Mobile Application Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit

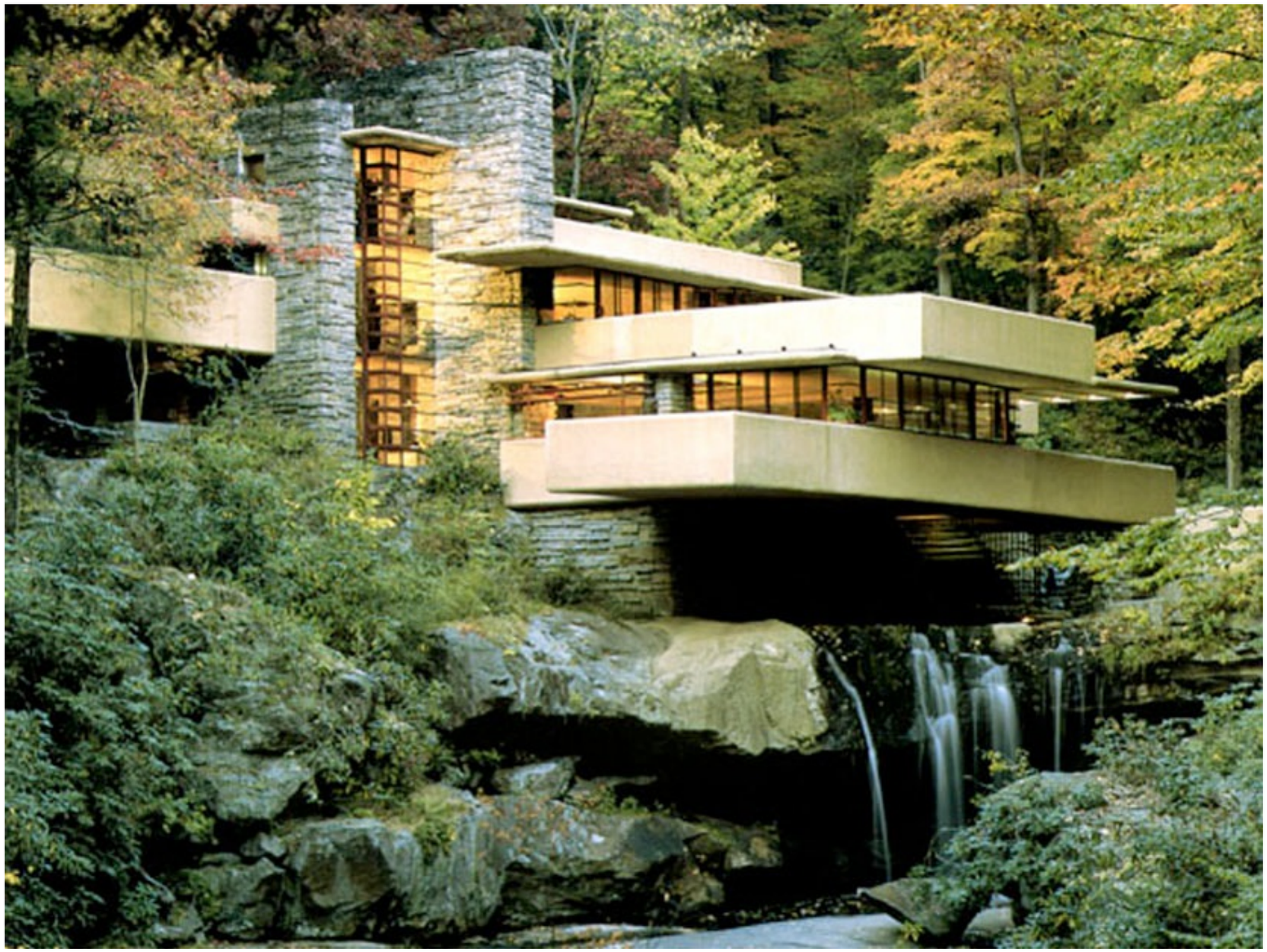
Introduction to UML

Why develop a UML model?

