

# Lập trình hướng đối tượng

Lecturer: Trinh Thanh Trung, trungtt@soict.hust.edu.vn

## Phòng thí nghiệm 02: Mô hình hóa và đóng gói vấn đề

Trong phòng thí nghiệm này, bạn sẽ thực hành với:

- Cài đặt công cụ thiết kế cho sơ đồ UML: Astah
- Lập mô hình sự cố với sơ đồ Trường hợp sử dụng
- Đóng gói và các kỹ thuật đóng gói khác nhau
- Thiết kế lớp học cho các trường hợp sử dụng liên quan đến quản lý giỏ hàng
- Java Implementation: Tạo các class trong Eclipse, constructor, getters và setters, tạo instance của các class

### 0. Nộp bài tập

Đối với lớp học phòng thí nghiệm này, bạn sẽ phải nộp công việc của mình hai lần, cụ thể:

- **Ngay sau giờ học:** đối với thời hạn này, bạn nên bao gồm bất kỳ công việc nào bạn đã làm trong lớp học phòng thí nghiệm.
- **10 giờ tối hai ngày sau lớp học:** đối với thời hạn này, bạn nên bao gồm sơ đồ trường hợp sử dụng cuối cùng, sơ đồ lớp final, **mã nguồn** của tất cả các phần của phòng thí nghiệm này và bài tập đọc trong một nhà thờ di cụt thể là "**Lab02**". Lưu ý rằng đối với sơ đồ trường hợp sử dụng, hãy gửi cả tệp nguồn (.astah) và hình ảnh đã xuất (.png) trong thư mục cụ thể là "**Yêu cầu**"; đối với sơ đồ lớp, cũng gửi cả tệp nguồn (.astah) và tệp hình ảnh đã xuất (.png) trong thư mục cụ thể là "**Thiết kế**"; đối với bài tập đọc, hãy gửi một tệp hình ảnh của bài tập đọc trong thư mục cụ thể là "**ReadingAssignment**", đặt tất cả vào một thư mục cụ thể là "**Lab02**" và đẩy nó vào nhánh **chính** của kho lưu trữ hợp lệ.

Lưu ý rằng tất cả các mã hoặc sơ đồ mẫu trong phòng thí nghiệm chỉ là các ví dụ / đề xuất. Bạn có thể cần phải thay đổi chúng để đáp ứng yêu cầu. Mỗi học sinh phải nộp bài làm của mình và không cho hoặc nhận viện trợ không được phép. Nếu không, chúng tôi sẽ áp dụng các phương pháp cực đoan để đo lường để ngăn chặn gian lận. P cho thuê ghi lại câu trả lời cho tất cả các câu hỏi vào một tệp văn bản có tên "**câu trả lời.txt**" và gửi nó trong kho lưu trữ của bạn.

### 1. UML & Astah

Ngôn ngữ mô hình hóa thống nhất (UML) là một họ các ký hiệu đồ họa, được hỗ trợ bởi siêu hình hóa duy nhất, giúp mô tả và thiết kế các hệ thống phần mềm, đặc biệt là các hệ thống phần mềm được xây dựng theo phong cách hướng đối tượng (Fowler, 2003).

Astah là một công cụ thiết kế hỗ trợ UML. Để nhận Astah UML, hãy truy cập <http://astah.net/student-license-request>, điền vào biểu mẫu và gửi yêu cầu. Sau đó làm theo 3 bước trong trang được chuyển hướng <http://astah.net/student/thank-you>. Xem sơ đồ trường hợp sử dụng với Astah tại <http://astah.net/manual/422-usecase-diagram>.

Cố gắng vẽ một số sơ đồ UML trong Astah UML, chẳng hạn như sơ đồ lớp, sơ đồ hoạt động, sơ đồ trình tự.

## 2. Báo cáo vấn đề của dự án AIMS

*Đưa ra tuyên bố vấn đề dưới đây của Dự án AIMS. Vui lòng đọc kỹ tuyên bố để đảm bảo rằng bạn hiểu rõ phần mềm mà bạn sẽ thiết kế và triển khai từ phòng thí nghiệm này. Nếu bạn có bất kỳ vấn đề, xin vui lòng hỏi giáo sư hoặc TA.*

Có thể có một tương lai mà Tiki và Sendo đang đàm phán về khả năng sáp nhập để cạnh tranh với các nền tảng thương mại điện tử khác và đặc biệt là những người có người ủng hộ nước ngoài. Việc sáp nhập hai công ty này sẽ tạo ra một công ty Ti-do, trong đó "Ti" đến từ Tiki, và "do" là từ Sendo, có nghĩa là một công ty tỷ đô la bằng tiếng Việt. Công ty đó, công ty Ti-do, muốn bạn giúp họ tạo ra một hệ thống hoàn toàn mới cho dự án AIMS (AIMS là viết tắt của An Internet Media Store). Hiện tại, chỉ có một loại phương tiện truyền thông: Digital Video Disc (DVD).

Khách hàng có thể duyệt danh sách các đĩa DVD có sẵn trong cửa hàng, thứ tự hiển thị dựa trên ngày được thêm vào của họ, từ mới nhất đến cũ nhất. Khi khách hàng muốn tìm kiếm DVD để thêm vào giỏ hàng, họ có thể chọn một trong ba tùy chọn tìm kiếm. Phần mềm sẽ hiển thị danh sách tất cả các trận đấu (DVD mới nhất đầu tiên) với tất cả thông tin của họ. Họ hoặc cô ấy cũng có thể chọn phát một đĩa DVD cụ thể. Phần mềm sẽ phát DVD (phần demo). Nếu một đĩa DVD có độ dài từ 0 trở xuống, hệ thống phải thông báo cho khách hàng rằng DVD không thể phát được.

