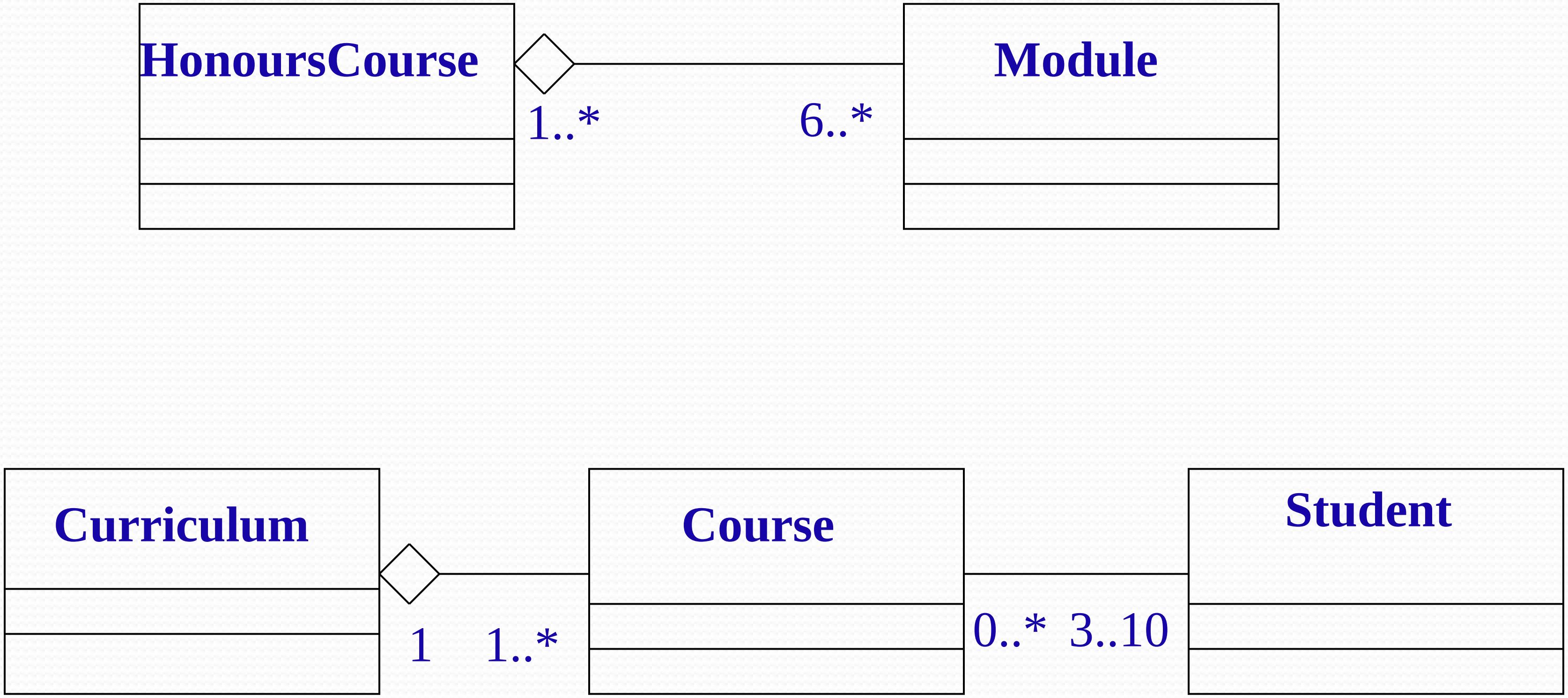
5. Advanced Class Relationships

- Associations can be used to model all relationships between objects
- UML identifies two special forms of relationship:
  - *Aggregation*: 'part of' relationship
  - *Inheritance*: implementation of class hierarchies through generalisation

