SYS466:
Analysis and Design using OO Models

Lecture 3: Associations

Domain Model Relationships

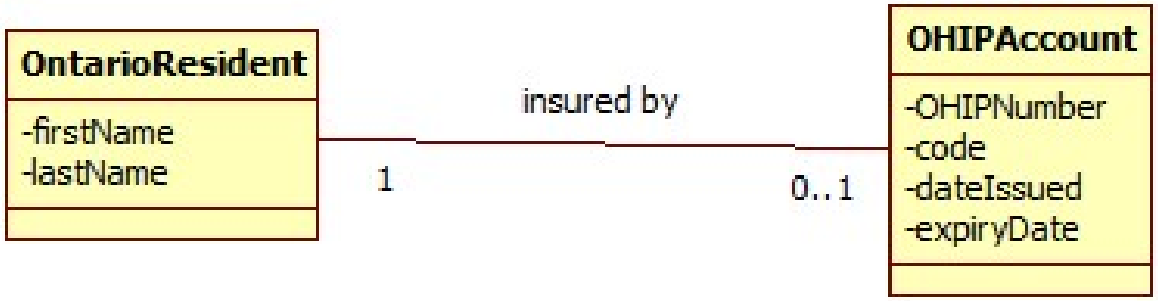
Classes/objects participate in a variety of relationships with other classes/objects including the following:

- **Simple Associations**—objects from associated classes know about each other and can pass messages and invoke functions. Can be uni-directional or bi-directional.
- **Compositions**—containment—a container class “contains” other classes.
- **Generalizations**—a relationship in which specific “child” or sub classes are derived from a generalized “parent” or super class.

When do we use Associations?

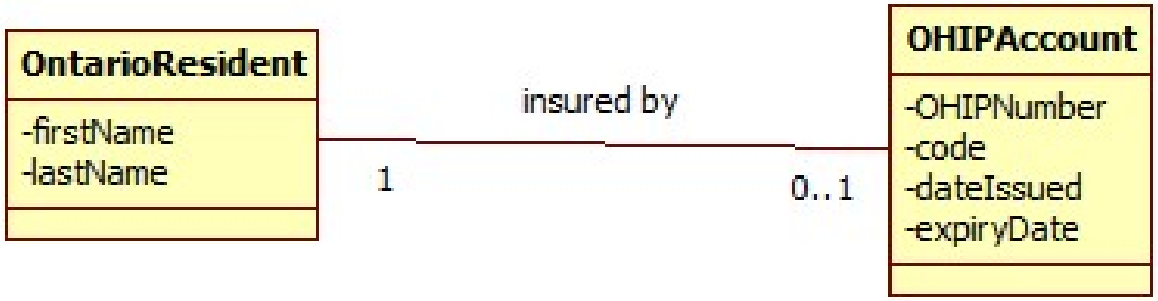
- We can only define and show associations successfully when we have enough information from our requirements or our business knowledge. For example:
  - Does the system need to remember that a specific clerk made a sale or is clerk not related to sale in the system?
  - Does the system need to remember the model of a bicycle or simply the manufacturer?
- Class associations typically indicate relationships that need to be remembered.

Drawing Associations



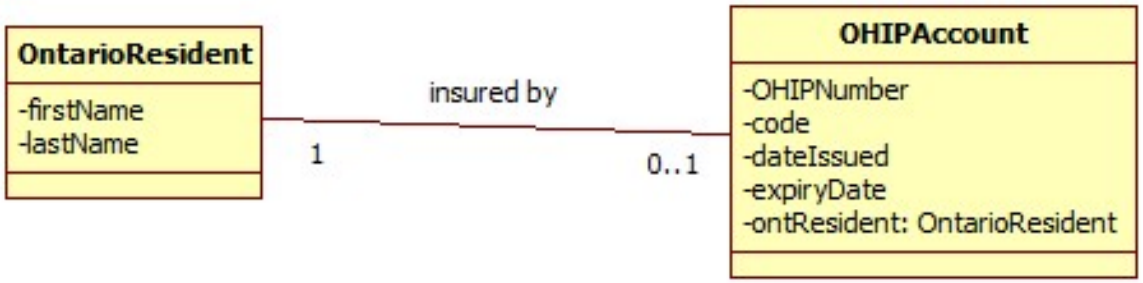
- Each Ontario resident is insured by **one or no** OHIP account. If a resident has an account then that resident has only one.
- Each OHIP account belongs to exactly **one** resident. No two residents share the same OHIP account.
- This means each OntarioResident object will be related to **one or zero** OHIPAccount objects.
- The name “insured by” tells us what the relationship is. To name associations it is best to use a verb or verb phrase.

