BÁO CÁO THỰC HÀNH LAP 2   
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục

[1. Problem Statement of AIMS Project 2](#_Toc119010216)

[2. Use case diagram 3](#_Toc119010217)

[3. UML Class Diagram for use cases related to cart management 4](#_Toc119010218)

[4. Create Aims class 5](#_Toc119010219)

[5. Create the DigitalVideoDisc class and its attributes 5](#_Toc119010220)

[6. Create accessors and mutators for the class DigitalVideoDisc 6](#_Toc119010221)

[7. Create Constructor method 7](#_Toc119010222)

[8. Create the Cart class to work with DigitalVideoDisc 8](#_Toc119010223)

[9. Create Carts of DigitalVideoDiscs 9](#_Toc119010224)

[10. Removing items from the cart 9](#_Toc119010225)

[11. Result 10](#_Toc119010226)

[a) totalCost 10](#_Toc119010227)

[b) listCart 11](#_Toc119010228)

[c) addDigitalVideoDisc 12](#_Toc119010229)

[The cart isn’t full 12](#_Toc119010230)

[The cart is full 13](#_Toc119010231)

[d) removeDigitalVideoDisc 14](#_Toc119010232)

[12. Reading Assignment: 16](#_Toc119010233)

[13. Question 17](#_Toc119010234)

# Problem Statement of AIMS Project

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:

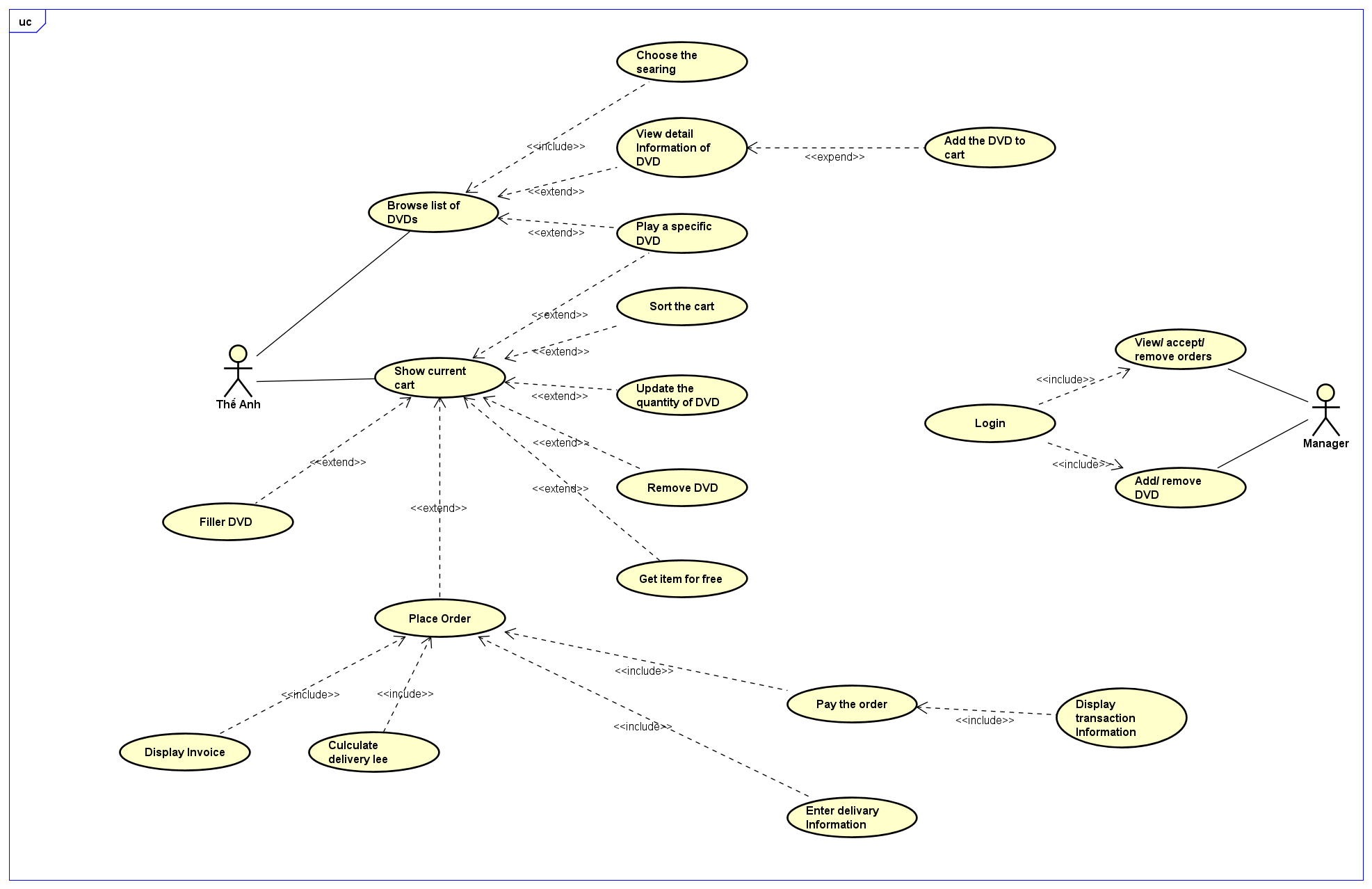
* Sort by title: the system displays all the DVDs in the alphabet sequence by title. In case they have the same title, the DVDs having the higher cost will be displayed first.
* Sort by cost: the system the system displays all the DVDs in decreasing cost order. In case they have the same cost, the DVDs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