- Khi khách hàng tìm kiếm DVD theo tiêu đề, họ sẽ cung cấp một chuỗi từ khóa. Nếu bất kỳ DVD nào có tiêu đề chứa bất kỳ từ nào trong chuỗi từ khóa, thì DVD đó sẽ được tính là video trùng khớp. Lưu ý rằng việc so sánh các từ ở đây là không phân biệt chữ hoa chữ thường.
- Khi khách hàng tìm kiếm DVD theo danh mục, họ sẽ cung cấp tên danh mục. Nếu bất kỳ DVD nào có danh mục phù hợp (không phân biệt chữ hoa chữ thường), nó sẽ được tính là một kết quả phù hợp.
- Khi khách hàng tìm kiếm DVD theo giá, họ sẽ cung cấp chi phí tối thiểu và tối đa hoặc chỉ là chi phí tối đa.

Customers có thể xem thông tin chi tiết của DVD từ danh sách DVD. Anh ấy / cô ấy có thể thêm DVD vào giỏ hàng từ danh sách DVD hoặc màn hình chi tiết.

Khi khách hàng muốn xem giỏ hàng hiện tại, hệ thống sẽ hiển thị tất cả thông tin của DVD, cùng với tổng chi phí. Khách hàng có thể nghe DVD (một phần demo) trong giỏ hàng trước khi xác nhận đặt hàng. Customers có thể sắp xếp tất cả các đĩa DVD trong giỏ hàng theo tiêu đề hoặc theo chi phí:

- Sắp xếp theo tiêu đề: hệ thống hiển thị tất cả các đĩa DVD trong trình tự bảng chữ cái theo tiêu đề. Trong trường hợp chúng có cùng tiêu đề, các đĩa DVD có chi phí cao hơn sẽ được hiển thị trước.
- Sắp xếp theo chi phí: hệ thống hiển thị tất cả các đĩa DVD theo thứ tự chi phí giảm dần. Trong trường hợp chúng có cùng chi phí, các đĩa DVD sẽ được đặt hàng bằng cách tăng tiêu đề.

Khách hàng có thể cập nhật số lượng DVD trong giỏ hàng hoặc lấy DVD ra khỏi giỏ hàng. Để tăng nhu cầu của người tiêu dùng đối với sản phẩm và tăng doanh số bán hàng, khách hàng được phép có một mặt hàng miễn phí được hệ thống chọn ngẫu nhiên trong giỏ hàng. Khách hàng có thể lọc DVD trong giỏ hàng bằng cách cung cấp ID hoặc tiêu đề. Nếu tìm thấy mặt hàng, hãy hiển thị thông tin của mặt hàng được tìm thấy trong giỏ hàng. Hoặc nếu không, hãy thông báo cho khách hàng rằng mặt hàng không được tìm thấy trong giỏ hàng hiện tại.

Khách hàng có thể yêu cầu đặt hàng khi họ đang nhìn thấy giỏ hàng hiện tại. Để đơn giản hóa, anh ấy / cô ấy không cần phải đăng nhập để đặt hàng. Ứng dụng sẽ nhắc nhở khách hàng nhập thông tin giao hàng và hướng dẫn giao hàng. Sau đó, phần mềm sẽ tính toán phí giao hàng dựa trên tổng khối lượng của đơn đặt hàng và địa điểm giao hàng. Sau đó, nó sẽ hiển thị cho khách hàng hóa đơn bao gồm danh sách DVD, tổng chi phí trước VAT, tổng chi phí sau VAT và phí giao hàng. Sau đó, khách hàng có thể tiến hành thanh toán cho đơn hàng. Hiện tại, chỉ có một phương thức thanh toán - tức là thẻ tín dụng - được phép bằng cách kết

nối với hệ thống liên kết thẻ để kiểm tra tính xác thực của thẻ hoặc thực hiện giao dịch thanh toán. Sau khi giao dịch, phần mềm AIMS sẽ hiển thị đầy đủ các thông tin chi tiết như ID giao dịch, chủ thẻ, số tiền giao dịch, tin nhắn giao dịch, số dư, ngày giao dịch cho khách hàng. Đơn đặt hàng sẽ ở trạng thái chờ xử lý và thông tin của đơn đặt hàng & giao dịch sẽ được gửi đến email của khách hàng.