- Provide structure for problem solving
- Experiment to explore multiple solutions
- Furnish abstractions to manage complexity
- Decrease development costs
- Manage the risk of mistakes

The Challenge

The Vision



Why do we model graphically?

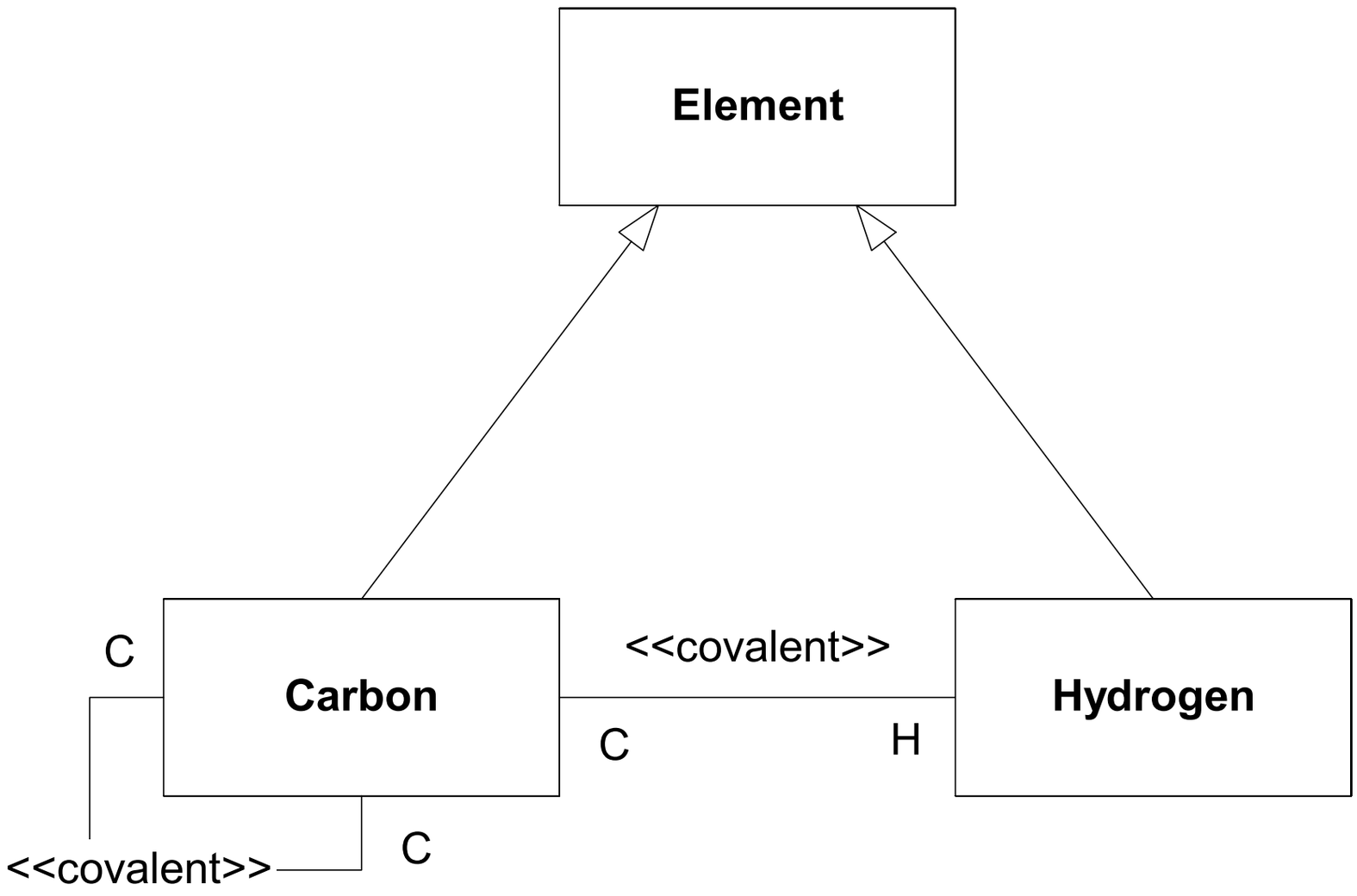
- Graphics reveal data.
    - Edward Tufte
      The Visual Display of Quantitative Information, 1983

