**BÁO CÁO THỰC HÀNH LAP 02  
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

Mục lục

[1. Assignment Submission 1](#_Toc119003661)

[2. UML & Astah 2](#_Toc119003662)

[3. Problem Statement of AIMS Project 2](#_Toc119003663)

[4. Use case diagram 3](#_Toc119003664)

[5. UML Class Diagram for use cases related to cart management 4](#_Toc119003665)

[6. Create Aims class 5](#_Toc119003666)

[7. Create the DigitalVideoDisc class and its attributes 6](#_Toc119003667)

[8. Create accessors and mutators for the class DigitalVideoDisc 7](#_Toc119003668)

[9. Create Constructor method 7](#_Toc119003669)

[10. Create the Cart class to work with DigitalVideoDisc 8](#_Toc119003670)

[11. Create Carts of DigitalVideoDiscs 9](#_Toc119003671)

[12. Removing items from the cart 11](#_Toc119003672)

# **Assignment Submission**

For this lab class, you will have to turn in your work twice, specifically:

* **Right after the class**: for this deadline, you should include any work you have done within the lab class.
* **10PM two days after the class**: for this deadline, you should include the final use case diagram, the final class diagram, the **source code** of all sections of this lab, and the reading assignment in a directory namely “**Lab02**”. Note that for the use case diagram, submit both source file (.astah) and its exported image (.png) in the folder namely “**Requirement**”; for the class diagram, also submit both source file (.astah) and its exported image file (.png) in the folder namely “**Design**”; for the reading assignment, submit an image file of the reading assignment in the folder namely “**ReadingAssignment**”, put all into a directory namely “**Lab02**” and push it to your **master** branch of the valid repository.

Note that all sample codes or diagrams in the lab are only the examples/suggestions. You may need to change them to satisfy the requirement. Each student is expected to turn in his or her own work and not give or receive unpermitted aid. Otherwise, we would apply extreme methods for measurement to prevent cheating. Please write down answers for all questions into a text file named “**answers.txt”** and submit it within your repository.

# **UML & Astah**

The Unified Modeling Language (UML) is a family of graphical notations, backed by single metamodel, that help in describing and designing software systems, particularly software systems built using the object-oriented style (Fowler, 2003).

Astah is a design tool which supports UML. To get Astah UML, go to <http://astah.net/student-license-request>, fill the form, and send the request. Then follow the 3 steps in the redirected page <http://astah.net/student/thank-you>. See use case diagram with Astah at <http://astah.net/manual/422-usecase-diagram>.

Try to draw some several UML diagrams in the Astah UML, such as class diagram, activity diagram, sequence diagram.

# **Problem Statement of AIMS Project**

*Given the below problem statement of AIMS Project. Please read carefully the statement to ensure that you understand clearly the software that you will design and implement from this lab. If you have any problems, please do not hesitate to ask the professor or TA.*

There might be a future that Tiki and Sendo be in talks over a potential merger to contend other e-commerce platforms and especially those who have foreign backers. The merger of these two firms would create a Ti-do company, where “Ti” is from Tiki, and “do” is from Sendo, which means a billion-dollar company in Vietnamese. That firm, Ti-do company, would like you to help them create a brand-new system for AIMS project (AIMS stands for An Internet Media Store). Currently, there is only one type of media: Digital Video Disc (DVD).

Customers can browse the list of DVDs available in the store, the display order is based on their added date, from latest to oldest. When a customer wants to search for DVDs to add to cart, he or she can choose one of three searching options. The software will display a list of all matches (latest DVDs first) with all their information. He or she can also choose to play a specific DVD. The software will play a DVD (a demo part). If a DVD has the length 0 or less, the system must notify the customer that the DVD cannot be played.

* + When a customer searches for DVDs by title, he or she provides a string of keywords. If any DVD has the title containing any word in the string of keywords, it is counted as a match. Note that the comparison of words here is case-insensitive.
  + When a customer searches for DVDs by category, he or she provides the category name. If any DVD has the matching category (case-insensitive), it is counted as a match.
  + When a customer searches for DVDs by price, he or she provides either the minimum and maximum cost, or just the maximum cost.

Customers can view the detail information of a DVD from the list of DVDs. He/she can add a DVD to a cart from a list of DVDs or the detail screen.