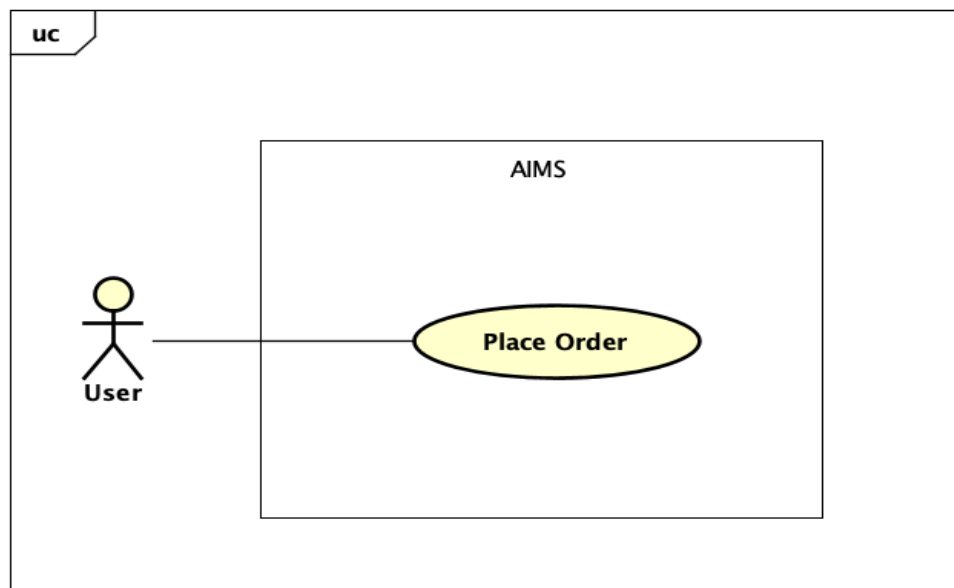
Quản lý cửa hàng cần đăng nhập vào hệ thống để điều hướng đến chế độ quản lý. Anh ấy / cô ấy có thể xem danh sách các đơn đặt hàng đang chờ xử lý, sau đó có thể chọn bất kỳ đơn hàng nào để xem chi tiết của nó để phê duyệt hoặc từ chối đơn đặt hàng. Cửa hàng manager có thể thêm DVD mới vào cửa hàng. Người đó phải cung cấp tất cả thông tin của DVD mới, bao gồm ID, chức danh, danh mục, đạo diễn, thời lượng và chi phí. Ngoài ra, người quản lý cũng có thể xóa DVD khỏi cửa hàng.

### 3. Sơ đồ trường hợp sử dụng

Dựa trên tuyên bố vấn đề trong phần 2, vui lòng vẽ sơ đồ trường hợp sử dụng thô bằng cách sử dụng Astah UML cho dự án AIMS.

Sơ đồ trường hợp sử dụng là một trong những sơ đồ UML để nắm bắt các hành vi động, tức là mô hình thay đổi trạng thái theo thời gian ([https://www.cs.uct.ac.za/mit\\_notes/software/htmls/ch05s08.html](https://www.cs.uct.ac.za/mit_notes/software/htmls/ch05s08.html)). Sơ đồ trường hợp sử dụng minh họa mối quan hệ giữa các trường hợp sử dụng. Mô hình trường hợp sử dụng mô tả các yêu cầu chức năng của phần mềm về các **trường hợp sử dụng**. Sơ đồ trường hợp sử dụng là một mô hình của các chức năng dự định của phần mềm và môi trường của nó và đóng vai trò như một hợp đồng giữa khách hàng và nhà phát triển. Bởi vì nó là một công cụ lập kế hoạch rất mạnh mẽ, sơ đồ trường hợp sử dụng thường được sử dụng trong tất cả các giai đoạn của chu kỳ phát triển. Để hiểu rõ hơn, hãy xem <https://www.uml-diagrams.org/use-case-diagrams.html>.

Hình 1 với một trường hợp sử dụng: đặt hàng. Bạn phải xác định nhiều trường hợp sử dụng hơn để cập nhật sơ đồ trường hợp sử dụng cho lần gửi cuối cùng.



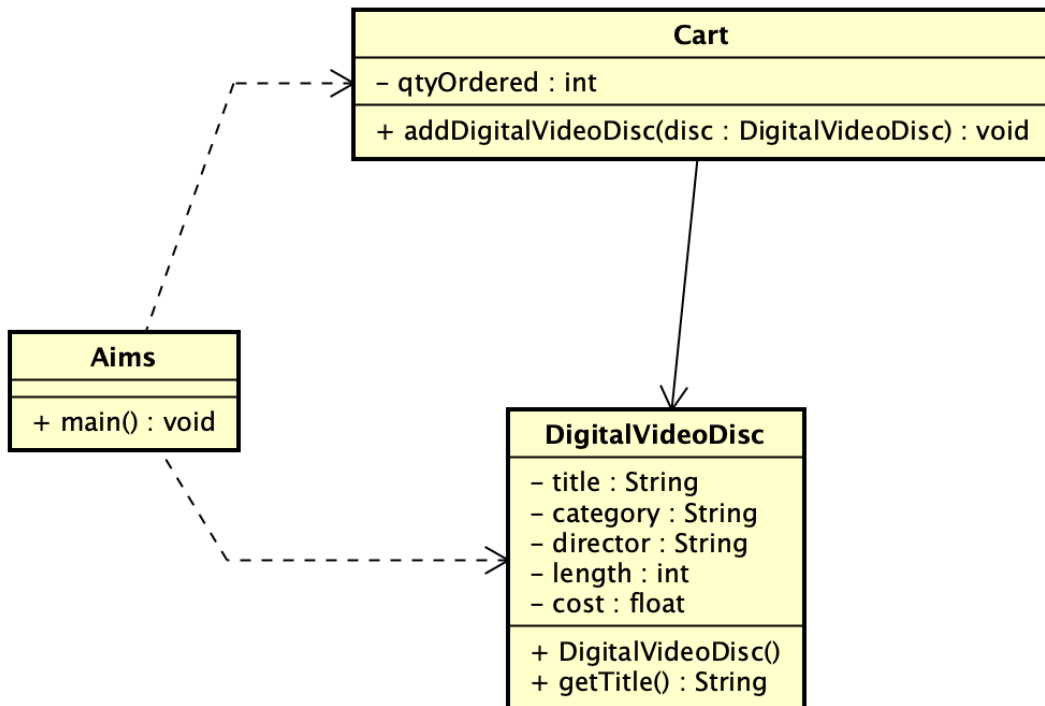
Hình 1. Mẫu sơ đồ trường hợp sử dụng cho Dự án AIMS.

