



LAB 3: Introduction to Domain Modeling and Class Diagram

OBJECTIVES

- Use the UML notation to represent classes and their properties.
 - Perform domain analysis to develop domain class models.
 - Model the structural aspects of problems with the class model.
-

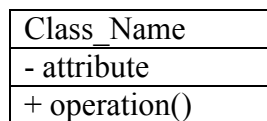
INTRODUCTION

Definition of Object

- An object is a self-contained entity with well-defined characteristics (properties or attributes) and behaviors (operations).
- Example of real-life objects include: School, Teacher, Client, Course, Account, etc.

Class Structure

- A class is a specification of a set of objects, not the actual object.
- In UML, a class is represented by a rectangle divided into three compartments (from top to bottom): the class name, a list of attributes and a list of operations.



Sample Class scheme

- Only attributes and operations relevant to the current context will be shown in a diagram. “-“ denotes an attribute while “+” denotes an operation within that class.

Attributes

Attributes refer to properties that define the class.

For example, a class Client will have attributes name, id and address.

Operations

- Functions which can be performed and related to a specific class.
 - Also commonly known as methods.
-



- Expandable to programming language, where operations are similar to functions in that they have parameters and return values.
- For example: class Account may have operations: withdraw(amount) and deposit(amount).

Definition of Domain Model

- Captures the most important types of objects in a system.
- Describing “things” in a system and how these things are related to each other.
- A “thing” can be an object, a class, an interface, a package or a subsystem, which is part of the system being developed.
- Very important process because it is employed throughout the entire system development life cycle.

Characteristics of Domain Modeling

- Visual representation of conceptual classes.
- Associations and relationships between concepts (e.g Payment PAYS-FOR Sales).
- Attributes for information content (e.g. Sale records DATE and TIME).
- Does not include operations / functions.
- Does not describe software classes.
- Does not describe software responsibilities.

Definition of Class Diagram

- A diagram that describe the structure of a system by showing the system's:
 - classes
 - its attributes and operations
 - relationships between the classes.

GENERAL STEPS IN DOMAIN MODELING

A recommended flow for domain modeling is shown below in sequential order:

1. Prepare problem statement for the system being developed.
2. Identify classes and objects.
3. Develop data dictionary.
4. Identify association between classes.
5. Identify attributes of classes and association classes.
6. Structure objects using inheritance.
7. Verify access paths for likely queries.
8. Iterate and refine the model.



Consider a sample Problem Statement for an Online Stock Trading System below:

A stock brokerage firm wants to provide an online stock trading service to enable its clients to make trades via the computer. With this system, a client must first be registered before he can trade online. The registration process involves the client providing his ID number, address and telephone number. A client may open one or more accounts for stock trading. The stock brokerage firm needs to be registered with a stock exchange before its clients can trade the stocks listed on the stock exchange. A stock brokerage firm can be registered with one or more stock exchanges. The stock brokerage firm may need to pay monthly charges for using the services provided by the stock exchange.

Once registered, the client can buy and sell stocks. The client can check the current price, bid price, ask price and traded volume of a stock in real time. The stock price and traded volume information is provided by the stock exchange on which the stock is listed and traded. When a client issues a buy order for an account, the client must specify the stock code, the number of shares and the maximum price (bid price) that he is willing to pay for them. A client must have sufficient funds in his account to settle the transaction when it is completed. When a client issues a sell order, the client must specify the stock code; the number of shares and the minimum price (ask price) that he is willing to sell them. The client must have sufficient number of shares of the stock in his account before he can issue the sell order. A client can check the status of execution of his (buy or sell) orders.

The client can issue a buy or sell order before the end of the trading day of the stock exchange which processed the order. All trade orders will be forwarded to the stock trading system of the stock exchange for execution. When an order is completed, the stock trading system of the stock exchange will return the transaction details of the order to the online stock trading system. The transaction details of a trade order may be a list of transactions, each transaction specifying the price and the number of shares traded.

For example, the result of a buy order of 20,000 Maybank (stock code: 0005) shares at MYR 12.00 in the Bursa Saham Kuala Lumpur may be as follows:

- 4,000 shares at MYR 12.00
- 8,000 shares at MYR 12.50
- 8,000 shares at MYR 12.75

An order will be kept on the system for execution until the order is completed or the end of a trading day. There are three possible outcomes for a trade order:

1. The trade order is completed. For a buy order, the total amount for the buy order will be deducted from the client's account and the number of shares of the stock purchased will be deposited into the account. For a sell order, the number of shares sold will be deducted from the client's account and proceeds of the sell order will be deposited into the client's account.

2. The trade order is partially completed. The number of shares actually traded (sell or



buy) is less than the number of shares specified in the order. The number of shares successfully traded in the order will be used to calculate the amount of the proceeds, and the client's account is adjusted accordingly.

