

Class Diagram

Lecture 3C

U08182 © Peter Lo 2010

1

Elements of a Class Diagram

- A class diagram is a diagram showing a collection of classes and interfaces, along with the collaborations and relationships among classes and interfaces.
- A class diagram is composed primarily of the following elements that represent the system's business entities:
 - ◆ Class
 - ◆ Interface
 - ◆ Package

U08182 © Peter Lo 2010

2

Class

- A class represents an entity of a given system that provides an encapsulated implementation of certain functionality of a given entity.
- These are exposed by the class to other classes as methods.
- Apart from business functionality, a class also has properties that reflect unique features of a class.
- The properties of a class are called attributes.
- Simply put, individual members of a family of our family tree example are analogous to classes in a class diagram.

Interface

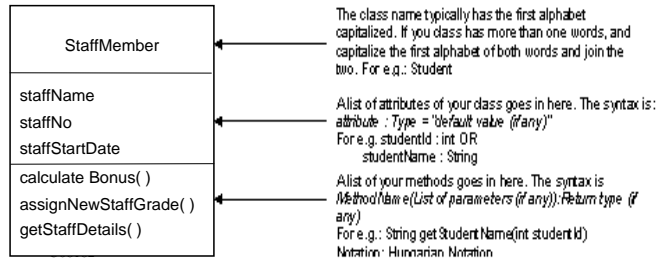
- An interface is a variation of a class.
- A class provides an encapsulated implementation of certain business functionality of a system.
- An interface on the other hand provides only a definition of business functionality of a system.
- A separate class implements the actual business functionality.

Package

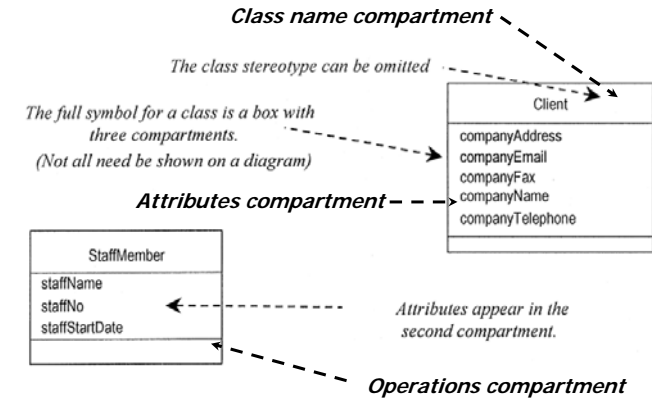
- A package provides the ability to group together classes and/or interfaces that are either similar in nature or related.
- Grouping these design elements in a package element provides for better readability of class diagrams, especially complex class diagrams.

Structure of a Class

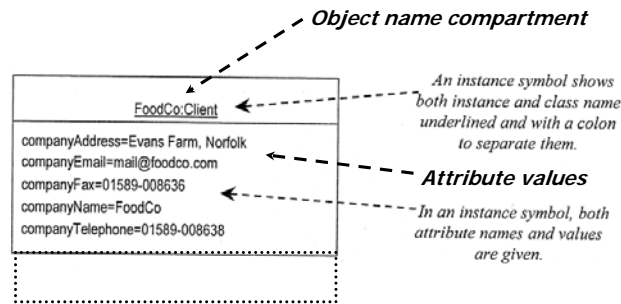
- A class is represented by a rectangle.
- The following diagram shows a typical class in a class diagram:



Class Diagram: Class Symbol



Class Diagram: Attributes



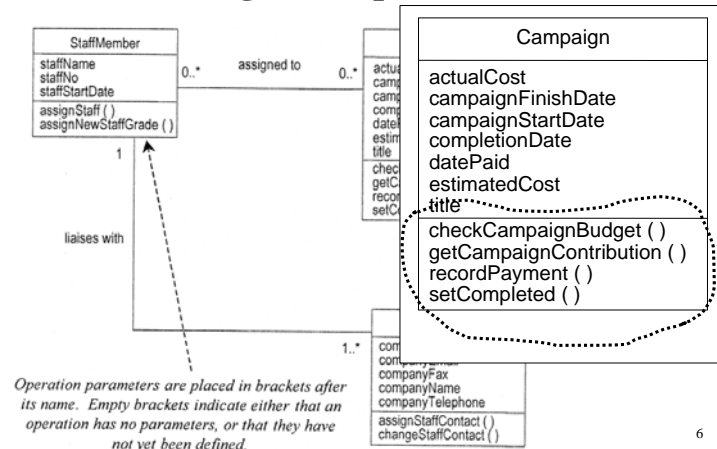
U08182 © Peter Lo 2010

5

Attributes are:

- Part of the essential description of a class
- The common structure of what the class can 'know'
- Each object has its own *value* for each attribute in its class

Class Diagram: Operations



6

Operations are:

- An essential part of the description of a class
- The common behaviour shared by all objects of the class
- Services that objects of a class can provide to other objects

Operations describe what instances of a class can do:

- Set or reveal attribute values
- Perform calculations
- Send messages to other objects
- Create or destroy links

Stereotype

- During the early phase of the system design conception, classes called **Analysis classes** are created.
- Analysis classes are also called **Stereotypes**.
- In the UML context, stereotypes are UML models that that represent an existing UML element, while showing additional characteristics that are common across the classes to be used for that application.
- Only one stereotype can be created for any UML element in the same system.

U08182 © Peter Lo 2010

7

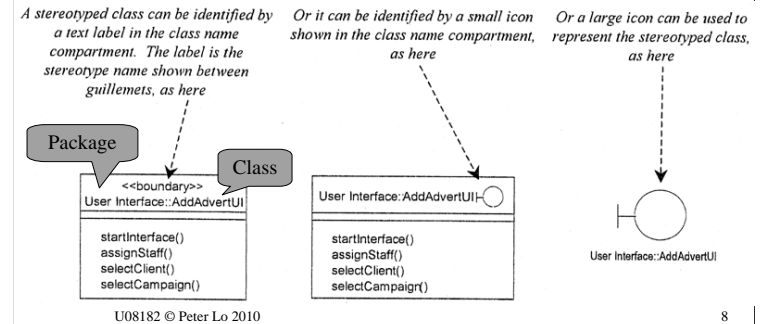
Type of Stereotype

Analysis class (Stereotypes) differentiate the roles objects can play:

- Boundary Class:** In an ideal multi tier system, the user interacts only with the boundary classes
- Control Class:** These classes typically don't contain any business functionality. However, their main task is to transfer control to the appropriate business logic class, depending on a few inputs received from the boundary classes.
- Entity Class:** These classes are those that contain the business functionality. Any interactions with back-end systems are generally done through these classes.

Boundary Class

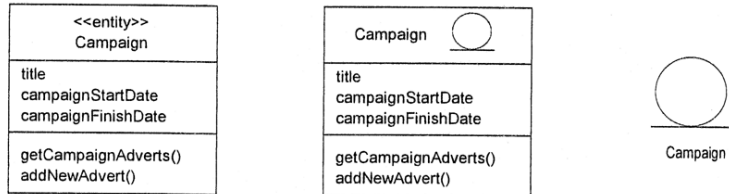
- Boundary Classes model interaction between the system and actors.



8

Entity Class

- Entity Class represent information and behavior of some phenomenon or concept such as an individual, a real-life object, or a real-life event.

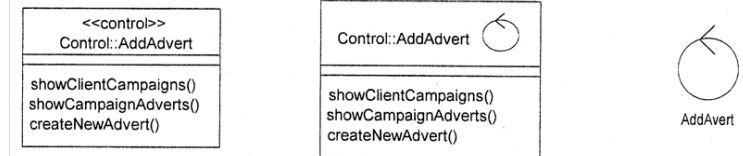


U08182 © Peter Lo 2010

9

Control Class

- Control Class represent coordination, sequencing, transactions and control of other objects.

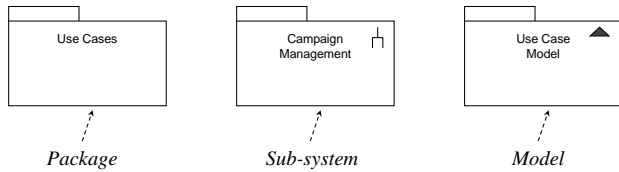


U08182 © Peter Lo 2010

10

Package

- A package is represented as a tabbed folder.
- A package can also have relationships with other packages similar to relationships between classes and interfaces



U08182 © Peter Lo 2010

11

Relationships Between Classes

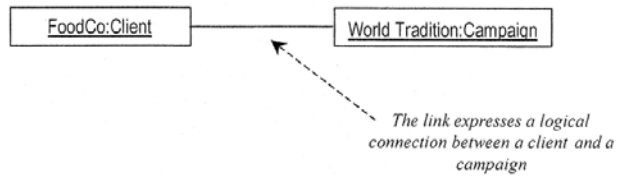
- In a class diagram, obviously you can't have classes just floating around; you need to see the relationship between them.
 - ◆ Link
 - ◆ Associations
 - ◆ Aggregation
 - ◆ Composition