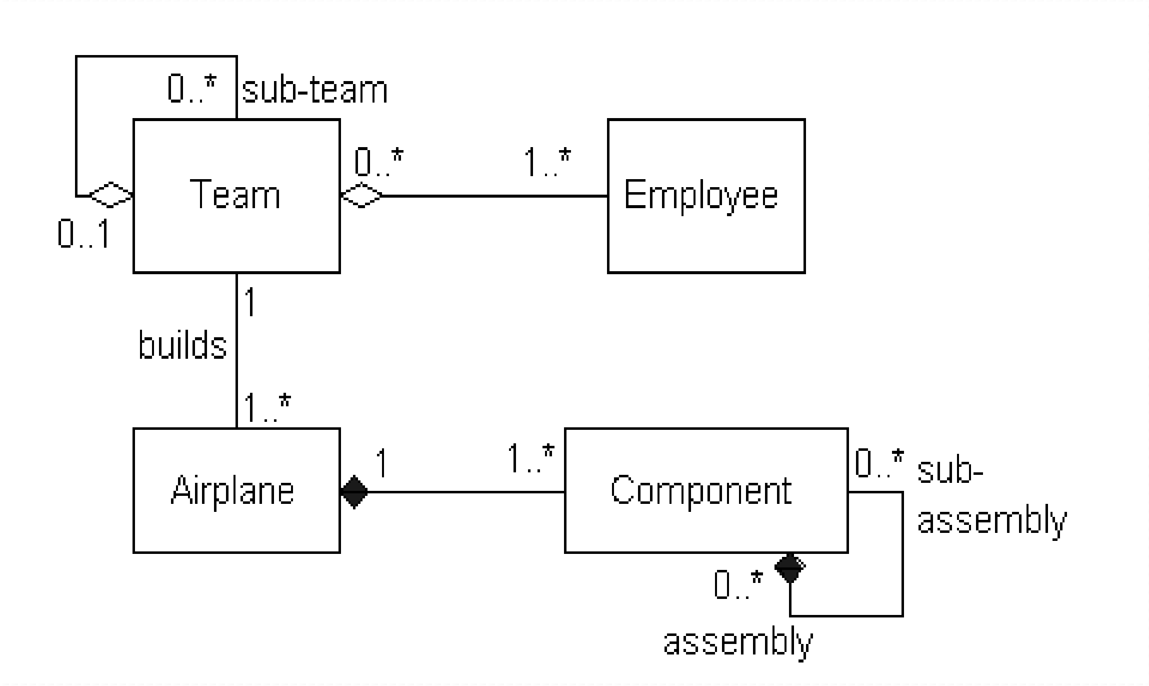
Aggregation

- A specialised form of association in which a whole is related to its part(s)
- Represented as an association with a diamond next to the aggregate class (whole)
- Tests:
  - “Part of”? And if so, Are some operations on the whole automatically applied to its parts?
  - Are some attribute values propagated from the whole to all or some of its parts?