3. The trade order is not executed by the end of a trading day. The order will be canceled.

A stock exchange may require that the number of shares specified in an order must be in multiples of the lot size of the stock. Each stock has its own lot size. Common lot sizes are 1, 400, 500, 1,000 and 2,000 shares. The client can deposit or withdraw cash or stock shares from his account. Upon the deposit or withdrawal of cash or stock shares, the account cash or stock balance will be updated accordingly.

Identifying Objects and Classes

To identify objects and classes, we first perform textual analysis to extract all available noun and noun phrases from the given problem statement.

This is to identify candidate objects that will be elaborated further.

Amour and Miller (2001) states several recommended categories to focus when finding represent-able objects.

- Tangibles (e.g. classroom, playground)
- Conceptual (e.g. course, module)
- Events (e.g. test, examination, seminar)
- External Organizations (e.g. publisher, supplier)
- Roles Played (student, teacher)
- Other System (admission system, grade reporting system)
- Attributes (e.g. cash balance)

In the case of the above problem statement, we have identified most of the nouns and noun phrases below:

Noun and Noun Phrases Extracted:

Stock brokerage firm (concept)	Buy order (event)
Monthly charge (attribute)	Stock code (attribute)
Trade (event)	Number of shares (attribute)
Trade order (event)	Transaction (event)
Computer (tangible)	Sell order (event)
Client (role played)	Trading hours (attribute)
Account (concept)	Stock trading system (other system)
Order (event)	Execution result (event)
Registration process (not an object)	Ask price (attribute)



Remove Inappropriate Classes

Now we analyze further by removing inappropriate classes, based on criteria set by categories below:

Categories	Description
Redundant classes	Classes that mean the same thing. E.g. order, trade and trade order mean the same thing. Eliminate trade and order, and retain trade order. Choose the most descriptive class.
Irrelevant classes	Classes that are not directly related to the problem. For example, monthly charge is not directly related to the system.
Vague classes	Classes that are loosely defined.
Attributes	Attributes of classes are also represented as nouns or noun phrases. Therefore, the list of nouns or noun phrases extracted by textual analysis may contain attributes of classes. E.g., address and telephone number.
Operations	The performance of actions is sometimes expressed as nouns or noun phrases. E.g. the registration process is the action taken by the client to register on the system. It should be considered an operation of a class, rather than a class.

Refer to Appendix A for complete list of inappropriate classes on the problem statement.

After excluding inappropriate classes, we will have the revised list of candidate classes:

Trade order (event)	Transaction (event)
Client (role played)	Sell order (event)
Account (concept)	Stock trading system (other system)
Stock Exchange (external organization)	Execution result (event)
Buy order (event)	Stock (concept)

Create Data Dictionary

After candidate classes have been identified, a data dictionary is prepared to record definition of a class. Each class is written a short description defining its scope / detail of the class.

Refer to Appendix B for a complete list of data dictionary of the revised classes above.



Identify Associations between Classes

Now determine all known list of associations between all classes. For example, consider the statement:

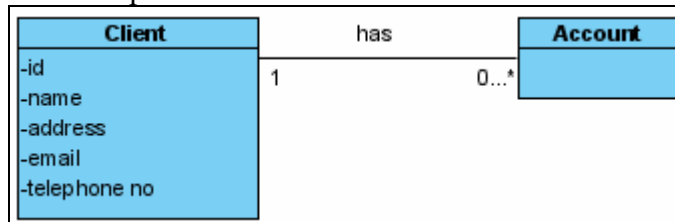
“A client may open one or more accounts for stock trading.”

This means that a client may have 1 or more account, thus denoting its relationship as: “client has [0 to *] account”. The association keyword is “has”.

This association relationship is further described by the multiplicity concept. Following the above example, a client may not have an account at all, or at minimum, 1 account. We refer to this as the number of instances of the class which is described below:

Relationship	Indicator
Exactly one	1
Zero or more (unlimited)	0...*
One or more	1...*
Zero or one only	0..1
Specified range	2..4
Multiple, disjoint ranges	2, 4..6, 8

For example:



1 client has 0 or more account registered.

Some other common UML notations concerning relationships in Classes:

Construct	Description	Syntax
Association	A relationship between two or more classifiers that involves connections between their instances.	—————
Aggregation	A special form of association that specifies a whole–parts relationship between the aggregate (whole) and the components (parts).	◊—————
Generalization	A taxonomic relationship between a more general and a more specific element.	—————>
Dependency	A relationship between two modeling elements, in which a change to one modeling element (the independent element) will affect the other modeling element (the dependent element).	- - - - ->
Realization	A relationship between a specification and its implementation.	- - - - ->



Refer to Appendix C for a revised list of all known associations between classes based on the problem statement discussed.