When a customer wants to see the current cart, the system displays all the information of the DVDs, along with the total cost. Customers may listen to a DVD (a demo part) in the cart before confirming to place an order. Customers can sort all DVDs in the cart by title or by cost:

* Sort by title: the system displays all the DVDs in the alphabet sequence by title. In case they have the same title, the DVDs having the higher cost will be displayed first.
* Sort by cost: the system the system displays all the DVDs in decreasing cost order. In case they have the same cost, the DVDs will be ordered by increasing title.

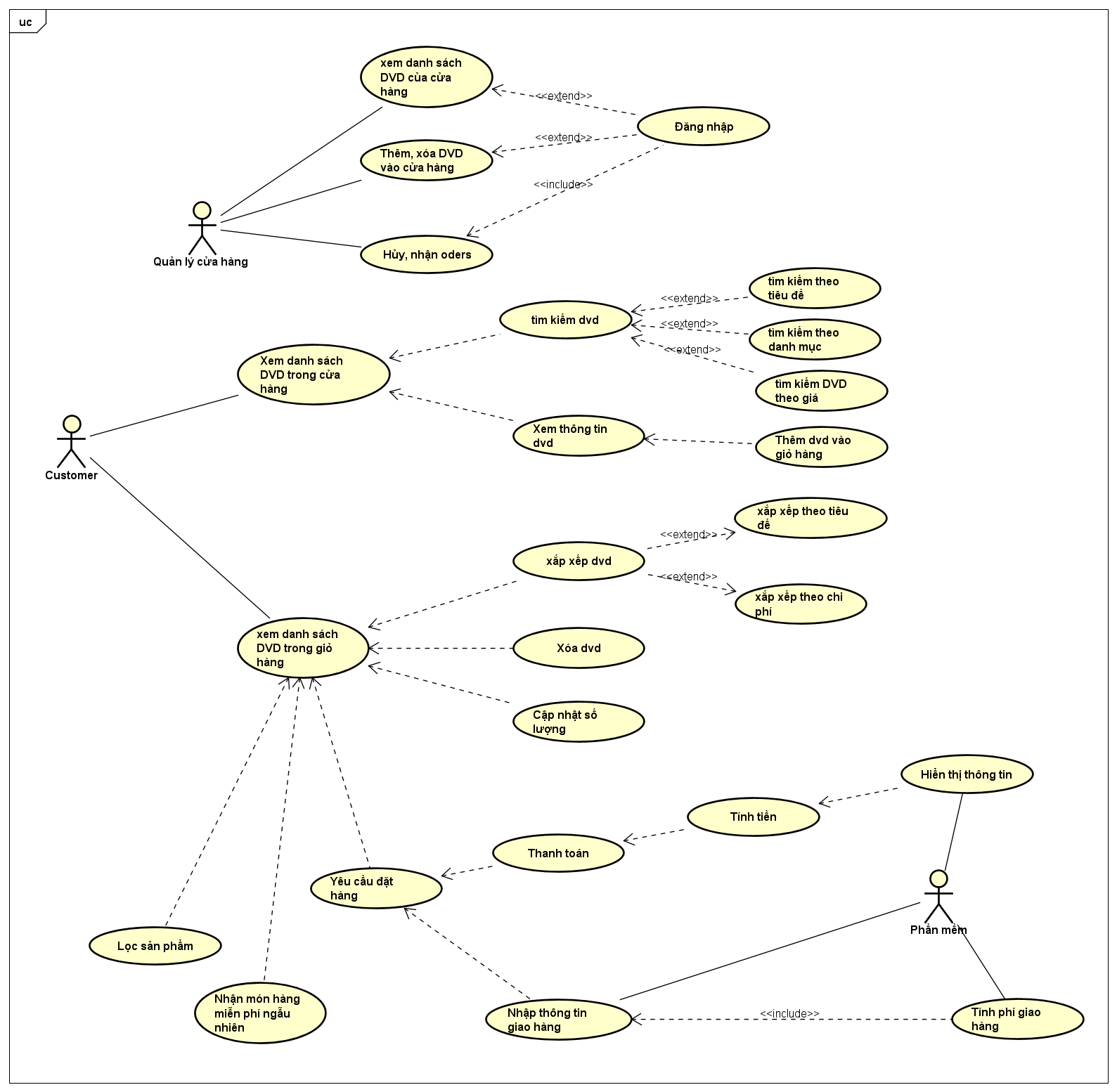
Customers can update the quantity of a DVD in a cart or remove a DVD from a cart. To increase consumer demand for the product and grow sales, customers are allowed to have an item for free which is randomly picked out in the cart by the system. Customers can filter DVDs in the cart by providing either its ID or title. If the item is found, display information of the found item in the cart. Or else, notify the customer the item is not found in the current cart.