Associations Denoting Reference Attributes



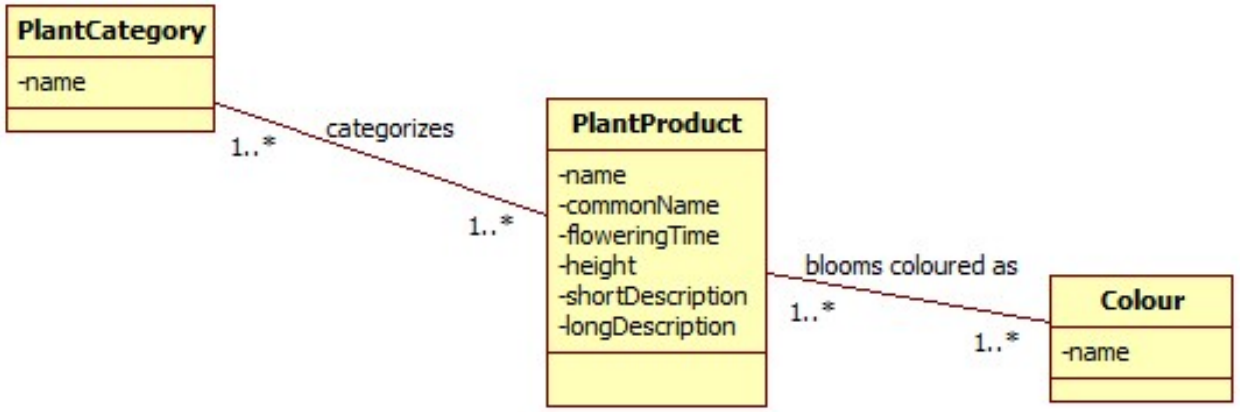
- Consider this scenario: in a doctor's office, the receptionist scans the patient's OHIP card; the patient's name appears on the screen.
- If we assume the OHIP card is attached to an OHIP Account then we know that the account refers to one Ontario Resident – we already see this in the association, but a reference attribute defines how we access this information.

Associations Denoting Reference Attributes



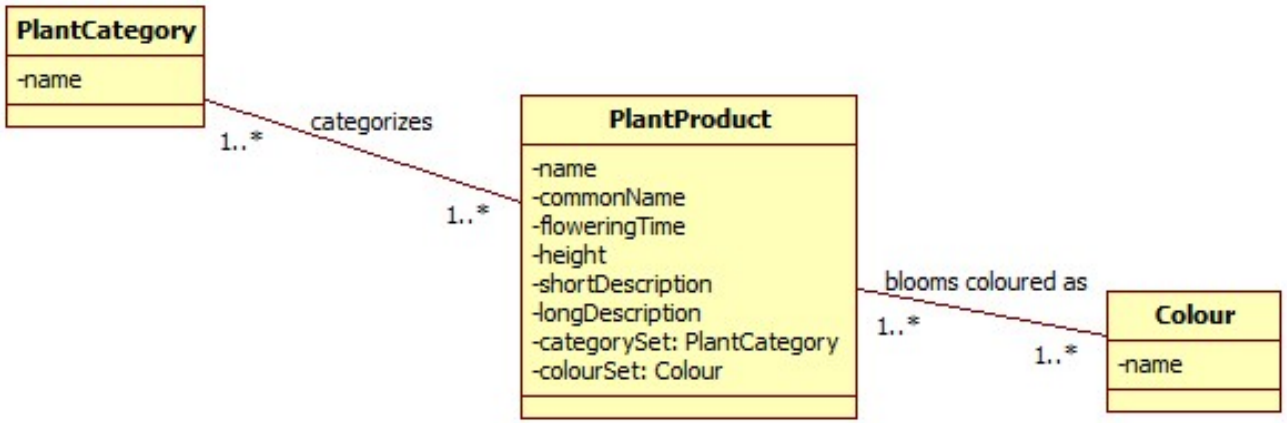
- The attribute “ontResident” is defined as an instance of the class OntarioResident.
  - In UML we can see this in the attribute properties—the type is defined as OntarioResident
- In the model above we see that we can access the attributes (and operations) of OntarioResident through ontResident. The reference attribute defines how we navigate this association.