Aggregation Example



Aggregation & Composition

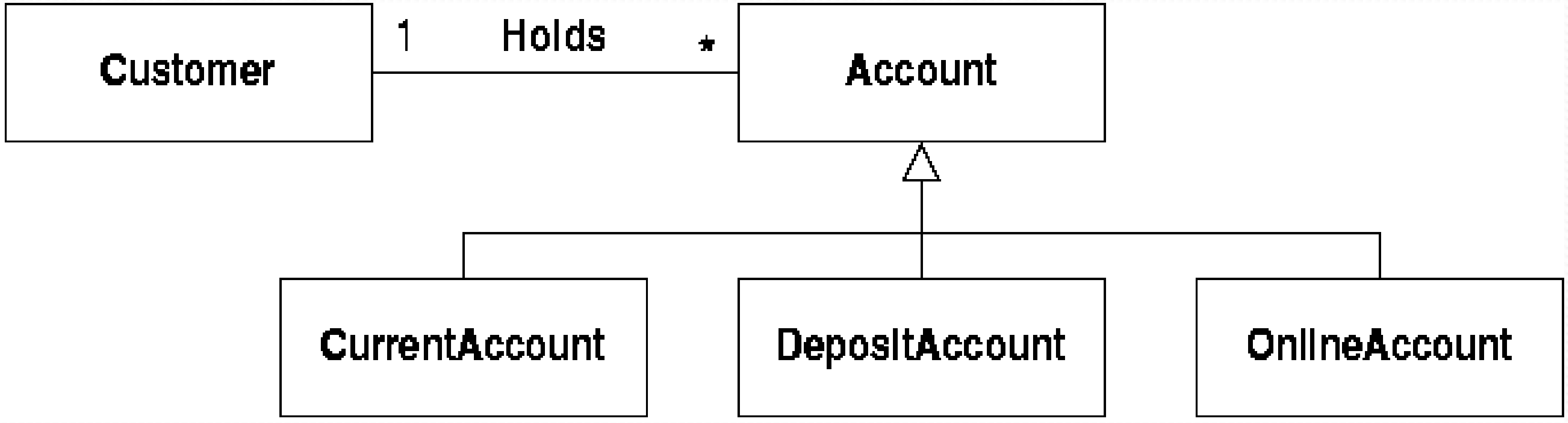


Generalisation

- Generalisation enables the creation of superclasses that encapsulate structure & behaviour common to several subclasses
- Represents conceptual relationship between classes
- Superclasses are more abstract than their subclasses
- Specialisation refines the structure & behaviour of superclass into subclasses
- Look for common behaviour and state in classes

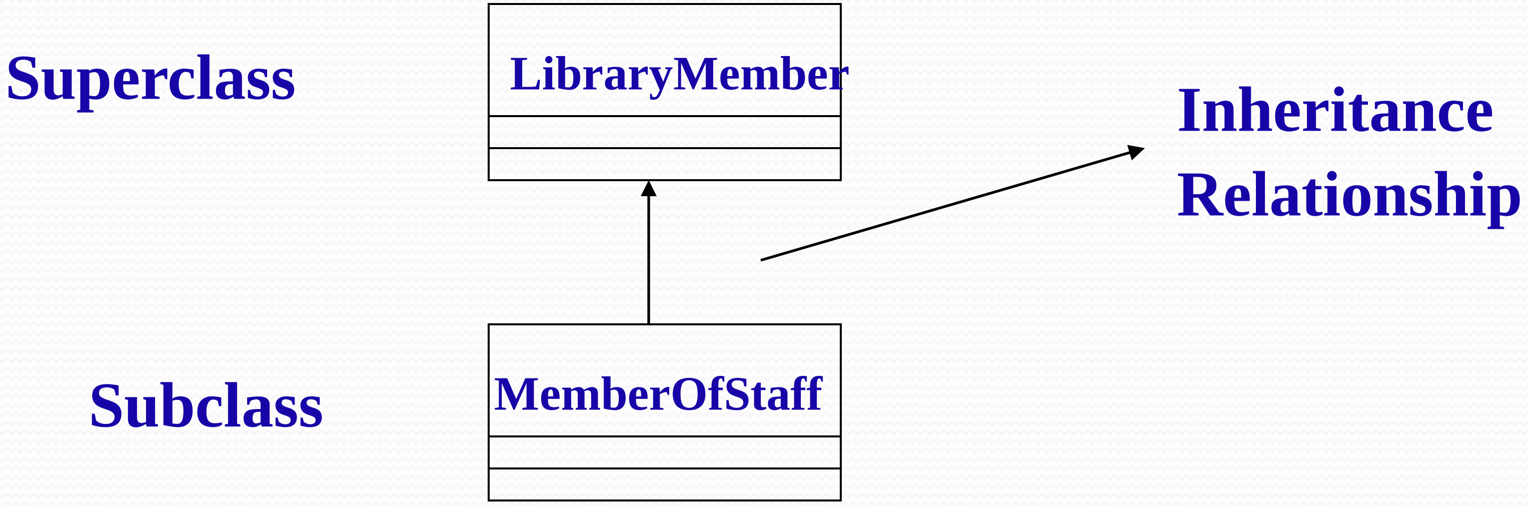
Generalisation

- The superclass defines the properties shared by all the specialised classes
  - eg customers can hold accounts of any sort