Identify Attributes of Classes and Association Classes

Determine all the known attributes of each class. For example, class Client may have the attributes name, address and telephone number.

However, it is not necessary to completely define all attributes in order to make successful domain class model at this stage. It is not as important as identifying the associations of classes.

Structuring classes with Inheritance

Identify any requirement / possibility to restructure the domain class model using inheritance. (Inheritance concept has a similar relationship explained under Generalization in Lab 2). This can be done in 2 ways of identification: “Bottom Up” and “Top Down”

Bottom Up approach: Compare properties of classes to look for commonality. For example:

Buy Order and Sell Order shares similar attributes (price, number of shares). A superclass can then be defined as Trade Order (Trade encompassing both buy and sell).

Top Down approach: Look from a broader view to identify specializations. For example: Cash Account and Margin Account is a specialization of Account.

Iterate and Refine the Model

It is almost unlikely that the correct domain class model is developed in one pass.

- Must be refined several times before it becomes robust.
- Necessary to repeatedly apply the above steps until the domain class model is finally stable.

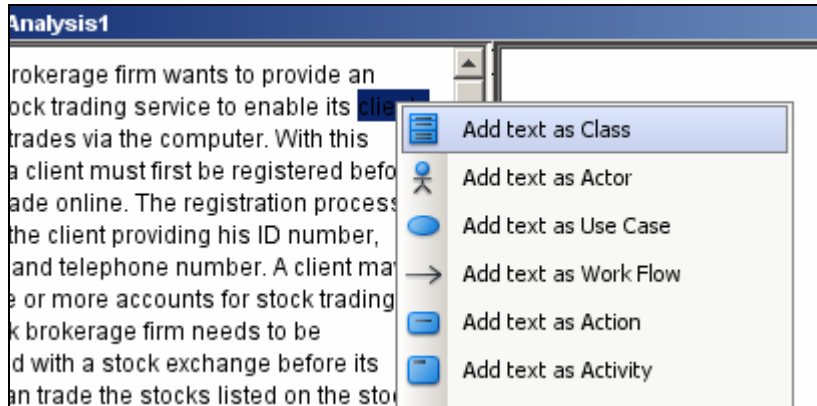
DOMAIN MODELING WITH VP-UML

Step 1: Prepare the Problem Statement

- Create a new Textual Analysis (as in Lab 2), and copy the sample stock trading problem statement above.

Step 2: Identify Objects and Classes

- Once the problem statement is defined, next is to identify objects and classes in the textual analysis working area.
- Highlight the term “client” as a candidate class from the textual analysis (lab 2).



Highlighting the term client as class candidate

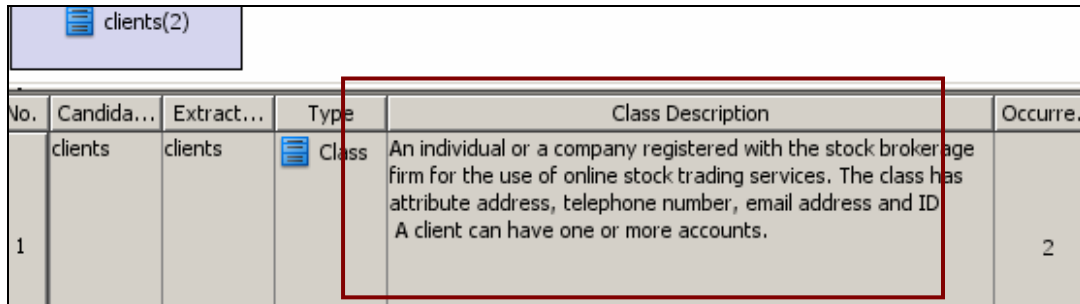
- Repeat the above steps to identify the remaining classes:
 - Trade Order
 - Account
 - Stock Exchange
 - Buy Order
 - Transaction
 - Sell Order
 - Stock Trading System
 - Execution Result
 - Stock

Step 3: Develop Data Dictionary

- To further define the candidate classes selected as in step 1, enter the following description in the Class Description cell (under the Object Description sub-window).

An individual or a company registered with the stock brokerage firm for the use of online stock trading services. The class has attribute address, telephone number, email address and ID. A client can have one or more accounts.