#### 4. Sơ đồ lớp UML cho các trường hợp sử dụng liên quan đến quản lý giỏ hàng

Hệ thống cần tạo một giỏ hàng mới cho người dùng, nơi nó sẽ lưu giữ thông tin trên các đĩa DVD mà người dùng muốn mua. Người dùng có thể thêm, xóa DVD khỏi giỏ hàng cũng như tính toán chi phí. Người dùng có thể thêm tối đa 20 DVD vào một giỏ hàng. Giỏ hàng với thông tin và hành vi của nó được mô hình hóa với lớp **Cart**. Nếu người dùng thêm DVD vào giỏ hàng, hệ thống cũng phải tạo một đĩa DVD mới dựa trên thông tin mà người dùng cung cấp. Thông tin này có thể được hiển thị bất cứ khi nào người dùng quyết định xem nó. DVD với thông tin và chức năng của nó được mô hình hóa với lớp **DigitalVideoDisc**. Cuối cùng, ứng dụng cần một điểm vào để hiển thị và lấy đầu vào từ người dùng (thông qua giao diện dòng lệnh), sẽ là lớp **Aims**. Sơ đồ lớp mẫu được minh họa trong Hình 2, bao gồm 3 lớp:

- Lớp **Aims** cung cấp một phương thức main() tương tác với phần còn lại của hệ thống
- Lớp **DigitalVideoDisc** lưu trữ tiêu đề, danh mục, chi phí, đạo diễn và độ dài
- Lớp **Cart** để duy trì một mảng của các đối tượng DigitalVideoDisc này

**Bạn phải cập nhật sơ đồ lớp học này sau các bài tập dưới đây để gửi bài cuối cùng.**



Hình 2. Sơ đồ lớp mẫu cho các trường hợp sử dụng liên quan đến quản lý giỏ hàng.

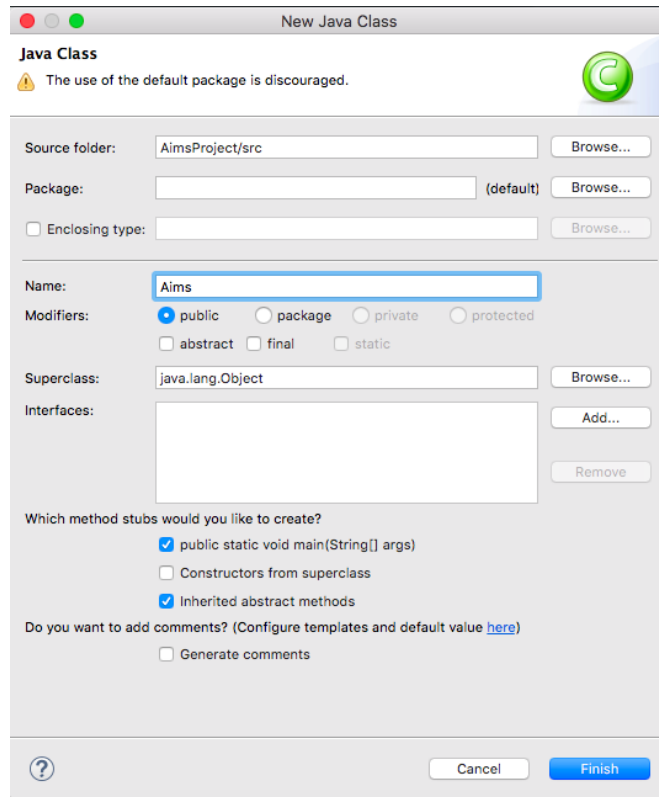
#### 5. Tạo lớp học Aims

- **Mở Eclipse**

- **Tạo một JavaProject mới có tên là "AimsProject"**

- **Tạo lớp Aims:** In thư mục src, tạo một lớp mới có tên là Aims:

+ Nhấp chuột phải vào thư mục và chọn New -> Class:



Hình 3. Tạo lớp Aims bằng Eclipse

+ Bạn có thể cần kiểm tra tùy chọn "public static void main(String[] args)"  
Điều này sẽ tự động tạo ra hàm chính trong lớp Aims.java như kết quả sau.

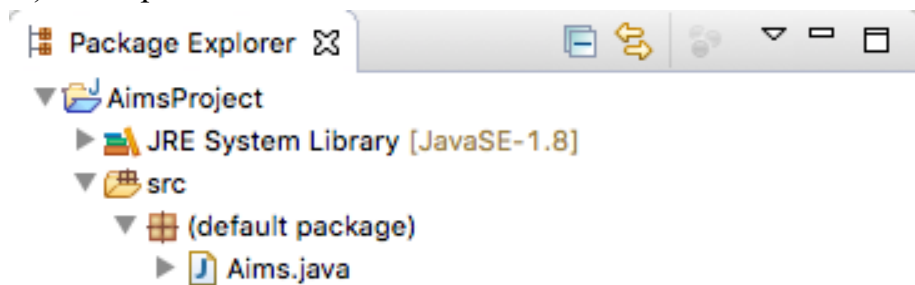
```

1  public class Aims {
2
3
4  public static void main(String[] args) {
5      // TODO Auto-generated method stub
6
7  }
8
9  }
10

```

Hình 4. Mã được tạo cho lớp Aims

+ Vì bạn không chọn bất kỳ gói nào cho lớp Aims nên Eclipse sau đó hiển thị gói biểu tượng và đề cập (gói mặc định) cho lớp của bạn.