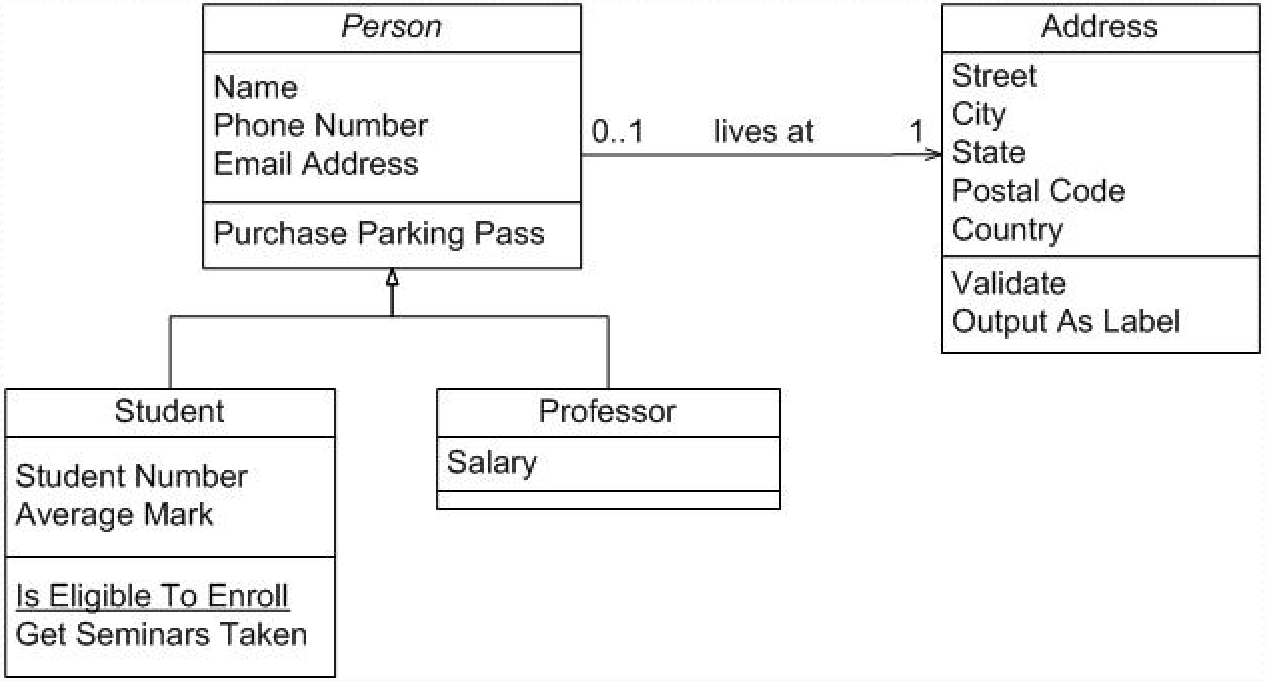


Inheritance

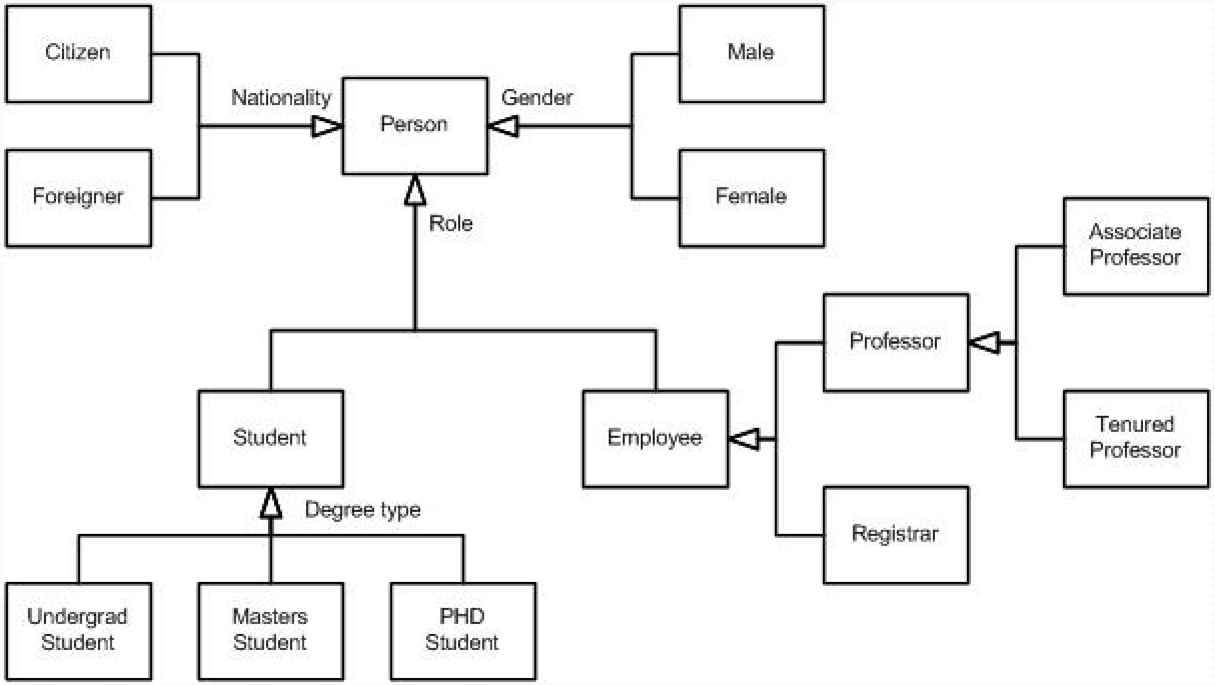
- Defines an implementation relationship among classes where one class shares the structure and/or behaviour of one or more classes
- An "is a" or "kind of" relationship



