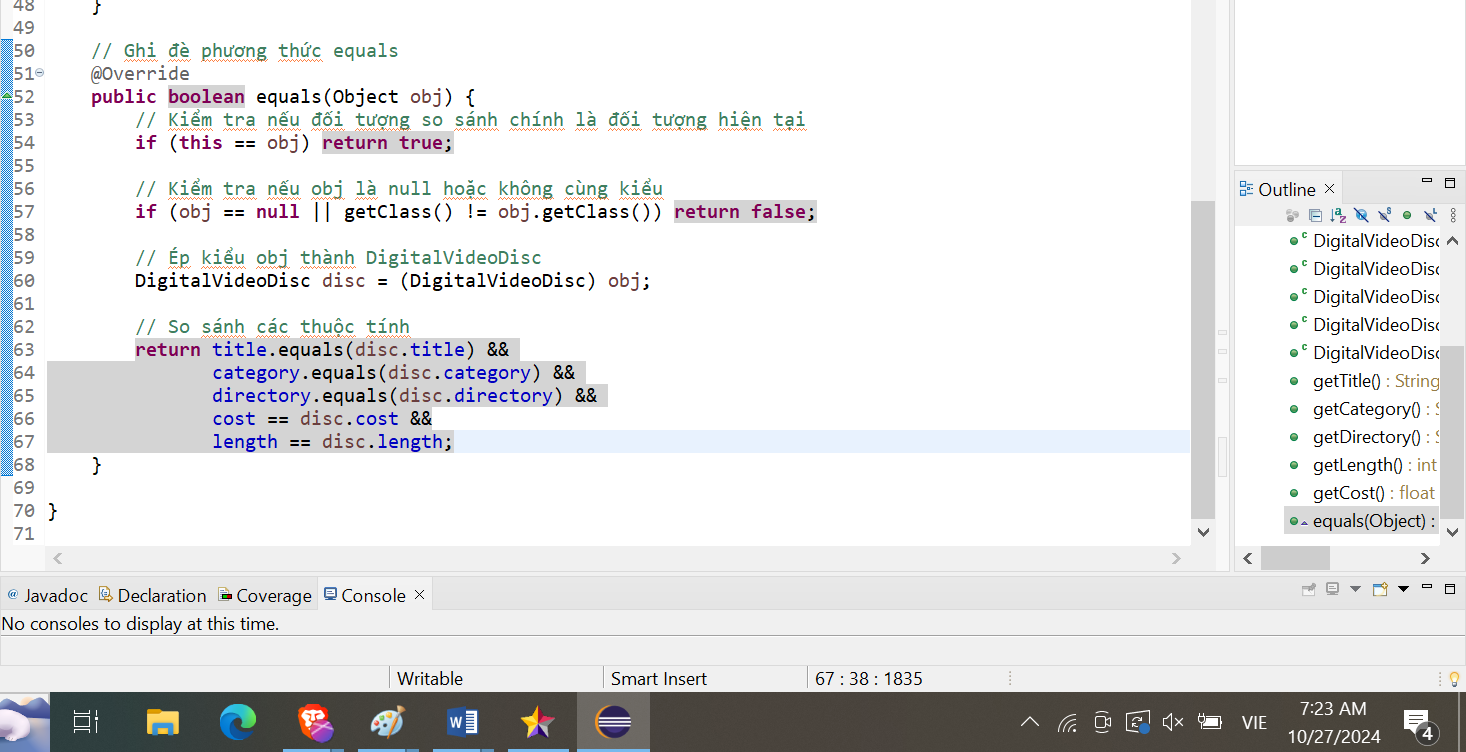
1. **Tạo các phương thức set và get cho lớp DigitalVideoDisc**

****

* Thêm các phương thức khởi tạo khác

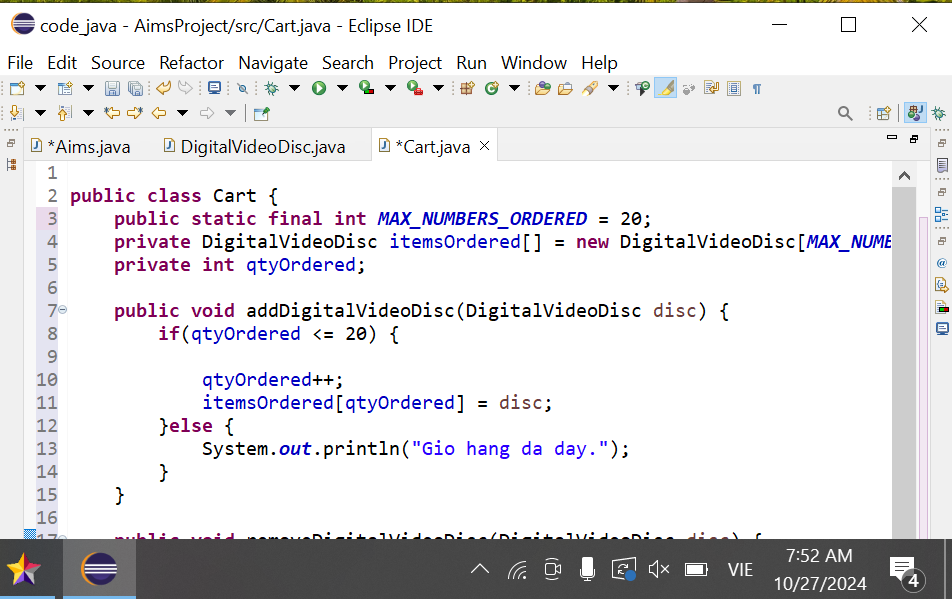


* Thêm phương thức equals để so sánh hai đối tượng.