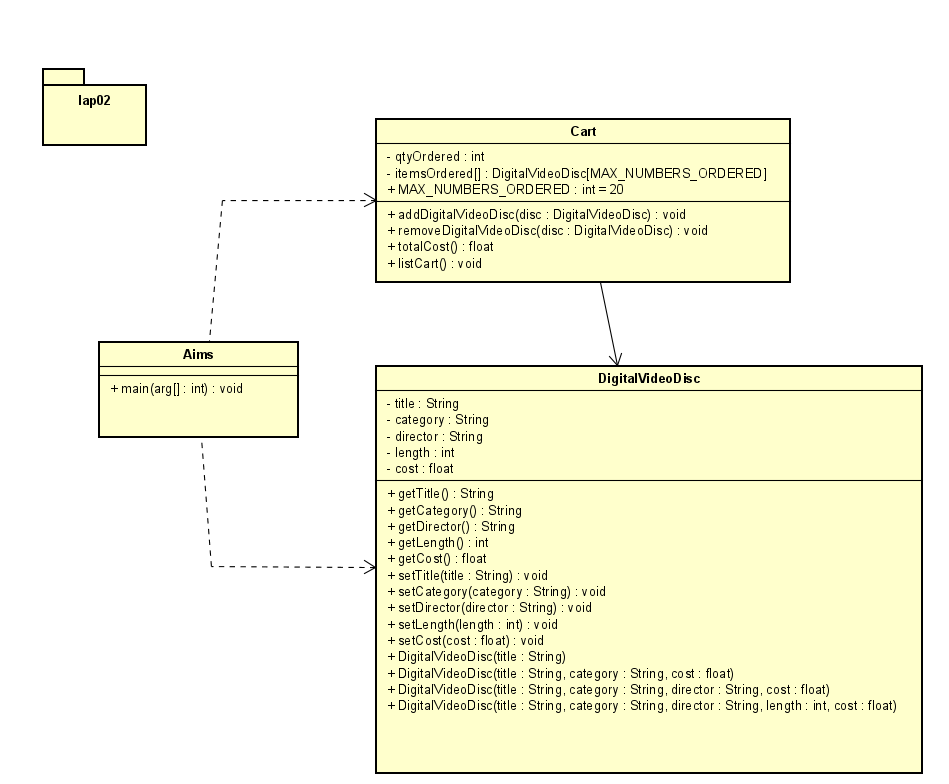
The customer can request to place order when they are seeing the current cart. For simplification, he/she does not need to log in to place an order. The application will prompt the customer to enter the delivery information and delivery instructions. The software will then calculate the delivery fee 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