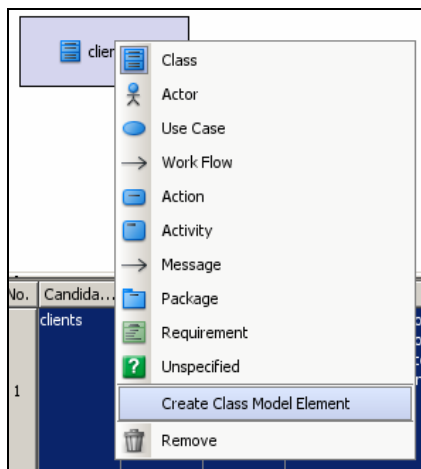
Class Description cell for Client



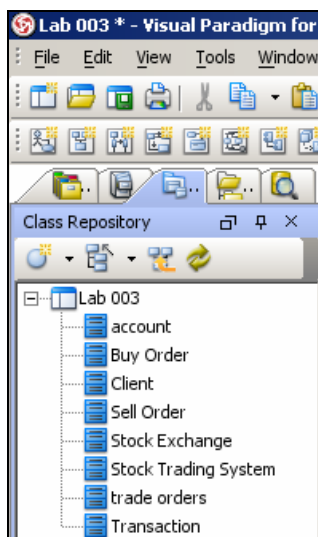
No.	Candida...	Extract...	Type	Class Description	Occurre..
1	clients	clients	Class	An individual or a company registered with the stock brokerage firm for the use of online stock trading services. The class has attribute address, telephone number, email address and ID. A client can have one or more accounts.	2

Candidate class Client description

- Repeat the above steps to complete the dictionary for all remaining candidate classes (Refer Appendix B).
- We can now create the models from the candidate classes. “Right click” on the candidate class and select “Create Class Model Element”.



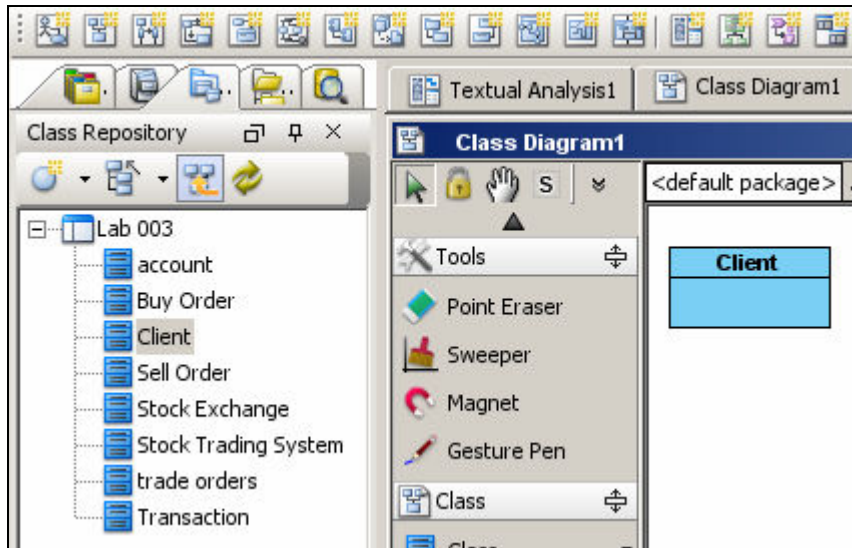
Creating a class model element from candidate class.



Candidate classes listed under Class Repository.

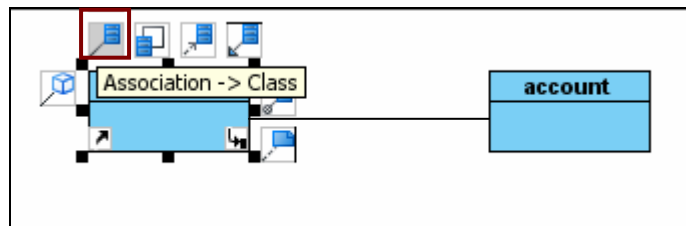
Step 4: Identification of Associations between Classes

- Create a new Class Diagram (refer to lab 1).
- Drag class client (displayed under Class Repository) into the Class Diagram pane.



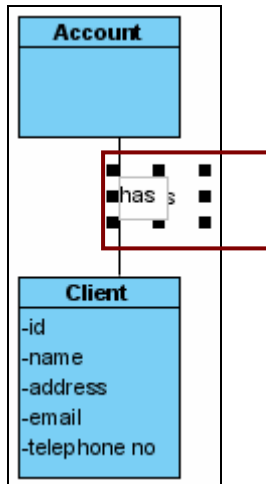
Using defined class Client in Class Diagram

- Repeat the previous steps to create class Account in the Class Diagram.
- Now create an association between classes “Client” and “Account”.
 - “Left click” the class Client and click the association icon from the Resource centric interface and drag it to class Account.