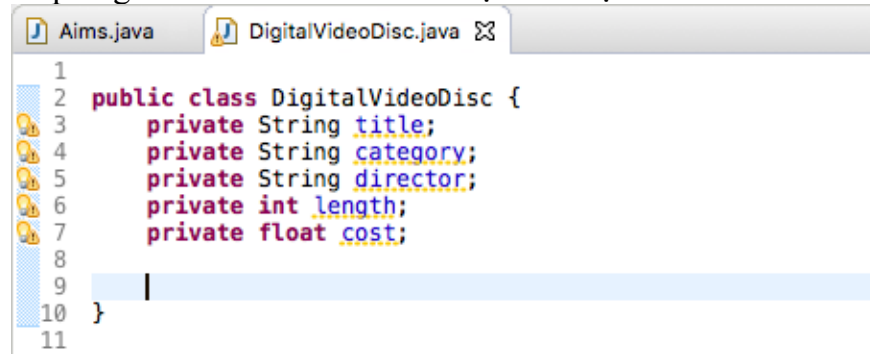
Hình 5. Mục tiêu trong gói mặc định

+ Bạn có thể tạo một gói và di chuyển lớp sang gói này nếu muốn. Trong thư mục src, một thư mục con sẽ được tạo (với tên của gói) để lưu trữ lớp. Tự làm và mở thư mục src để xem kết quả.

## 6. Tạo lớp DigitalVideoDisc và các thuộc tính của nó

Đảm bảo rằng tùy chọn cho phương pháp chính không được chọn.

Mở mã nguồn của lớp DigitalVideoDisc và thêm một số thuộc tính bên dưới:



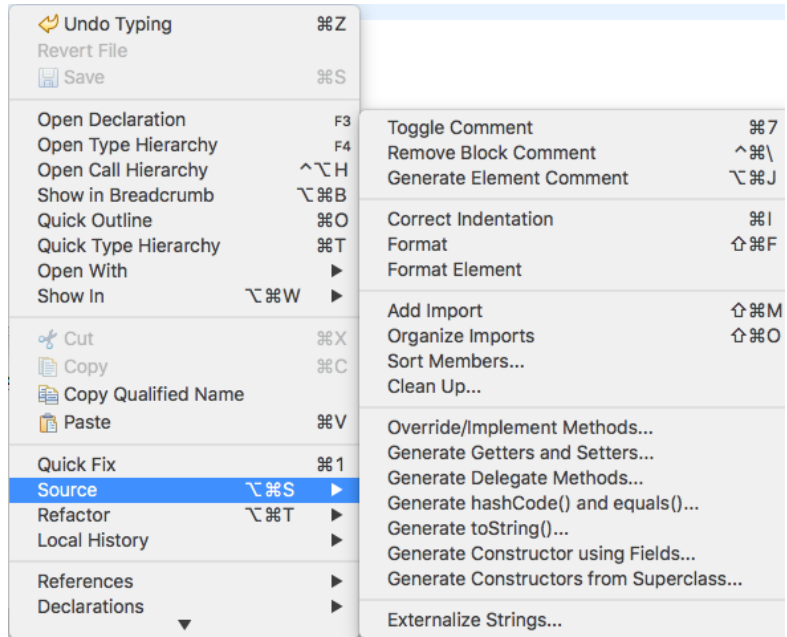
```
1
2 public class DigitalVideoDisc {
3     private String title;
4     private String category;
5     private String director;
6     private int length;
7     private float cost;
8
9
10 }
11
```

Hình 6. Lớp DigitalVideoDisc

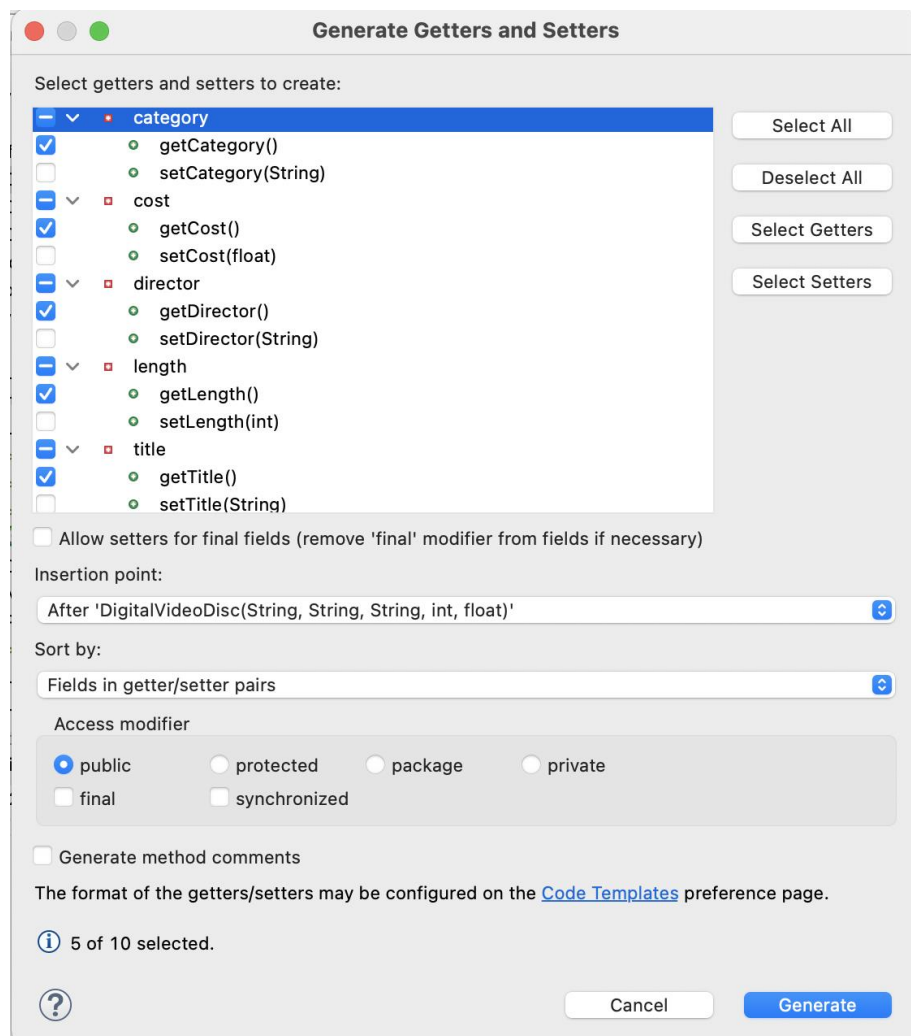
## 7. Tạo phụ kiện và đột biến cho lớp DigitalVideoDisc