Reference attributes as sets



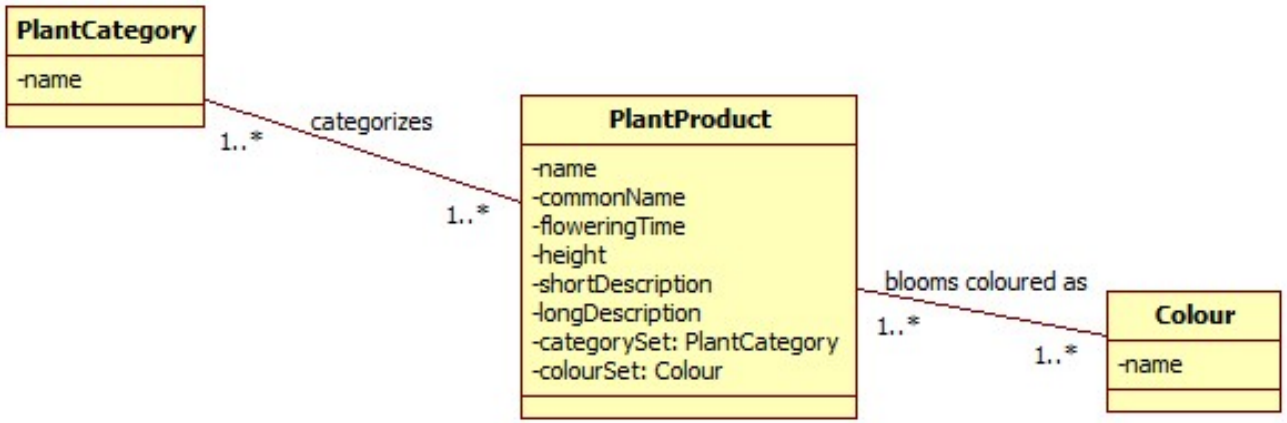
- Consider this “display plant product” scenario: The user selects a plant, the system displays all plant data including all categories that the plant is part of and all the colours of the blooms.

Reference Attributes as sets



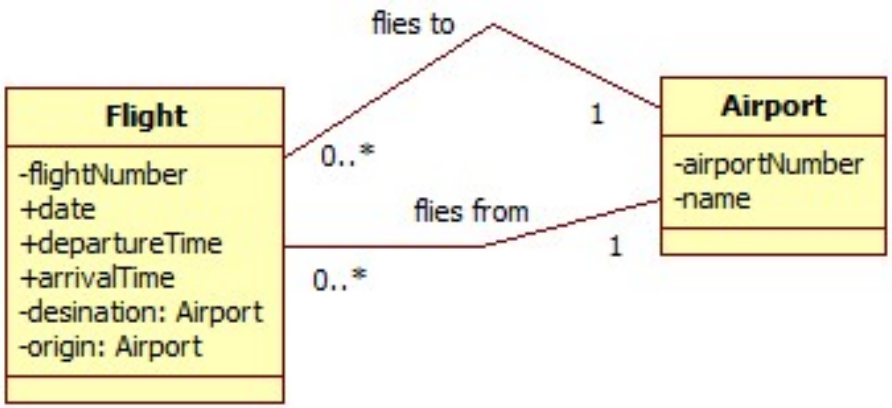
- We want to use PlantCategory and Colour as reference attributes to show how we navigate the associations but we have more than one of each.
- To denote this multiple reference attribute relationship we use the term SET:
  - categorySet:PlantCategory
  - colourSet:Colour

Reference Attributes as sets



- We now have a way of accessing category name for each category that the plant is part of and colour name for each colour of the plant blooms.