The customer can request to place order when they are seeing the current cart. For simplification, he/she does not need to log in to place an order. The application will prompt the customer to enter the delivery information and delivery instructions. The software will then calculate the delivery fee based on the total mass of the order & the delivery location. Then, it will display to the customer the invoice including the DVD list, total cost before VAT, total cost after VAT, and the delivery fee. The customer can then proceed to pay for the order. Currently, only one payment method – i.e. credit card – is allowed by connecting to a card association system for checking the validation of the card or performing the pay transaction. After the transaction, the AIMS software will display all the detailed information such as transaction ID, card owner, transaction amount, transaction message, balance, transaction date to the customer. The order will be in pending state and the information of the order & the transaction will be sent to the customer’s email.

The store manager needs to log in to the system to navigate to the management mode. He/she can see the list of pending orders, then can pick any order to see its detail to approve or reject the order. The store manager can add new DVDs to the store. He or she must provide all information of the new DVD, including its ID, title, category, director, length, and the cost. Additionally, the manager can also remove DVDs from the store.

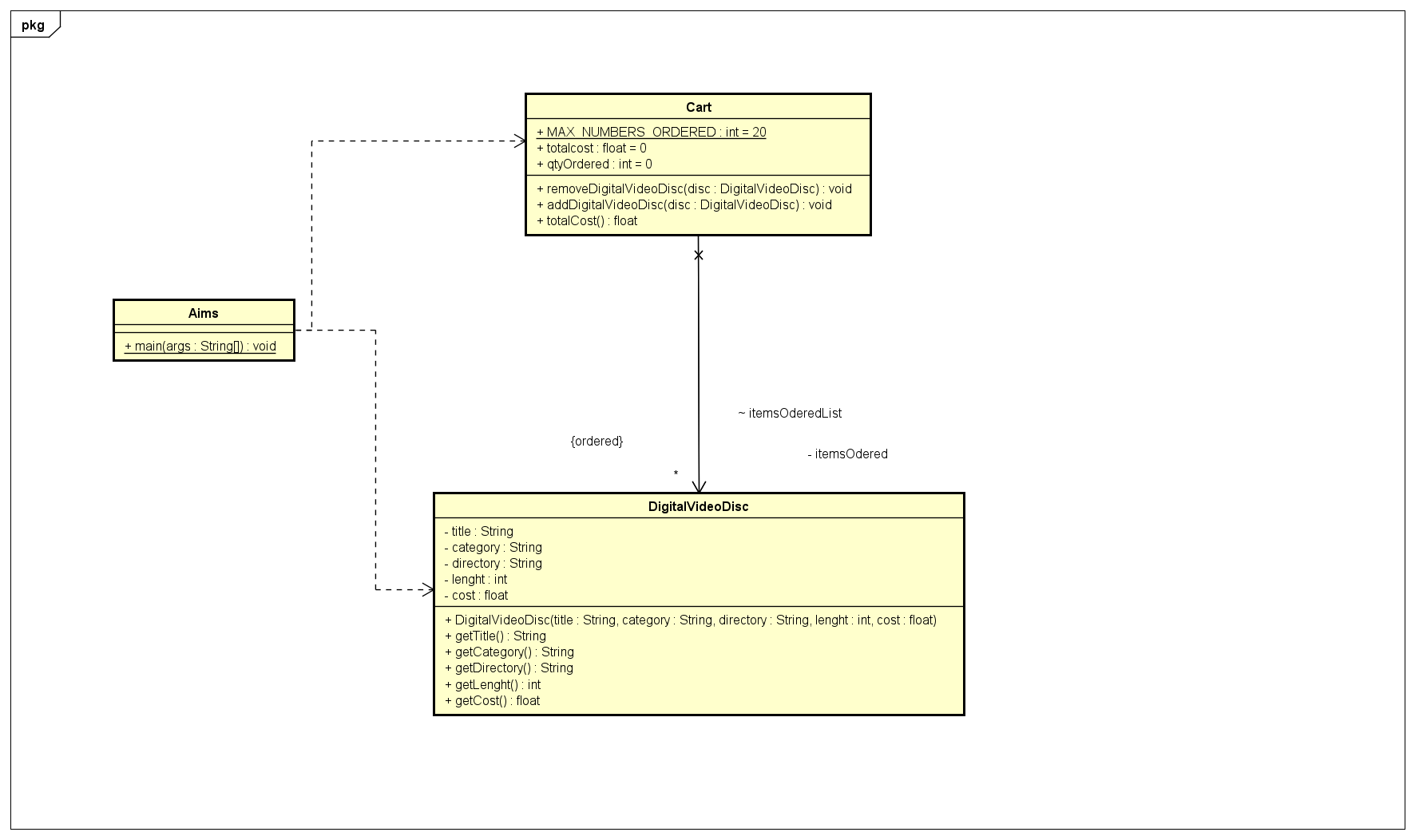
# **Use case diagram**

*Based on the problem statement in the section 2, please draw the use case diagram using Astah UML for the AIMS project.*



Hình :Sơ đồ usecase dùng trong dự án AIMS

# **UML Class Diagram for use cases related to cart management**



Hình 2: Sơ đồ class dùng trong dự án Aims

# **Create Aims class**

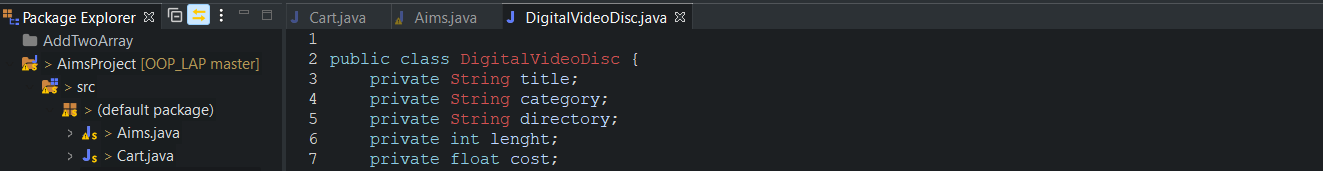
Ảnh có chứa văn bản

Mô tả được tạo tự động

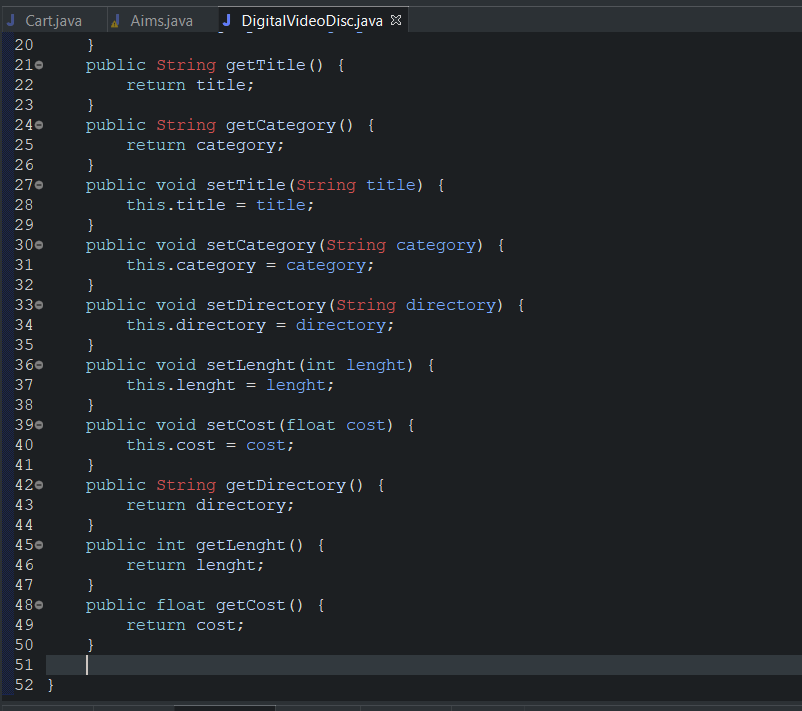
Ảnh có chứa văn bản

Mô tả được tạo tự động

# **Create the DigitalVideoDisc class and its attributes**



# **Create accessors and mutators for the class DigitalVideoDisc**



# **Create Constructor method**

By default, all classes of Java will herit from the **java.lang.Object**. If a class has no constructor method, this class in fact uses the constructor method of **java.lang.Object**. Therefore, you can always create an instance of class by a no-arguments constructor method.For example:

**DigitalVideoDisc dvd1 = new DigitalVideoDisc();**

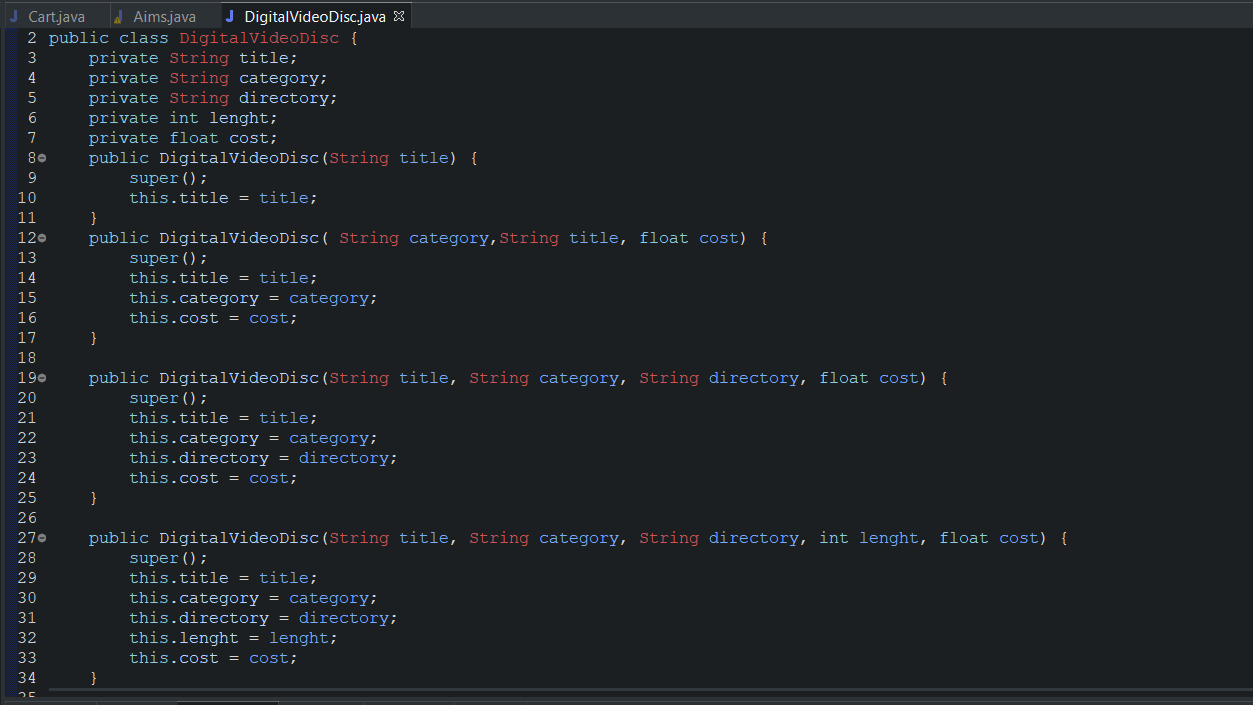
In this part, you will create yourself constructor method for DigitalVideoDisc for different purposes:

- Create a DVD object by title

- Create a DVD object by category, title and cost

- Create a DVD object by director, category, title and cost

- Create a DVD object by all attributes: title, category, director, length and cost

**

# **Create the Cart class to work with DigitalVideoDisc**

Add a field as an array to store a list of **DigitalVideoDisc**.