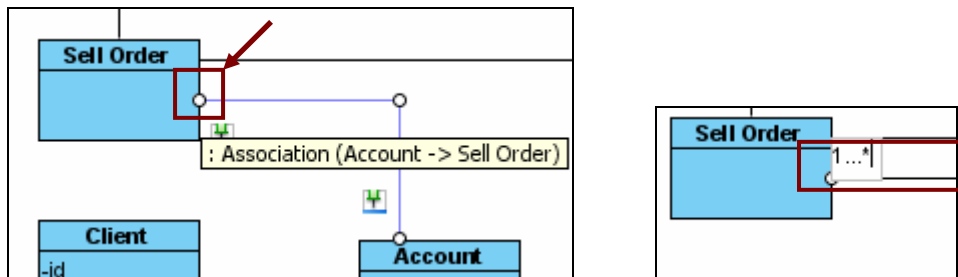
Association between classes example

- Repeat the above steps to complete all other associations (Refer to Appendix C for description of all other classes)
- Now double click the association line connecting the classes Account and Client. An in-line text editing area will be displayed to enter a text as shown below. Type the text “has”.



Defining association between classes

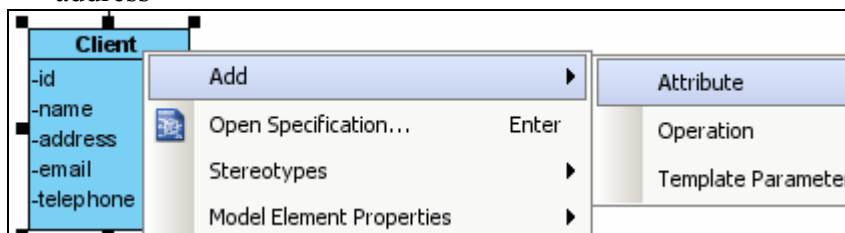
- To add numeric associative (multiplicity) relationship between classes, do this by “double clicking” the end connector dots of the association line to add numerical relationship status as shown below:



Step 5: Identify Attributes of Classes and Association Classes

We will now concentrate on adding attributes of the classes.

- Select class Client. Right click on the class Client, and then select New Attribute. Add the following as attributes for class Client:
 - Id
 - Name
 - Telephone
 - address

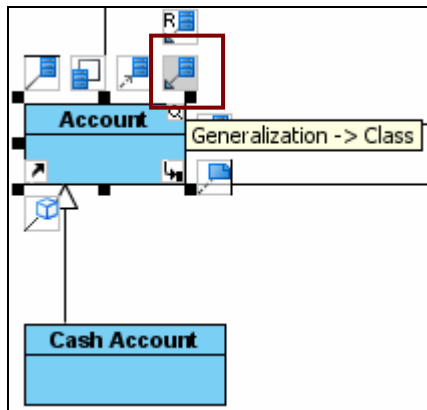


Right click the class to access this menu

Step 6: Structure Object Classes using Inheritance

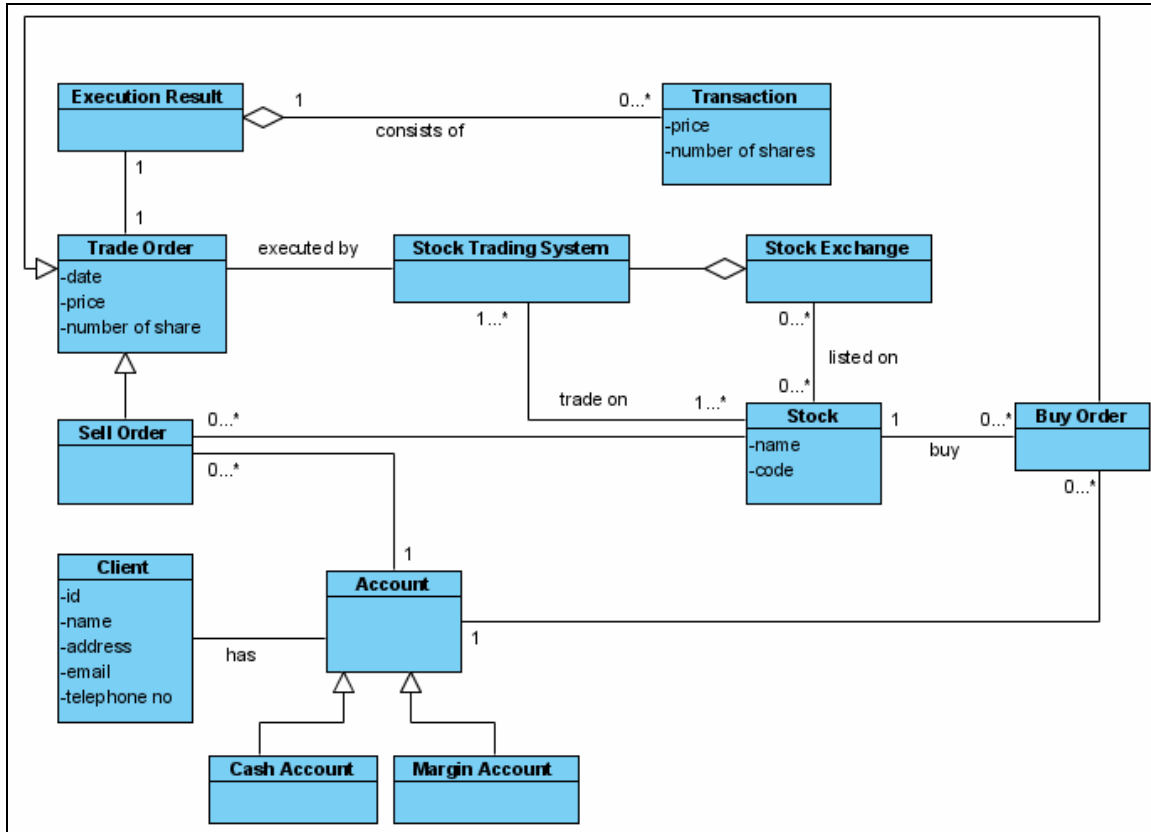
By adopting a top-down approach, we discover that the class Account has 2 subtypes, “Cash Account” and “Margin Account”. To structure them using inheritance, we need to create them as subclasses.