Reference Attributes to navigate Multiple Associations



- Airport is associated to Flight in two ways – one as origin and one as destination

In-Class Exercises

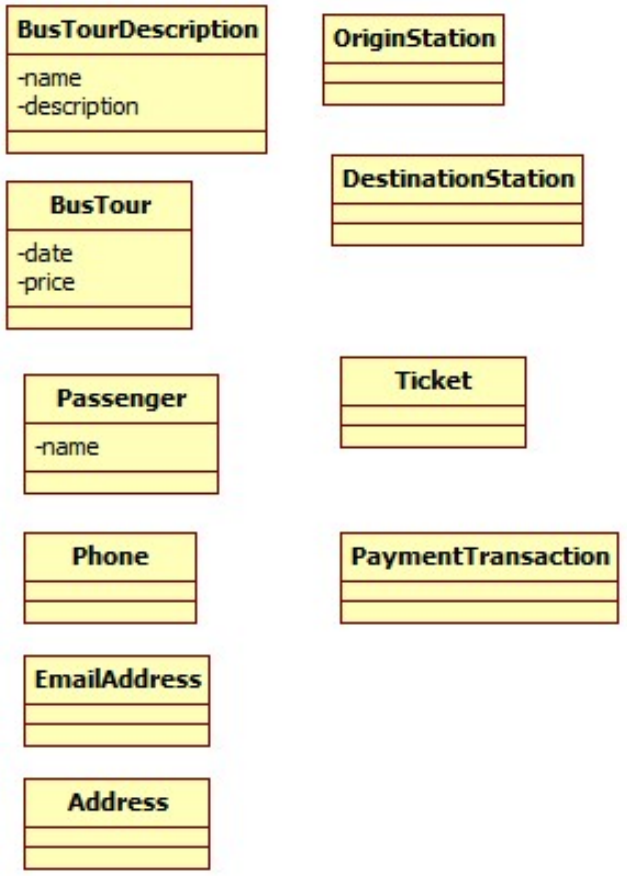
- In the following exercises (from your lab) do the following:
  - Add associations, association names and multiplicity
  - Add reference attributes that are required – singles and/or sets
  - Remove classes you consider extraneous
  - Add any classes you feel were missed

Exercise 1

Actor (Tournament Coordinator)	System
Enters date, name of tournament and maximum number of golfers.	Creates the tournament and displays an entry area for 10 golfers with spaces for name, contact information—email address and/or phone number, handicap.
Enters golfer information and requests to add.	Checks that maximum number of golfers has not been exceeded and adds the golfers to the tournament. Displays an entry area for more golfers.
Repeats step 2 until done	Displays and prints a list of golfers registered for the tournament.

Exercise 2

Actor (online customer)	System
Chooses a bus tour from a list of tours	Displays tour name, originating station and destination station and tour description. Also displays a list of dates on which the tour is offered. Displays tour price for each of the dates (summer tours are more expensive than spring and fall tours).
Selects one of the dates and requests to book a ticket for the tour	Displays an entry form for name, address, phone and email.
Enters name, address, phone, email.	Displays total price and all tour information for confirmation.
Confirms	Transfers to paypal and completes the payment transaction. Emails a ticket to the traveller.



Exercise 3

Actor (online customer)	System
	Displays a list of all shows (names and descriptions) for the theatre.
Chooses a show	Lists all performance dates for the show.
Chooses a performance date	Displays a list of theatre sections with prices for each section.
Chooses a section.	Searches for available seats based on the section and performance date selected. Displays available seats—shows row and seat number for each.
Chooses seats.	Displays total price and requests confirmation.
Confirms	Creates a purchase transaction and transfers control to the PAYPAL system. PAYPAL processes the request and returns control to the system being designed. The system displays ticket information for the purchased seats along with the total cost and a link to the pdf ticket file which the patron can print. The system also sends a confirmation email to the patron containing the link.