Để tạo setters và getters cho các thuộc tính riêng tư, bạn có thể tạo các phương thức để cho phép truy cập công khai có kiểm soát vào từng biến riêng tư này. Eclipse cho phép bạn làm điều này một cách tự động. Tuy nhiên, trong nhiều trường hợp, bạn không cho phép tạo phụ kiện và đột biến cho tất cả các thuộc tính, mà tùy thuộc vào doanh nghiệp. Ví dụ: trong tài khoản ngân hàng, số dư không thể được sửa đổi trực tiếp thông qua bộ đột biến, nhưng nên được tăng hoặc giảm thông qua các trường hợp sử dụng tín dụng hoặc ghi nợ.

- Nhấp chuột phải vào bất kỳ đâu trong tệp nguồn của DigitalVideoDisc.
  - Chọn Nguồn, sau đó chọn Tạo Getters và Setters (Hình 7)
  - Chọn các thuộc tính cần người nhận/người thiết lập
    - Đối với mỗi người trong số họ, hãy chọn mũi tên thả xuống bên cạnh hộp kiểm và chọn chỉ tạo setter, getter hoặc cả hai.
- Gợi ý:** Để chọn các getters / setters thích hợp, người ta nên kiểm tra cẩn thận các yêu cầu của hệ thống. Trong trường hợp của Aims, dựa trên mô tả của hệ thống, chúng ta có thể quyết định các phương thức accessor thích hợp cho từng thuộc tính của lớp DVD như sau: Thứ nhất, không có trường hợp sử dụng nào yêu cầu thay đổi các thuộc tính của DVD sau khi nó được thêm vào, vì vậy chúng tôi loại bỏ tất cả các **setters**. Thứ hai, vì hệ thống cần hiển thị tất cả thông tin của DVD khi người dùng nhìn thấy giỏ hàng hiện tại, nên tất cả các **getters** đều được chọn. (Hình 8)
- Chọn tùy chọn "public" trong công cụ sửa đổi Access
  - Nhấp vào Tạo



Hình 7. Tạo getters & setters bằng Eclipse



Hình 8. Chọn phụ kiện phù hợp

```

public String getTitle() {
    return title;
}
public String getCategory() {
    return category;
}
public String getDirector() {
    return director;
}
public int getLength() {
    return length;
}
public float getCost() {
    return cost;
}

```

Hình 9. Phụ kiện được tạo

### **Đọc bài tập:** Khi nào nên sử dụng các phương pháp phụ kiện?

Đọc bài viết sau và tìm câu trả lời tốt nhất có thể cho câu hỏi trên: Holub, Allen. "Why getter and setter methods is evil" *JavaWorld*, ngày 5 tháng 9 năm 2003,

<https://www.infoworld.com/article/2073723/why-getter-and-setter-methods-are-evil.html>

Bạn cũng nên mở rộng nghiên cứu của mình sang các nguồn khác. Đối với câu trả lời, hãy tóm tắt những phát hiện của bạn dưới dạng sơ đồ tư duy. Bạn có thể vẽ sơ đồ tư duy này bằng tay và chụp ảnh công việc của mình hoặc sử dụng bất kỳ công cụ trực tuyến nào. Trong cả hai trường hợp, định dạng được chấp nhận cho tệp hình ảnh là một trong những định dạng sau: .png, .jpg, .jpeg và .pdf.

## **8. Tạo phương thức Constructor**

Mặc định thì tất cả các class của Java sẽ herit từ **java.lang.Object**. Nếu một lớp không có phương thức constructor, thì trên thực tế, lớp này sử dụng phương thức constructor của **java.lang.Object**. Do đó, bạn luôn có thể tạo một đối tượng của lớp bằng phương thức hàm tạo không đối số. Chẳng hạn:

**DigitalVideoDisc dvd1 = mới DigitalVideoDisc();**

Trong phần này, bạn sẽ tạo cho mình phương thức xây dựng cho DigitalVideoDisc cho các mục đích khác nhau:

- Tạo một đối tượng DVD theo tiêu đề
- Tạo một đối tượng DVD theo danh mục, tiêu đề và chi phí
- Tạo một đối tượng DVD theo đạo diễn, thể loại, tiêu đề và chi phí