- 1 bitmap = 1 megaword.
    - Anonymous visual modeler

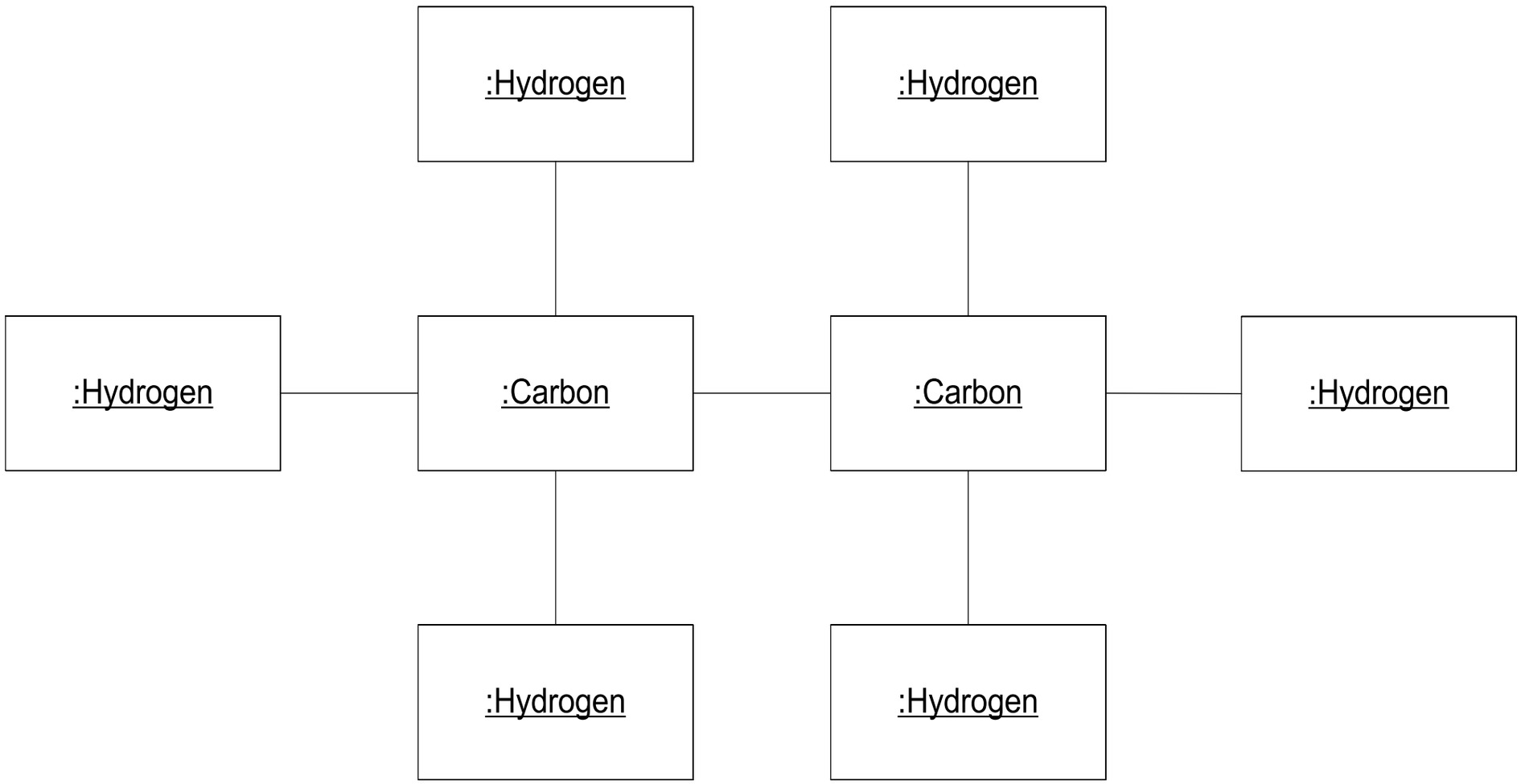
Building Blocks of UML

- The basic building blocks of UML are:
  - model elements (classes, interfaces, components, use cases, etc.)
  - relationships (associations, generalization, dependencies, etc.)
  - diagrams (class diagrams, use case diagrams, interaction diagrams, etc.)
- Simple building blocks are used to create large, complex structures
  - eg elements, bonds and molecules in chemistry
  - eg components, connectors and circuit boards in hardware