1. **Tạo lớp Cart để làm việc với DigitalVideoDisc**

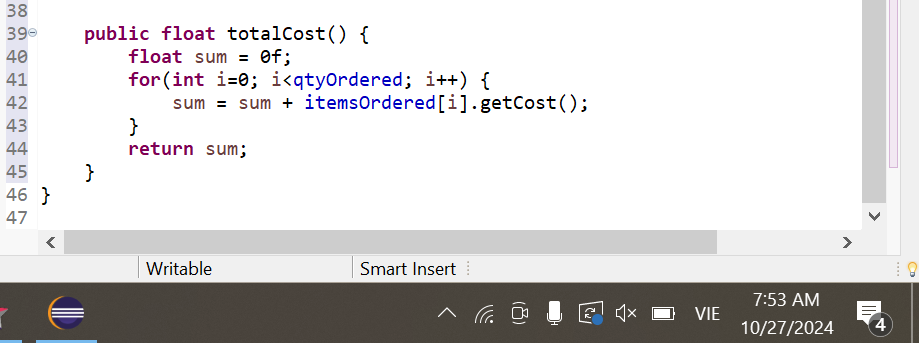
* Đặt MAX\_NUMBERS\_ORDERED = 20 số luọng DVD tối đa trong giỏ hàng.
* Thêm phương thức addDigitalVideoDisc(); thêm một DVD vào giỏ hàng



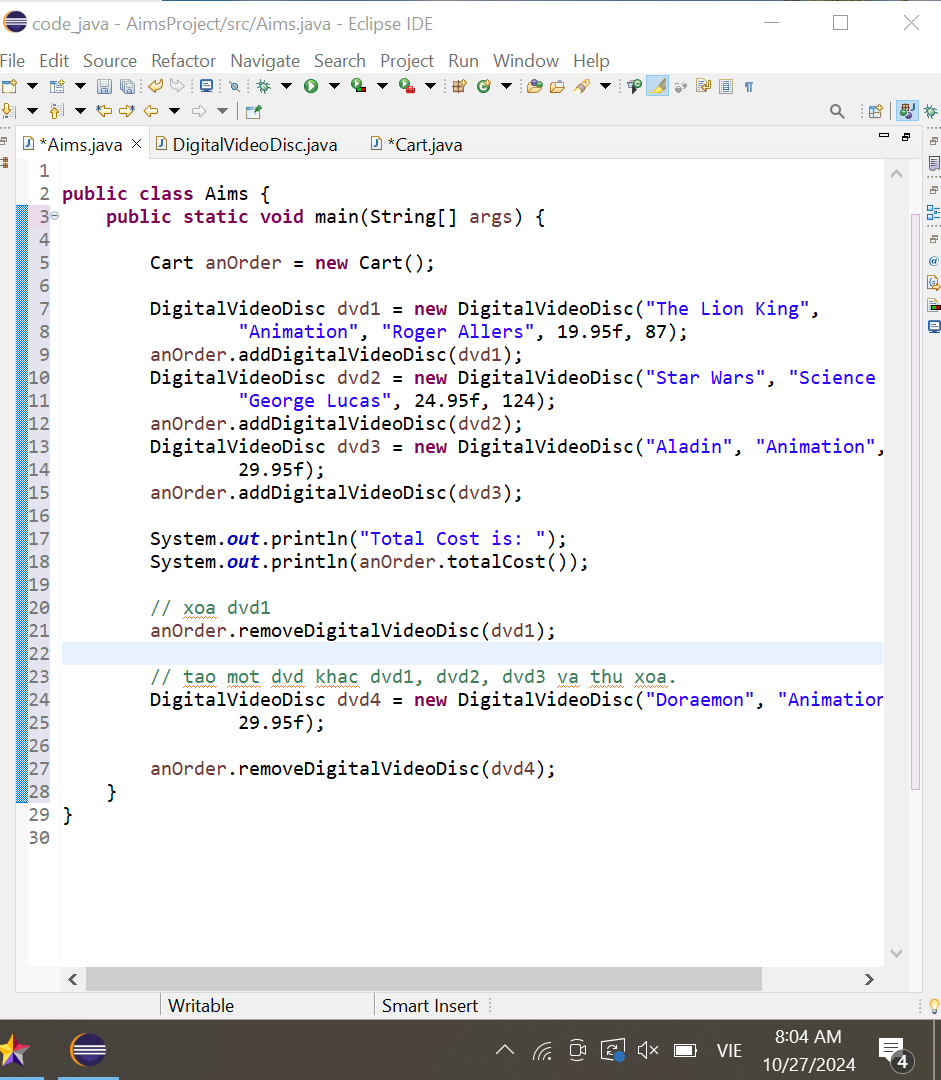
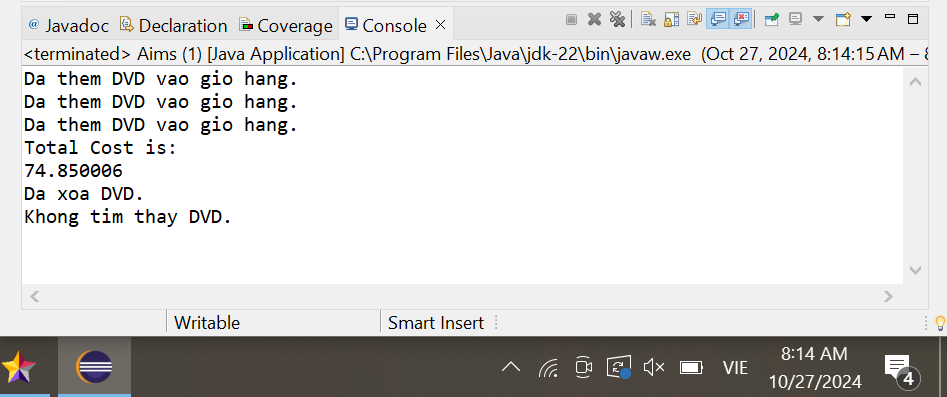
* Tạo phương thức removeDigitalVideoDisc(): xóa một disc khỏi giỏ hàng.