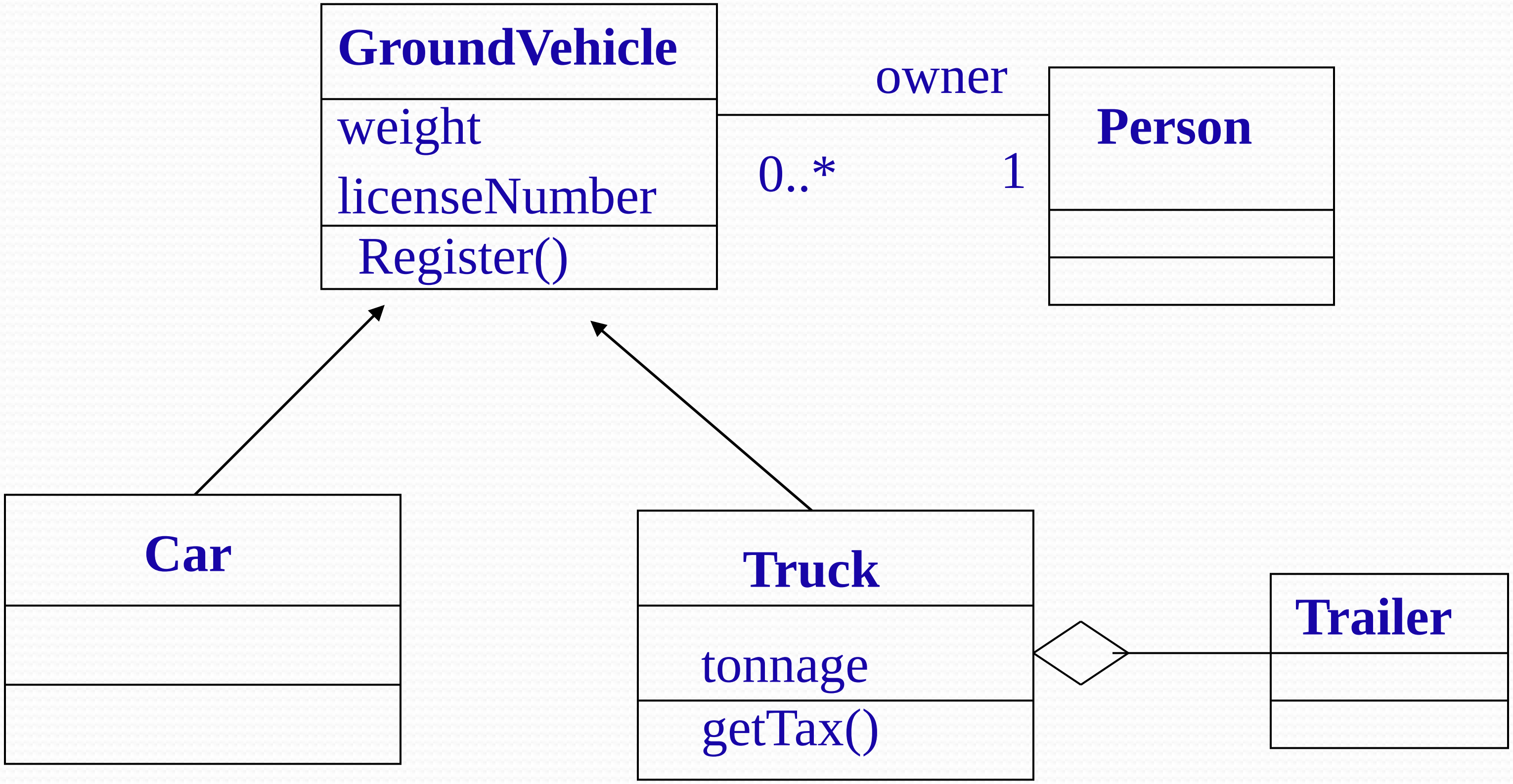
Inheritance



Inheritance



Aggregation & Inheritance

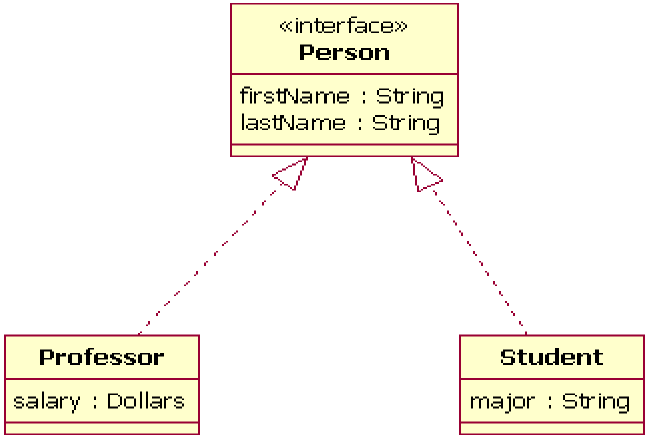


Inheritance vs. Aggregation

Inheritance	Aggregation
Keywords “is a”	Keywords “has a”
One class	Relates objects in different classes
Represented by an arrow	Represented by a diamond

6. Class: Interface

- UML 2: Interface is specialisation of a class
- Class can have actual instance of its type; Interface must have at least one class to implement it



