- Arrows in UML can have various purposes.
- Open arrows most often describe an inheritance relation.
- Closed arrows are used to describe associations.
- Square arrows define composition relations.
Stereotypes are also annotations put over arrows in class diagrams.

A collection of stereotypes is a profile.
We can use stereotype relations to better define the type of inheritance.

The three attributes are
- Disjoint or Overlapping
- Complete or Non-complete
- Dynamic or Static

These attributes are best understood using set theory.
Animal Example

Animal

{overlapping, complete}

Carnivore

Herbivore

Omnivore
Vehicle Example

Vehicle

\text{«abstract»}

\{\text{disjoint, complete}\}

power location

ExternalPower

InternalPower
Implementation

- Employees can get promotion and become managers.
- Employees can get demoted and become non-managers.
- How would you implement this change?
How do we implement the dynamic change of a Non-Manager becoming a Manager?

Option A: Create new object Manager. Copy fields. Destroy old object Non-Manager.

Option B: Flag if Manager or not. So we only need an object employee and it contains all the attributes for Non-Manager and Manager.
Associations describe which/how classes interact with each other.

- You can give an association a name (always a noun)
  - A full black arrow next to a name indicates the direction the diagram can be read.
- You can put roles at the end of connectors.
- You can also put numbers to indicate cardinality.
Cardinal Relations

- One-to-one
- Many-to-one or One-to-many
- Many-to-many

- Can you give me examples?
Sometimes, a class can be used to represent the association.
- Person object has a reference to dog object(s).

  ![Diagram](image)

- Dog object has a reference to person object.

  ![Diagram](image)

- Both objects have references to each other.

  ![Diagram](image)

- This relates to performance (at the cost of space).  

  ![Diagram](image)
Two special types of associations exists:

- Composition
- Aggregation
Which is which?

Scenario 1
- You can find books in a bookcase.

Scenario 2
- You can find shelves in a bookcase.
A composite object does not exist without its components.
- Composite does not outline its components.
- If we delete a composite object, we should cascade-delete its components.

For each components C, each C is a part of a single composite.
- Aggregate (whole) object can exist without aggregands (parts).
- Objects may be part of multiple aggregates.
- Typically, components are of same types.