U08182 © Peter Lo 2010

12

Link

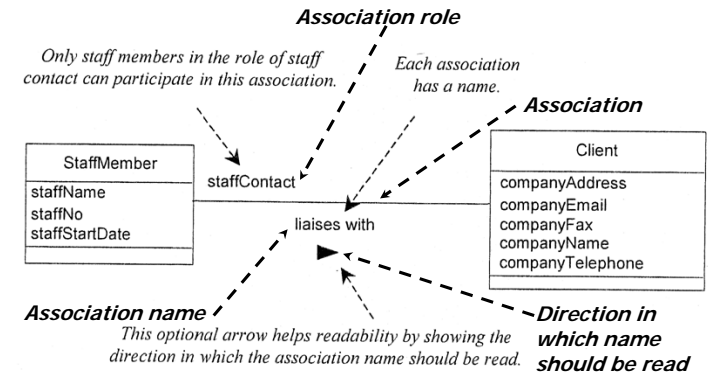
- A link expresses a logical connection between instances/objects



U08182 © Peter Lo 2010

13

Associations



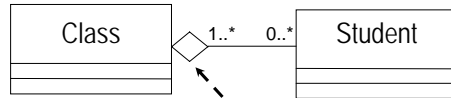
U08182 © Peter Lo 2010

14

- When two classes are connected to each other in any way, an association relation is established.
 - For example: A "student studies in a college" association can be shown as:
- Associations represent:
 - The possibility of a logical relationship or connection between objects of one class and objects of another
 - If two objects can be linked, their classes have an association

Aggregation

- An everyday example

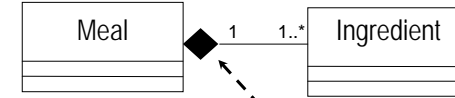


*Unfilled diamond
signifies aggregation*

- Clearly not composition
 - ◆ Students could be in several classes
 - ◆ If class is cancelled, students are not destroyed!

Composition

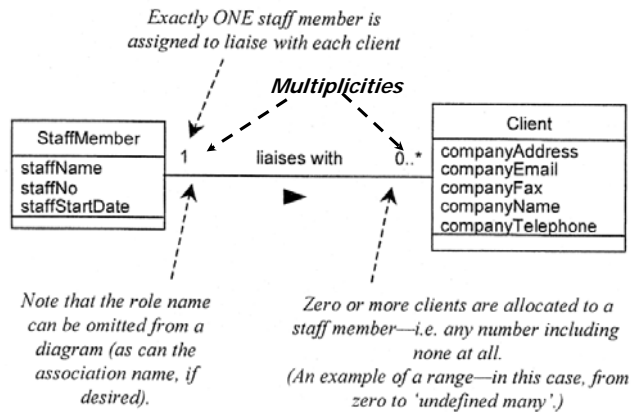
- Another everyday example



Filled diamond signifies composition

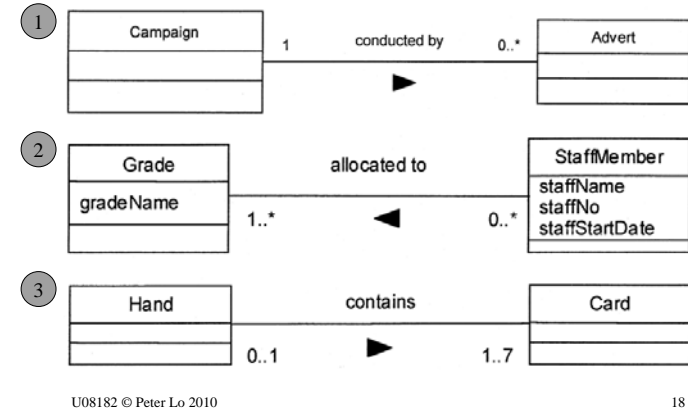
- This is (probably) composition
 - ◆ Ingredient is in only one meal at a time
 - ◆ If you drop your dinner on the floor, you probably lose the ingredients too

Class Diagram: Multiplicity



- Associations have multiplicity
- Multiplicity is the range of permitted cardinalities of an association
- Represent *enterprise* (or *business*) rules
- For example:
 - Any bank customer may have one or more accounts
 - Every account is for one, and only one, customer

Class Diagram: Multiplicity Example



1. A Campaign is conducted by zero or more Adverts while each Advert belongs to exactly one Campaign
2. Every StaffMember must be allocated to one or more Grades, while a Grade may have zero, one or more staff allocated to it.
3. A Poker Hand contains up to 7 cards. Each card dealt must be in only one hand (although a card may still be undealt in the pack). This assumes no cheating.