# UML Class Diagram for use cases related to cart management



# Create Aims class

Ảnh có chứa văn bản

Mô tả được tạo tự động

# Create the DigitalVideoDisc class and its attributes

Ảnh có chứa văn bản

Mô tả được tạo tự động

# Create accessors and mutators for the class DigitalVideoDisc

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

Mô tả được tạo tự động

# Create Constructor method

Ảnh có chứa văn bản

Mô tả được tạo tự động

# Create the Cart class to work with DigitalVideoDisc

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

# Create Carts of DigitalVideoDiscs

**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 động**

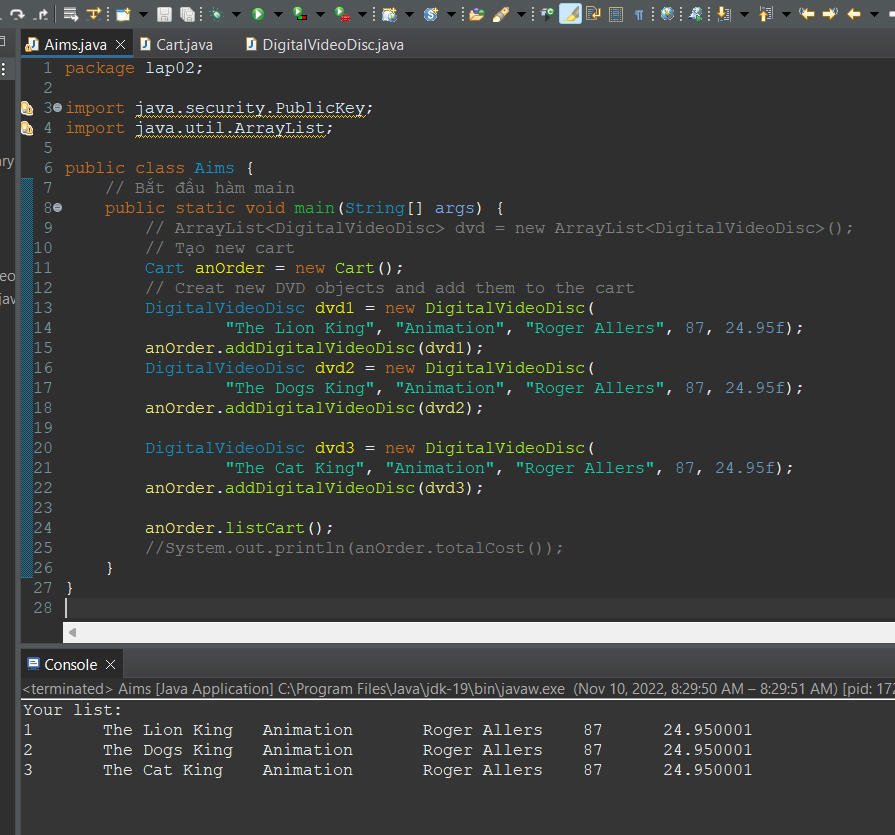
# Result

## totalCost

Ảnh có chứa văn bản

Mô tả được tạo tự động

## listCart



## addDigitalVideoDisc

Ảnh có chứa văn bản

Mô tả được tạo tự động

### The cart isn’t full

Ảnh có chứa bàn

Mô tả được tạo tự động

### The cart is full

Ảnh có chứa văn bản

Mô tả được tạo tự động

## removeDigitalVideoDisc

* Code

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Result

Trước khi xóaẢnh có chứa văn bản

Mô tả được tạo tự động

Sau khi xóa Ảnh có chứa văn bản

Mô tả được tạo tự động

# Reading Assignment:

**When should accessor methods be used?**

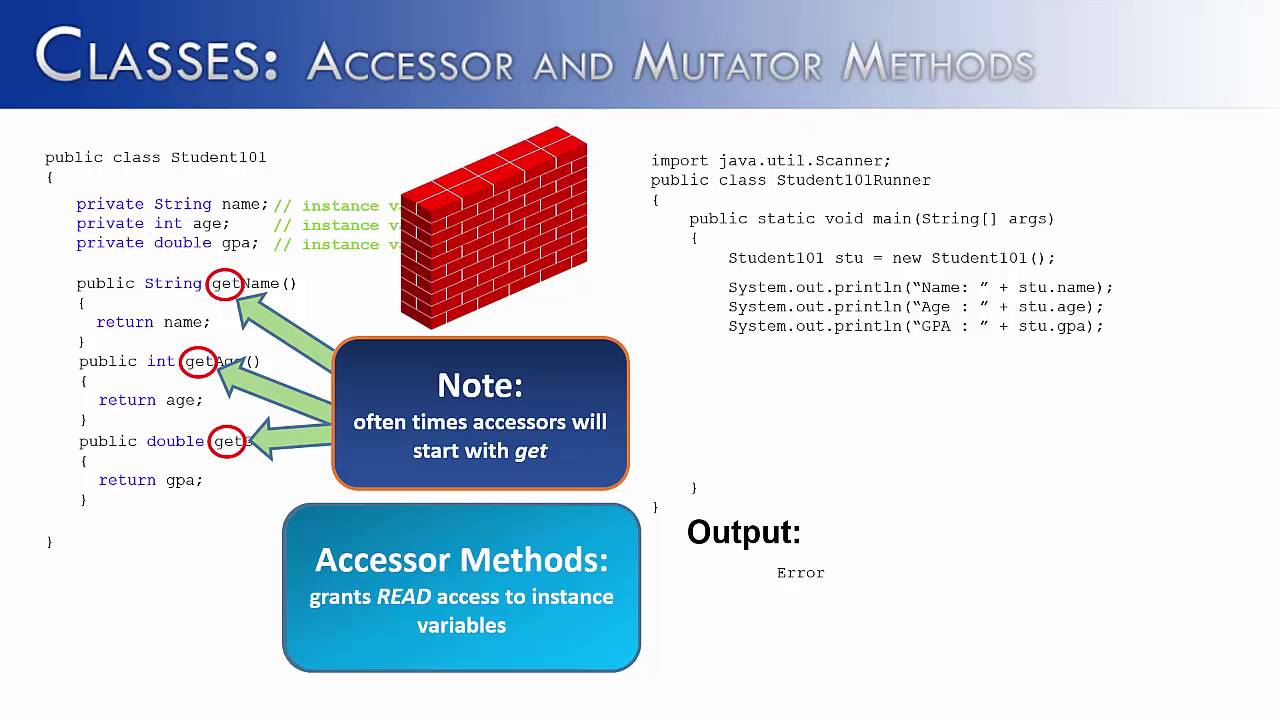
Answer:

Accessor methods are used to access fields of an object. So getters and setters are both accessor methods. Observer method is the right term for a method that makes a more general observation about an object, without causing externally observable side effects.

You should use accessor methods rather than directly accessing state data because they provide an abstraction layer. Here are just two of the benefits that accessor methods provide: You don’t need to rewrite your code if the manner in which a property is represented or stored changes.

In computer programming, an accessor method is a method that fetches private data that is stored within an object. An accessor provides the means by which to obtain the state of an object from other program parts.

It's okay for a method to return an object in terms of an interface that the object implements because that interface isolates you from changes to the implementing class. This sort of method (that returns an interface reference) is not really a "getter" in the sense of a method that just provides access to a field. If you change the provider's internal implementation, you just change the returned object's definition to accommodate the changes. You still protect the external code that uses the object through its interface.



# Question

If you create a constructor method to build a **DVD** by title then create a constructor method to build a **DVD** by category. Does JAVA allow you to do this?

Answer:

Java does not allow to do this. Because when create a constructor method to build a **DVD** by title then create a constructor method to build a **DVD** by category, they have duplicate signatures (Name constructor method: DigitalVideoDisc, datatypes: String).