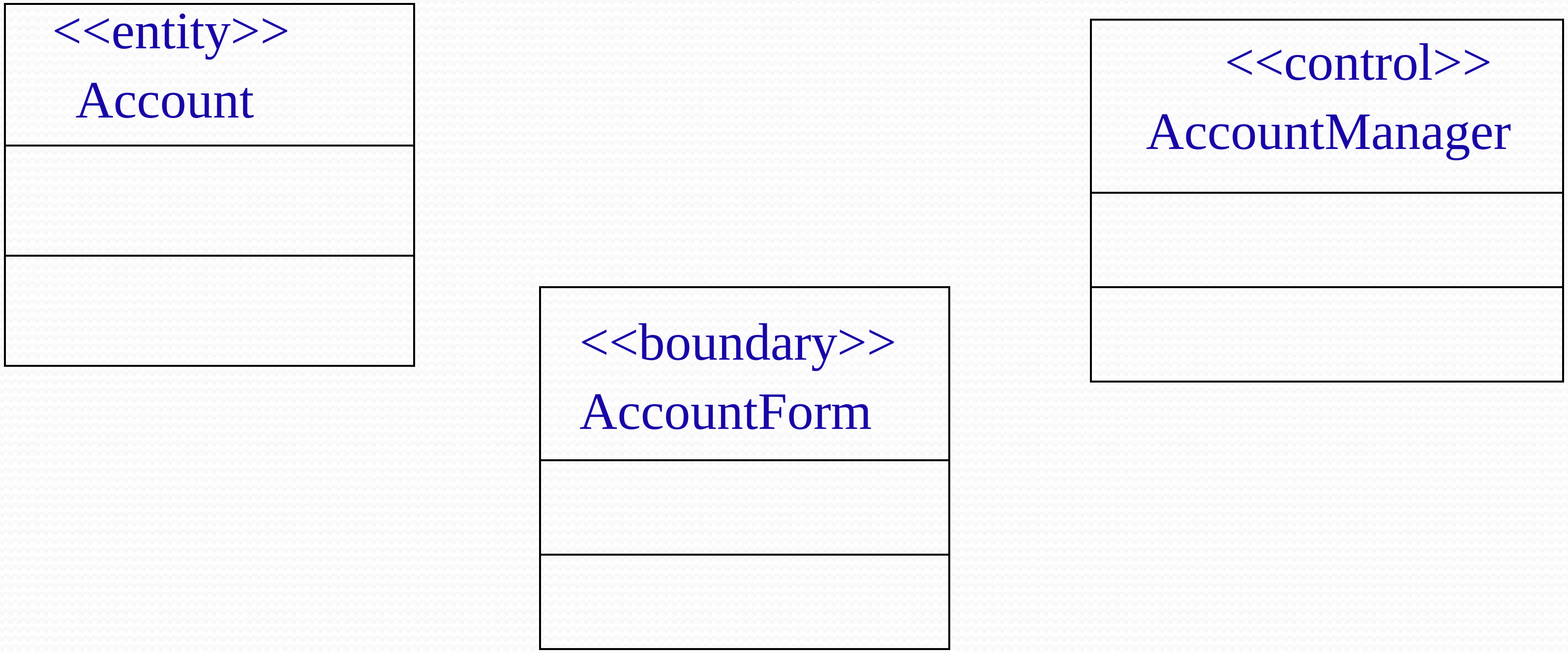
7. Class Model Organisation

- Class diagrams can become very large for even average sized systems
- Average system could result in over 100 classes
- In order to promote readability, and organise the presentation of this detail, UML has two different organisational mechanisms:
    - **Stereotype**: facilitates grouping of classes into common types
    - **Package**: partitions full class model into groups of elements with a common theme

Stereotypes

- Means of grouping objects that makes sense: attaches extra classifications to model items
- Every class may have at most one stereotype
- Analysis class stereotypes:
  - Boundary objects model interaction between the system and actors (and other systems)
  - Entity objects represent information and behaviour in the application domain
  - Control objects co-ordinate and control other objects

