UML Diagram Types

Dynamic Models
- activity diagrams
- statechart diagrams
- interaction diagrams
  - sequence diagrams
  - collaboration diagrams
- use case diagrams

Structural Models
- class diagrams
- *object diagrams*
- packages

Architectural Models
- component diagrams
- deployment diagrams

Structural Family: Instance

def'n: concrete manifestation of an abstraction to which a set of operations may be applied and which may have a state that stores the effects of the operation
- in most cases, an object is synonymous with an instance
- instances can occur for every building block of UML
- abstraction: ideal essence of a thing
- instance: concrete manifestation

Object Diagram

Contents of Object Diagram
- objects: instance of a class
- links: relationship between objects

Convention
- rectangle

Object Naming

- **name**
  - unique instance name:abstraction name
  - unique instance name :package name::abstraction name
  - name is underlined
- **anonymous:** no unique instance name (e.g., :Frame)
- **orphan:** no abstraction name (e.g., agent:)
- sometimes, just a simple name is appropriate if the class is obvious in the given context
- for simplicity, always name instance, class, and package (if package exists)

Operations and State

Convention/use of operations
- declared in the abstraction to which the instance is associated
- referenced by the instance name, followed by the operation name (e.g., instance *t* of class *Transaction* with operation *commit*() $\Rightarrow$ *t.commit*()

Convention/use of state (attributes)
- snapshot of current state of object
- attributes have values at given moment in time and space

Standard Elements

- instanceOf:

- instantiate:

- become:

- copy:

Modeling Concrete Objects

- Identify instances that are necessary and sufficient to visualize, specify, construct, or document the problem
- Render the objects, giving each a name
- Expose relevant stereotypes, tagged values, attributes from the abstraction if necessary for the model
- Render relationships in the Object Diagram

Hints and Tips

- Model instances associated with a specific abstraction
- Give each instance a unique name