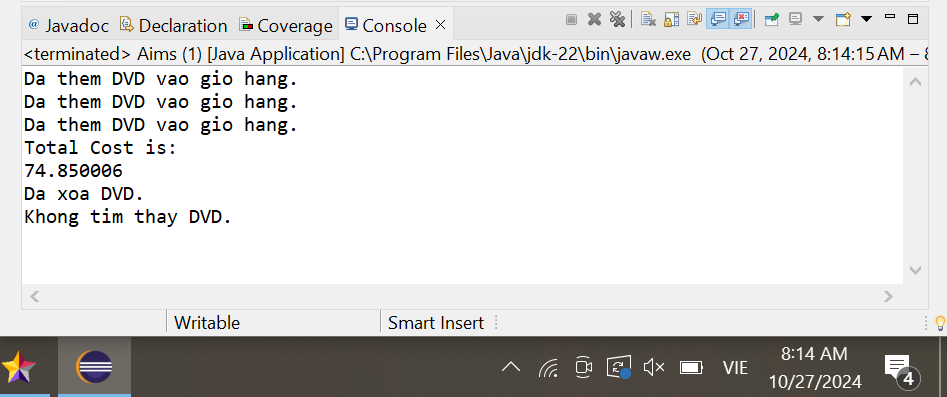
* Tạo phương thức totalCost(): trả về tổng giá trị các DVD trong giỏ hàng.



1. **Tạo giỏ hàng cho các DigitalVideoDisc**

* Tạo giỏ hàng trong hàm main thử các chức năng.  
  
* Kết quả chạy   
  

1. **Xóa DVD khỏi giỏ hàng**

****

1. **Trả lời các câu hỏi**
2. Khi nào thì nên sử dụng các phương thức truy cập accessor methods

* Các phương thức truy cập (accessor methods), thường được biết đến như là getter và setter, có thể rất hữu ích trong một số tình huống nhất định, nhưng cũng có những điểm yếu mà lập trình viên cần phải xem xét. Dưới đây là những điểm quan trọng để xác định khi nào nên sử dụng chúng:
* Bảo mật và kiểm soát
  + Sử dụng các phương thức truy cập cho phép lập trình viên kiểm soát cách mà thuộc tính của đối tượng được truy cập và thay đổi.
  + Setter có thể bao gồm logic kiểm tra để đảm bảo giá trị được gán là hợp lệ, từ đó tăng cường tính bảo mật của dữ liệu.
* Ẩn thông tin
  + Các phương thức truy cập giúp bạn duy trì nguyên tắc đóng gói (encapsulation) trong lập trình hướng đối tượng. Điều này có nghĩa là bạn có thể ẩn các thuộc tính của lớp và chỉ cho phép truy cập thông qua các phương thức
* Cung cấp tính năng bổ xung
  + Getter và setter có thể được mở rộng để thực hiện các hành động bổ sung khi dữ liệu được truy cập hoặc thay đổi, như ghi log hoặc kích hoạt sự kiện.

1. Nếu bạn tạo một phương thức khởi tạo để tạo một DVD theo tiêu đề, sau đó tạo một phương thức khởi tạo để tạo một DVD theo thể loại. Liệu JAVA có cho phép bạn làm điều này không?

* Có, Java cho phép bạn tạo nhiều phương thức khởi tạo (constructors) với các tham số khác nhau trong cùng một lớp. Đây được gọi là "overloading" (nạp chồng) phương thức.