- Select the class “Account”. On the resource centric interface, click and drag the icon for “Generalization -> Class” to create a new subclass. Name it as Cash Account. Repeat the step again by including subclass Margin Account.



Generalization option under Resource centric option

The restructured domain class diagram is shown below:



Restructured Domain Class Diagram of an Online Stock Trading System

Step 7: Verifying Access Paths for Likely Queries

At this stage, we usually need to enhance the domain class diagram by including support for typical queries of the application domain. Sometimes during the modeling phase, there will be other type of queries / between associated classes. Consider a query based on our example above:

How does a client find out the stock balance of his account?

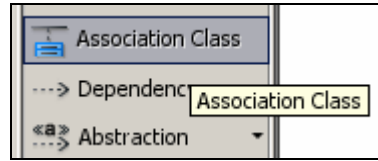
The Domain Model cannot directly answer this as it only show transactions between classes (and remember we can't put in operations under the classes at this stage). To do this:

We need to associate class Account with Stock.

- Create an association between class Account and Stock as in Step 4.

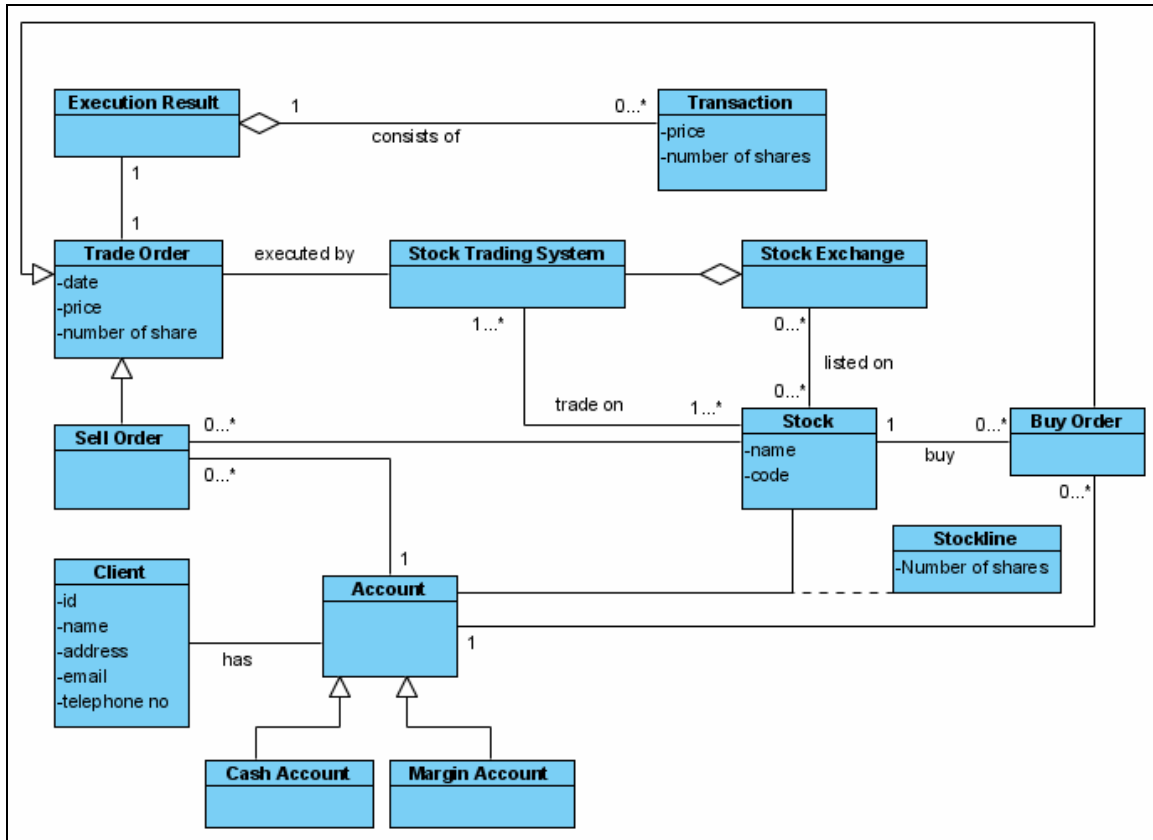
Then, we need to create an "Association Class" between these 2 classes.