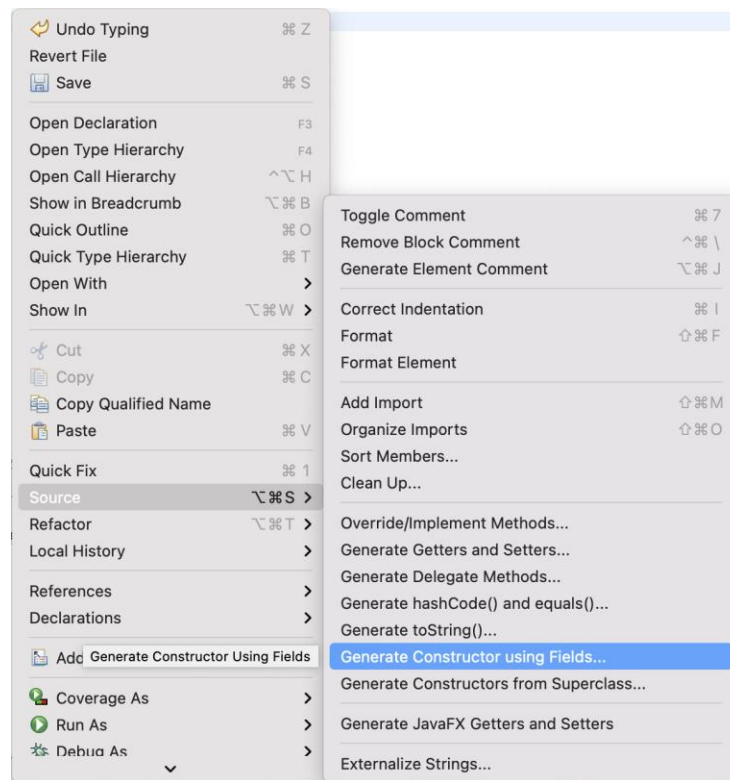
- Tạo một đối tượng DVD theo tất cả các thuộc tính: tiêu đề, thể loại, đạo diễn, độ dài và chi phí

Mỗi mục đích sẽ tương ứng với một phương thức xây dựng. Bằng cách đó, bạn đã thực hành với phương pháp quá tải.

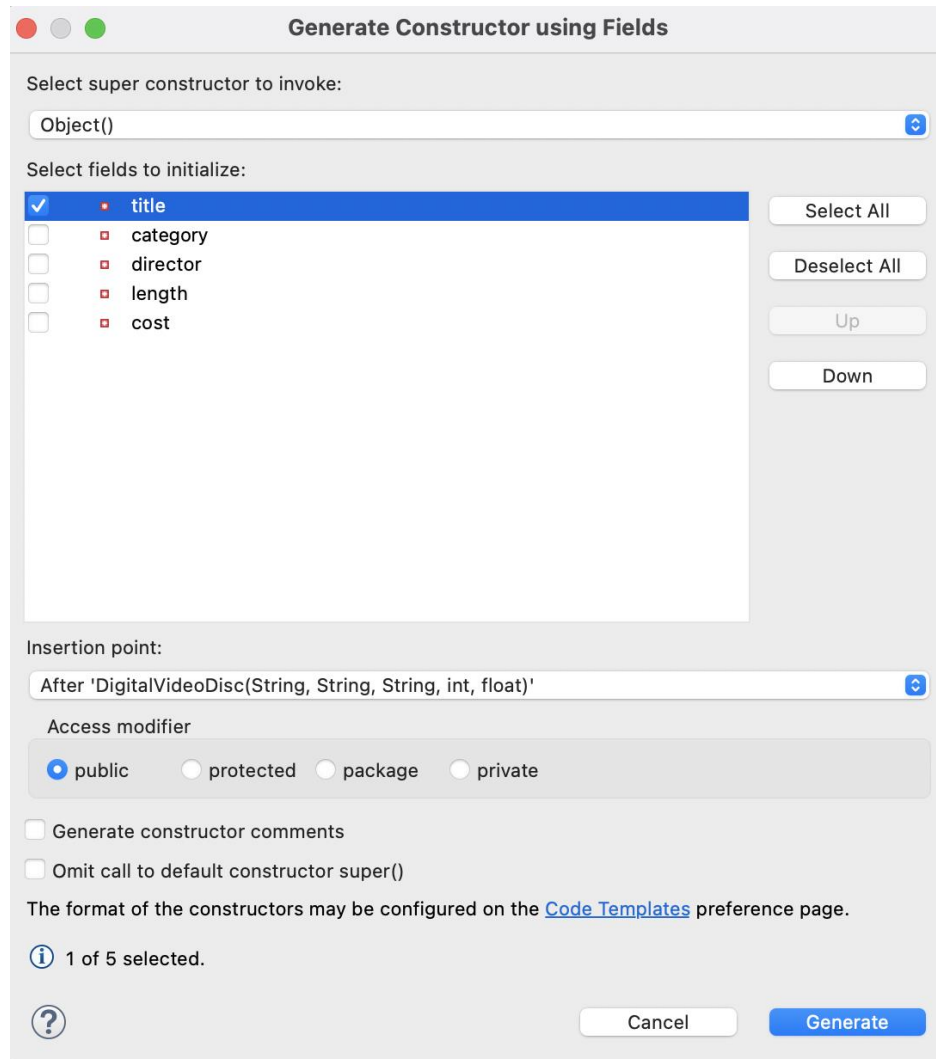
### Câu hỏi:

- Nếu bạn tạo một phương thức xây dựng để xây dựng một đĩa DVD theo tiêu đề sau đó tạo một phương thức xây dựng để xây dựng **một đĩa DVD** theo thể loại. JAVA có cho phép bạn làm điều này không?