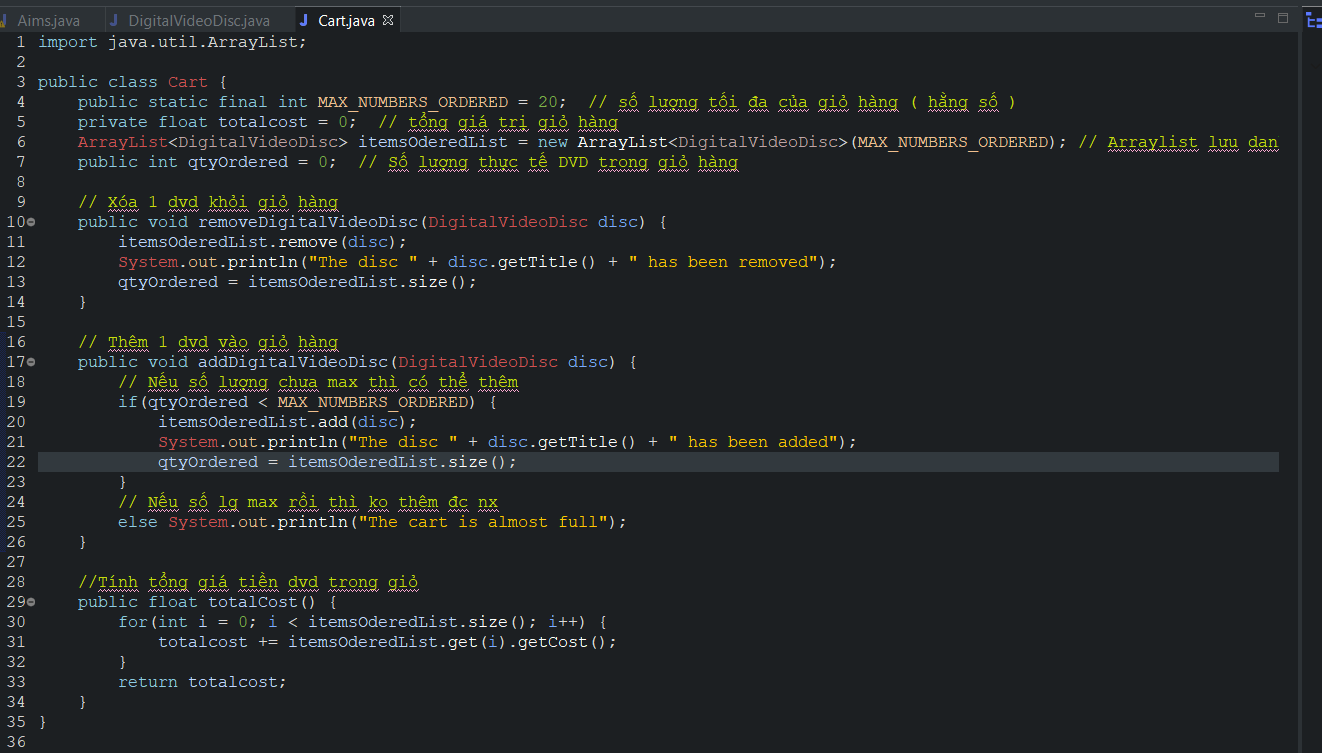
To keep track of how many DigitalVideoDiscs are in the cart, you must create a field named **qtyOrdered** in the Cart class which stores this information.

Create the method **addDigitalVideoDisc(DigitalVideoDisc disc)** to add an item to the list. You should check the current quantity to assure that the cart is not already full

* Create the method **removeDigitalVideoDisc(DigitalVideoDisc disc)** to remove the item passed by argument from the list.

Create the **totalCost()** method which loops through the values of the array and sums the costs of the individual **DigitalVideoDiscs**. This method returns the total cost of the current cart.