- Click the "Association Class" icon in the Diagram Palette, which will create a class. Name it as class "Stockline".



The Association Class option under Diagram Palette

- Highlight the class “Stockline” and drag it to the association line between Account and Stock.



A revised Domain Model for Online Stock Trading System

Step 8: Iterate and Refine Model

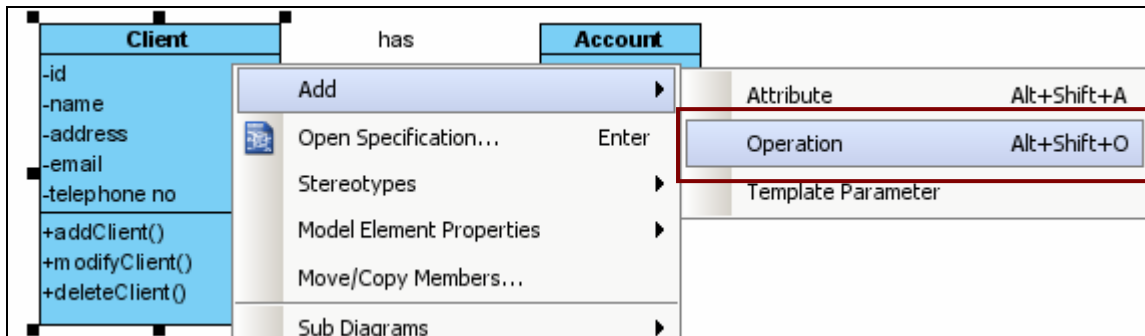
Repeatedly apply the steps 2 to 7 to refine the domain class model until it becomes stable.

CLASS DIAGRAM MODELING

Doing both domain (this lab) and use case modeling (covered in Lab 2) will yield improved results in our class models. We can now define in much more detail the classes and its important attributes and operations involved in the class modeling.