Eclipse cũng cho phép bạn tự động tạo các phương thức xây dựng theo trường. Chỉ cần làm tương tự như tạo getters và setters. Click chuột phải vào bất cứ nơi nào trong source file, chọn Source, chọn Generate constructor by fields() Hình 10 sau đó chọn các fields(Hình 11) để generate constructor methods.



Hình 10. Tạo năng lượng xây dựng bằng cách sử dụng các trường



Hình 11. Thiết lập để tạo hàm tạo bằng các trường

Kết quả là:

```
public DigitalVideoDisc(String title) {  
    super();  
    this.title = title;  
}
```

Hình 12. Mã mẫu cho hàm tạo được tạo

Đây là cách bạn tạo phương thức constructor đầu tiên. Tương tự, bạn sẽ tạo ra bởi mình những người khác.

## 9. Tạo lớp Cart để làm việc với DigitalVideoDisc

Lớp **Cart** sẽ chứa một danh sách các đối tượng **DigitalVideoDisc** và có các phương thức có khả năng sửa đổi danh sách.

```
public class Cart {  
  
    public static final int MAX_NUMBERS_ORDERED = 20;  
    private DigitalVideoDisc itemsOrdered[] =  
        new DigitalVideoDisc[MAX_NUMBERS_ORDERED];  
  
}
```

Hình 13. Mã mẫu của lớp Cart

Thêm một trường dưới dạng mảng để lưu trữ danh sách **DigitalVideoDisc**.

Để theo dõi có bao nhiêu **DigitalVideoDiscs** trong giỏ hàng, bạn phải tạo một trường có tên **qtyOrdered** trong lớp **Cart** lưu trữ thông tin này.

Tạo phương thức **addDigitalVideoDisc(đĩa DigitalVideoDisc)** để thêm một mục vào danh sách. Bạn nên kiểm tra số lượng hiện tại để đảm bảo rằng giỏ hàng chưa đầy

- Tạo phương thức **removeDigitalVideoDisc (đĩa DigitalVideoDisc)** để xóa mục được truyền bởi đối số khỏi danh sách.

Tạo phương thức **totalCost()** lặp qua các giá trị của mảng và tính tổng chi phí của các **DigitalVideoDiscs** riêng lẻ. Phương thức này trả về tổng chi phí của giỏ hàng hiện tại.

Lưu ý rằng các phương thức của bạn nên tương tác với người dùng. Ví dụ: sau khi thêm nó nên thông báo cho người dùng: "Đĩa đã được thêm vào" hoặc "Giỏ hàng gần đầy" nếu giỏ hàng đã đầy.

Bây giờ bạn có tất cả các lớp cho ứng dụng. Chỉ cần thực hành với họ trong phần tiếp theo.

## 10. Tạo giỏ hàng của DigitalVideoDiscs

Lớp **Aims** nên tạo một Giỏ hàng mới, sau đó tạo các DVD mới và điền vào giỏ hàng các DVD đó.

Điều này sẽ được thực hiện trong phương thức **main()** của lớp **Aims**.

Thực hiện đoạn mã sau trong phương thức chính của bạn và chạy chương trình để kiểm tra.

```

public static void main(String[] args) {

    //Create a new cart
    Cart anOrder = new Cart();

    //Create new dvd objects and add them to the cart
    DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
        "Animation", "Roger Allers", 87, 19.95f);
    anOrder.addDigitalVideoDisc(dvd1);

    DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
        "Science Fiction", "George Lucas", 87, 24.95f);
    anOrder.addDigitalVideoDisc(dvd2);

    DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
        "Animation", 18.99f);
    anOrder.addDigitalVideoDisc(dvd3);

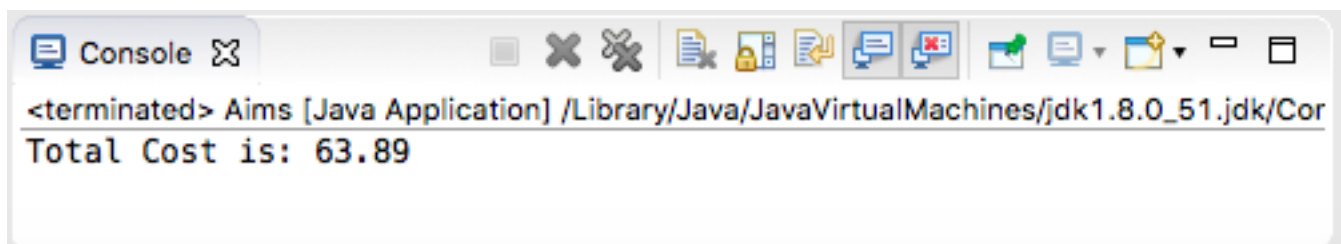
    //print total cost of the items in the cart
    System.out.println("Total Cost is: ");
    System.out.println(anOrder.totalCost());

}

```

Hình 14. Mã mẫu của lớp Aims

Kết quả sẽ là:



Hình 15. Kết quả để tạo một giỏ đĩa video kỹ thuật số.

## 11.Xóa các mặt hàng khỏi giỏ hàng

Bạn phải viết mã trong phương thức chính của mình để kiểm tra phương thức **removeDigitalVideoDisc** (đĩa **DigitalVideoDisc**) của lớp **Cart** và kiểm tra xem mã có được chạy thành công hay không.

## 12. Tham khảo

James Rumbaugh, Ivar Jacobson và Grady Booch (2004). *Hướng dẫn tham khảo ngôn ngữ mô hình hóa hợp nhất, The (Phiên bản thứ 2)*. Giáo dục đại học Pearson