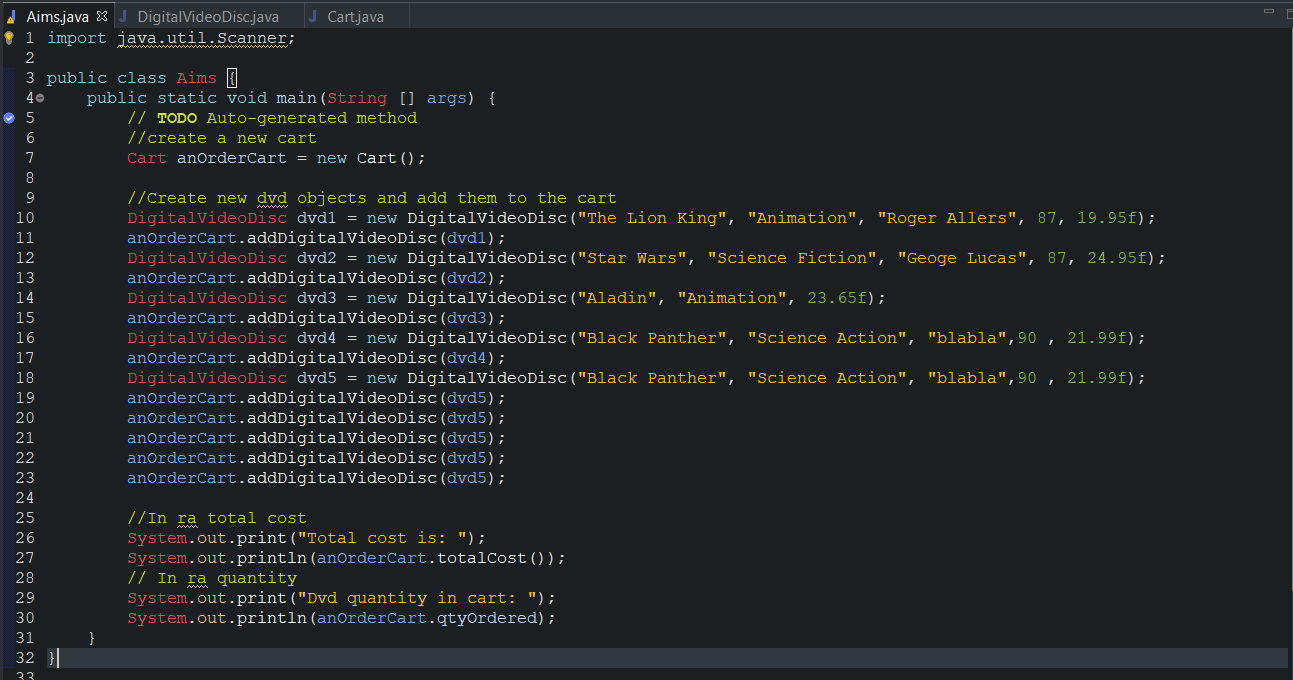
Note that your methods should interact with users. For example: after adding it should inform the user: "**The disc has been added**" or "**The cart is almost full**" if the cart is full.



# **Create Carts of DigitalVideoDiscs**

The **Aims** class should create a new Cart, and then create new DVDs and populate the cart with those DVDs. This will be done in the **main()** method of the Aims class.

Do the following code in your main method and run the program to test.



Ảnh có chứa văn bản

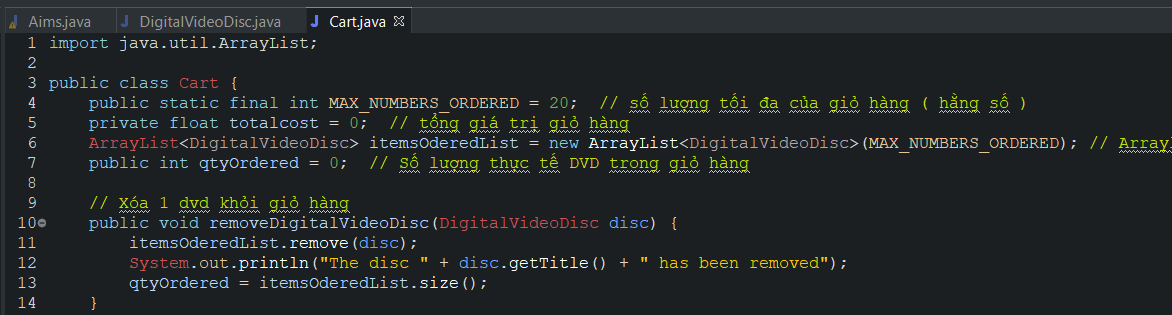
Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

# Removing items from the cart

Ảnh có chứa văn bản

Mô tả được tạo tự độngYou have to write code in your main method to test the **removeDigitalVideoDisc(DigitalVideoDisc disc)** method of the Cart class and check if the code is successfully run.