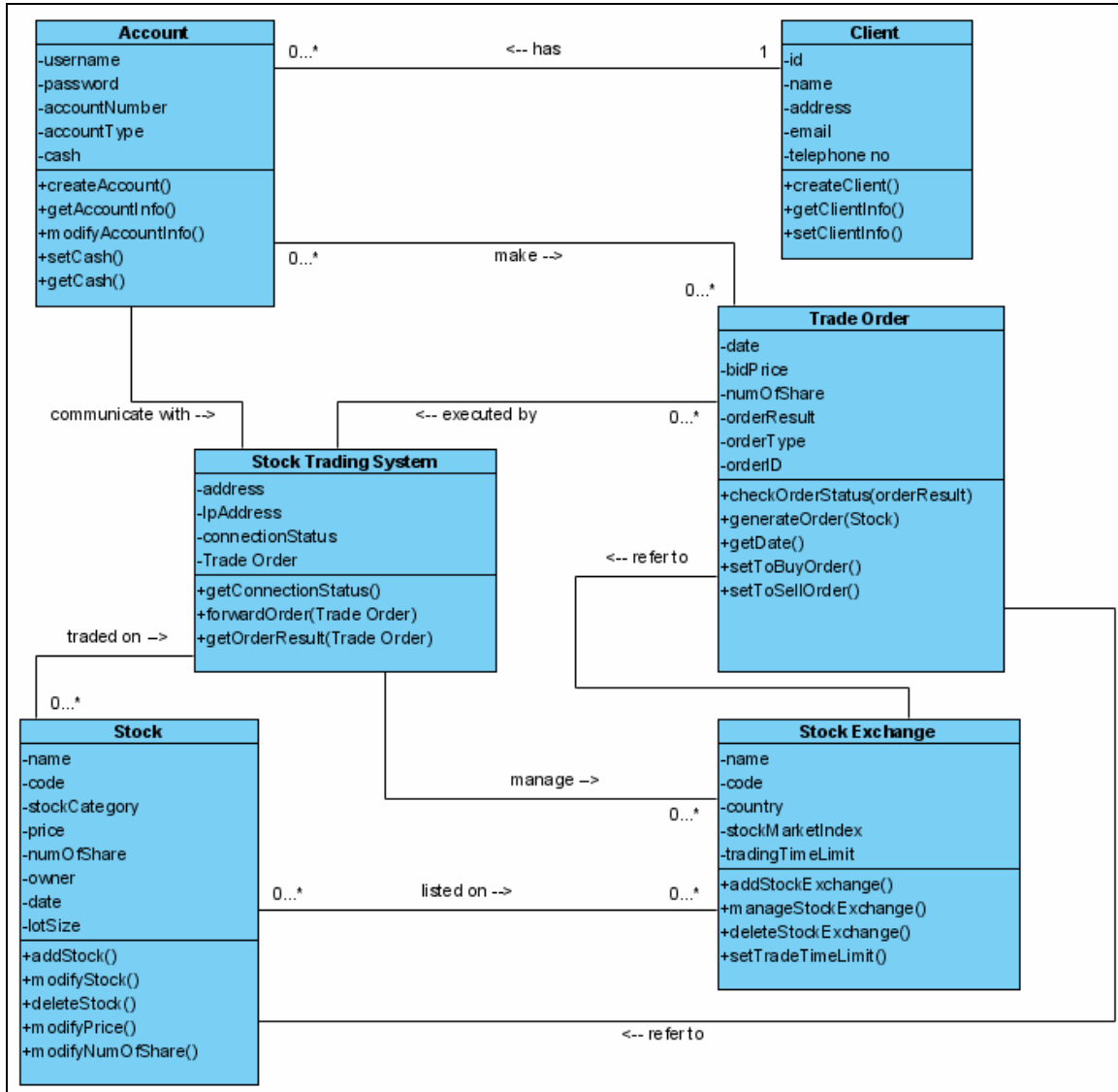
Assuming that we have completed our Domain / Use Case modeling phase, we will start designing the Class Diagram. It is important to note that our Class Diagram will implement in more specific the attributes and operations concerned with each classes. This is demonstrated by following the steps below:

- Create a new Class Diagram.
- Drag class Client from the Class Repository list into the Class Diagram.
- Add operations [addClient(), modifyClient() and deleteClient()] under class Client by “Right clicking” the class and choosing “ Add > Operation”.



Adding operations under class

- Now try to complete your Class Diagram by following the a well described Class Diagram shown below:



A well defined Class Diagram for the Online Stock Trading System (after much requirement analysis)



APPENDIX A – Common List of Inappropriate Classes

Stock brokerage firm (irrelevant)	Stock code (attribute)
Monthly charge (irrelevant)	Number of shares (attribute)
Trade (redundant)	Maximum price (attribute)
Trading hours (attribute)	Current price (attribute)
Lot size (attribute)	Bursa Saham Kuala Lumpur (instance)
ID (attribute)	Order (redundant)

APPENDIX B – Complete Data Dictionary

Class	Definition
Client	An individual or a company registered with the stock brokerage firm for online stock trading services. The class has attributes address, telephone number and ID. A client may have one or more accounts.
Account	A client can issue trade order on his or her accounts. An account holds details about the cash and stock balances for trading.
Stock exchange	A financial institution that provides a platform where stock trading is carried out.
Stock trading system	A platform for the execution of the trade orders of stock.
Trade order	A trade order specifies the price, stock code and number of shares. A trade order can be a buy order or a sell order.
Buy order	A buy order specifies the bid price, stock code and number of shares.
Sell order	A sell order specifies the ask price, stock code and number of shares.
Stock	A company listed in a stock exchange. Shares of a company can be traded only in a multiple of its lot size.
Execution result	The result of the execution of a trade order. It contains a list of transactions.
Transaction	The execution of a trade order at a particular price. It also contains the number of shares traded at that price.

APPENDIX C – Complete List of Association between Classes

Verb Phrase	Association
A client may open one or more accounts for stock trading.	Has
When a client issues a buy order for an account, the client must specify the stock code, the number of shares and the maximum price that he is willing to pay for them (the bid price).	Issued by, buy



When a client issues a sell order for an account, the client must issued by, sell specify the stock code, the number of shares and the minimum price that he is willing to sell them at (the ask price).	Issued by, sell
All trade orders will be forwarded to the stock trading system of executed by the stock exchange for execution.	Executed by
When an order is completed, the stock trading system of the stock returned by exchange will return the transaction details of the order to the online stock trading system.	Returned by
The transaction details of a trade order may be a list of consists of transactions, and each transaction specifies the price and the number of shares traded.	Consists of