Stereotypes: Examples



<<entity>>
Account

<<boundary>>
AccountForm

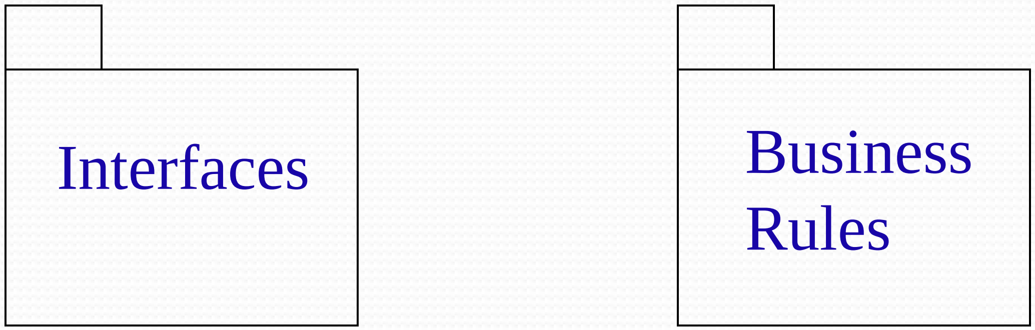
<<control>>
AccountManager

Packages

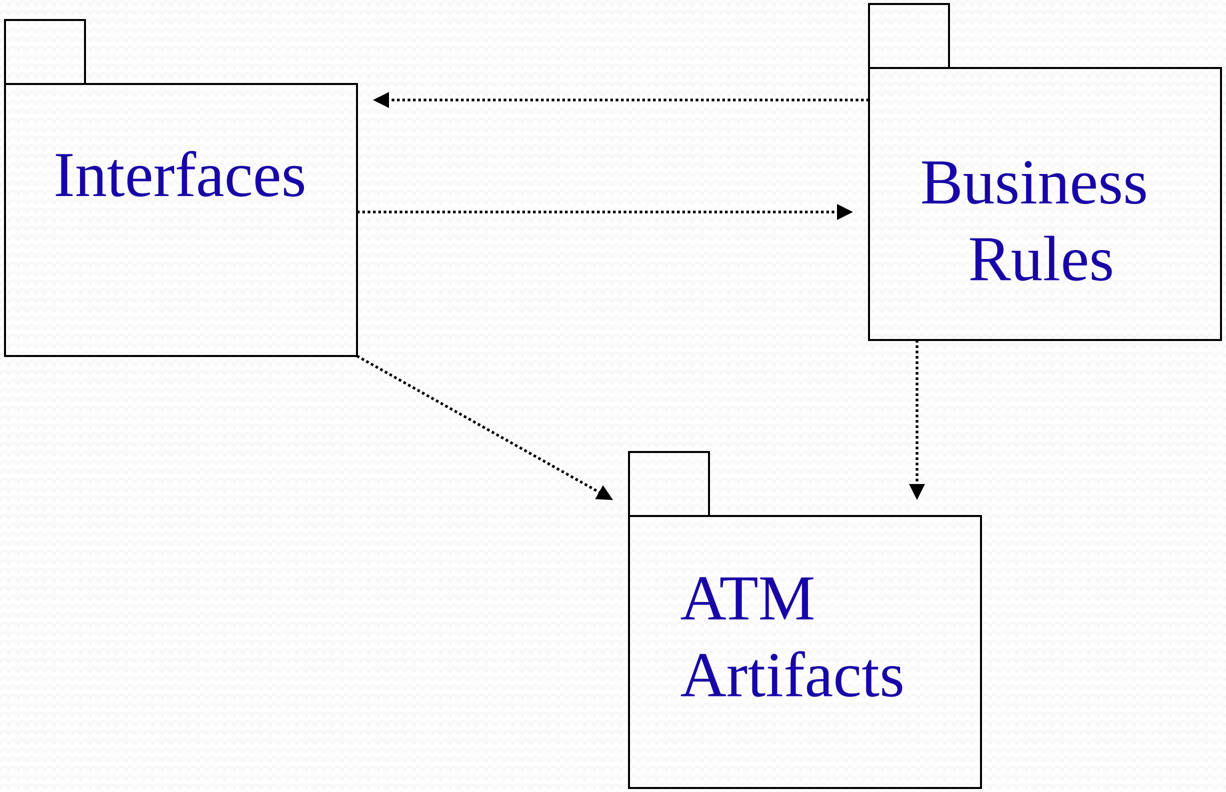
- A general purpose mechanism for organising elements into groups
- Shows the breakdown of larger systems into a logical grouping of smaller subsystems
- Package defines a namespace for these subsystems
- Organises the model under development
- A package is represented as a tabbed folder

Packages & Class Diagrams

- A class diagram is a view of some (or all) of the packages and classes in the logical view
- Main class diagram is a view of the high level packages in the logical view
- Each package has its own main class diagram
- Additional class diagrams are added as needed



Main Class Diagram



7. Case Studies

- UML Class Modeling Process:
  - Find classes and associations first
  - Add attributes
  - Combine and organise classes with inheritance
- UML Class Model Example:
  - ATM System

ATM System

- ATM Problem description is presented in Blaha and Rumbaugh 11.3.1, pg. 178
- The domain model for this system is covered in detail in chap. 12 of Blaha and Rumbaugh text
- Key elements of this class model are highlighted in the next five slides to illustrate the class modeling concepts covered in this workshop
- The focus is on understanding the real-world essence of a problem: real-world classes and their relationships to each other

ATM System: Associations

- Problem analysis: finding associations
- Structural relationships between two or more classes
- Often correspond to verbs: physical location, directed actions, communication, ownership, and satisfaction of some condition
- List all possible associations and discard unnecessary ones: irrelevant, derived, qualified

Lecture 3 Review

- UML Class Model

1. Class Diagram
2. Attributes & Operations
3. Relationships
4. Finding Classes
5. Relationships: Aggregation, Generalisation
6. Organisation: stereotypes & packages

- Case Study:

7. Example Class Model: ATM