UML Diagram Types

Behavioral 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

Class Diagram

def'n: set of classes and relationships
- most common type of diagram
- static design view
- foundation for other diagrams
- construction of executable systems is directly dependent upon class diagram

Class

def'n: set of objects that share same attributes, operations, relationships, and semantics

alt def'n: abstraction of thing that is part of vocabulary

Convention
- rectangle with several compartments
- simple (name) or path name (package:name)
- noun or noun phrase with 1st letter of each noun capitalized (TempSensor)

Hints and Tips

A well structured class

- provides a crisp abstraction of something drawn from vocabulary of problem domain
- embodies a small, well-defined set of responsibilities
- provides a clear separation of abstraction's specification and its implementation
- is understandable and simple yet extensible and adaptable

Responsibility

def'n: contract or obligation of class

- textual description of what class is/does
- each class should have 1 < resp <= 4

Convention

- free form text
- phrase, sentence, or short paragraph

Attribute

def'n: named property that describes a range of values that instances of a property may hold

- class may have any number of attributes

Convention

- noun or noun phrase
- capitalize 1st letter of each noun except first (loadBearing)

Operation

def'n: implementation of a service that can be requested from any object of the class to affect behavior

- class may have any number of op's

Convention

- verb or verb phrase
- capitalize 1st letter of each noun except first (op1())

Attributes and Operations

- Don't need to show all attributes and operations
- Only show those relevant to current view (at the proper level of abstraction)
- Empty compartments does not mean 0 att or ops, only that choose not to show

Hints and Tips

Well structured class diagram

- is focused on communicating one aspect of a system's static design view
- contains only elements that are essential to understanding that aspect
- provides detail consistent with its level of abstraction
- is not so minimalist that it misinforms reader about important semantics

Drawing Class Diagrams

- Try the sticky-pad approach
- Early on, responsibilities are more important than attributes and operations
  - don't overthink upfront
- Early on, don't try to completely populate attributes and operations
  - other diagrams help to add details
- Start with 'simple' relationships
  - over time add complexity

Summary

- Class diagrams are composed of classes and relationships
- Classes
  - responsibilities
  - attributes
  - operations
- Relationships
  - association
  - generalization
  - aggregation
  - composition
  - dependency