Example : Classifier View



Example: Instance View



UML Modelling Process

- Use Case
- Structural
- Behavioural
- Architectural

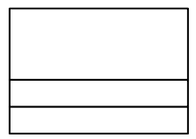
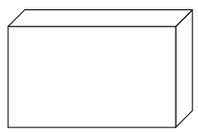
Structural Modelling

- Core concepts

- Diagram Types

Structural Modeling Core Elements

- a view of an system that emphasizes the structure of the objects, including their classifiers, relationships, attributes and operations.

Construct	Description	Syntax
class	a description of a set of objects that share the same attributes, operations, methods, relationships and semantics.	
interface	a named set of operations that characterize the behavior of an element.	
component	a modular, replaceable and significant part of a system that packages implementation and exposes a set of interfaces.	
node	a run-time physical object that represents a computational resource.	

Structural Modelling: Core Relationships

Construct	Description	Syntax
association	a relationship between two or more classifiers that involves connections among their instances.	
aggregation	A special form of association that specifies a whole-part relationship between the aggregate (whole) and the component part.	
generalization	a taxonomic relationship between a more general and a more specific element.	
dependency	a relationship between two modeling elements, in which a change to one modeling element (the independent element) will affect the other modeling element (the dependent element).	

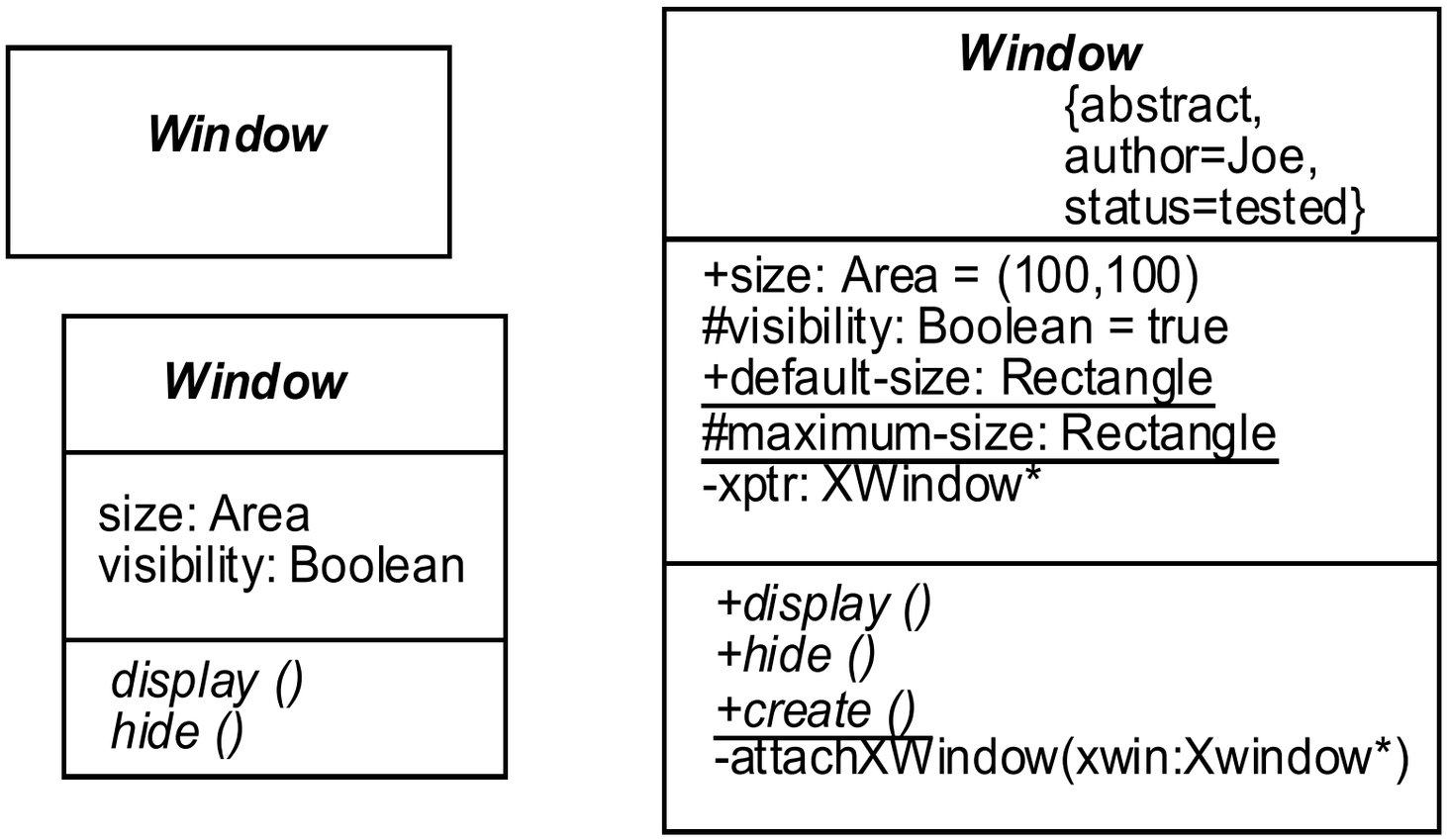
Structural Diagram Tour

- Show the static structure of the model
  - the entities that exist (e.g., classes, interfaces, components, nodes)
  - internal structure
  - relationship to other entities
- Do not show
  - temporal information
- Kinds
  - static structural diagrams
    - class diagram
    - object diagram
  - implementation diagrams
    - component diagram
  - deployment diagram

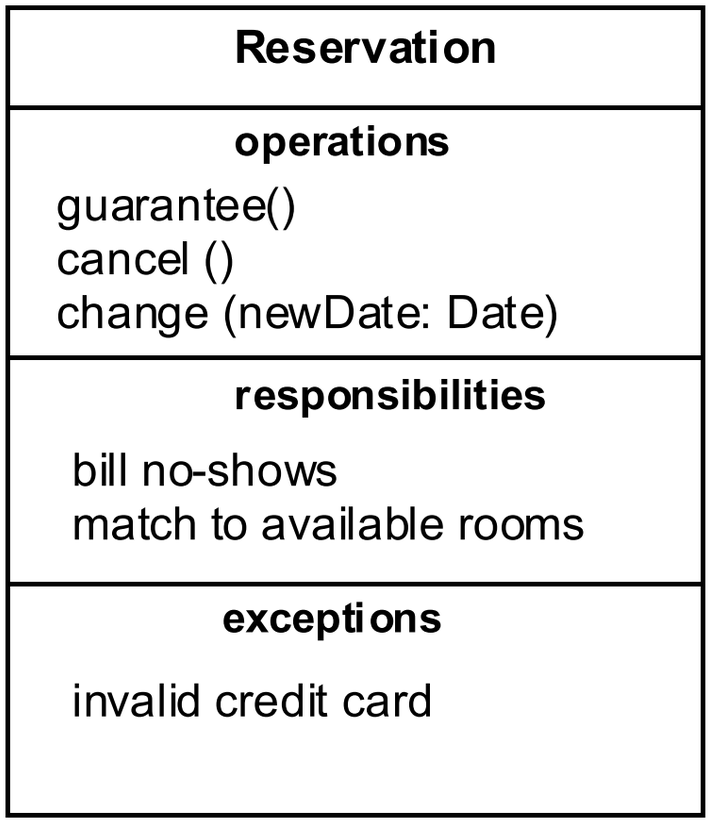
Static Structural Diagram Examples

- Shows a graph of classifier elements connected by static relationships.
- kinds
  - class diagram: classifier view
  - object diagram: instance view

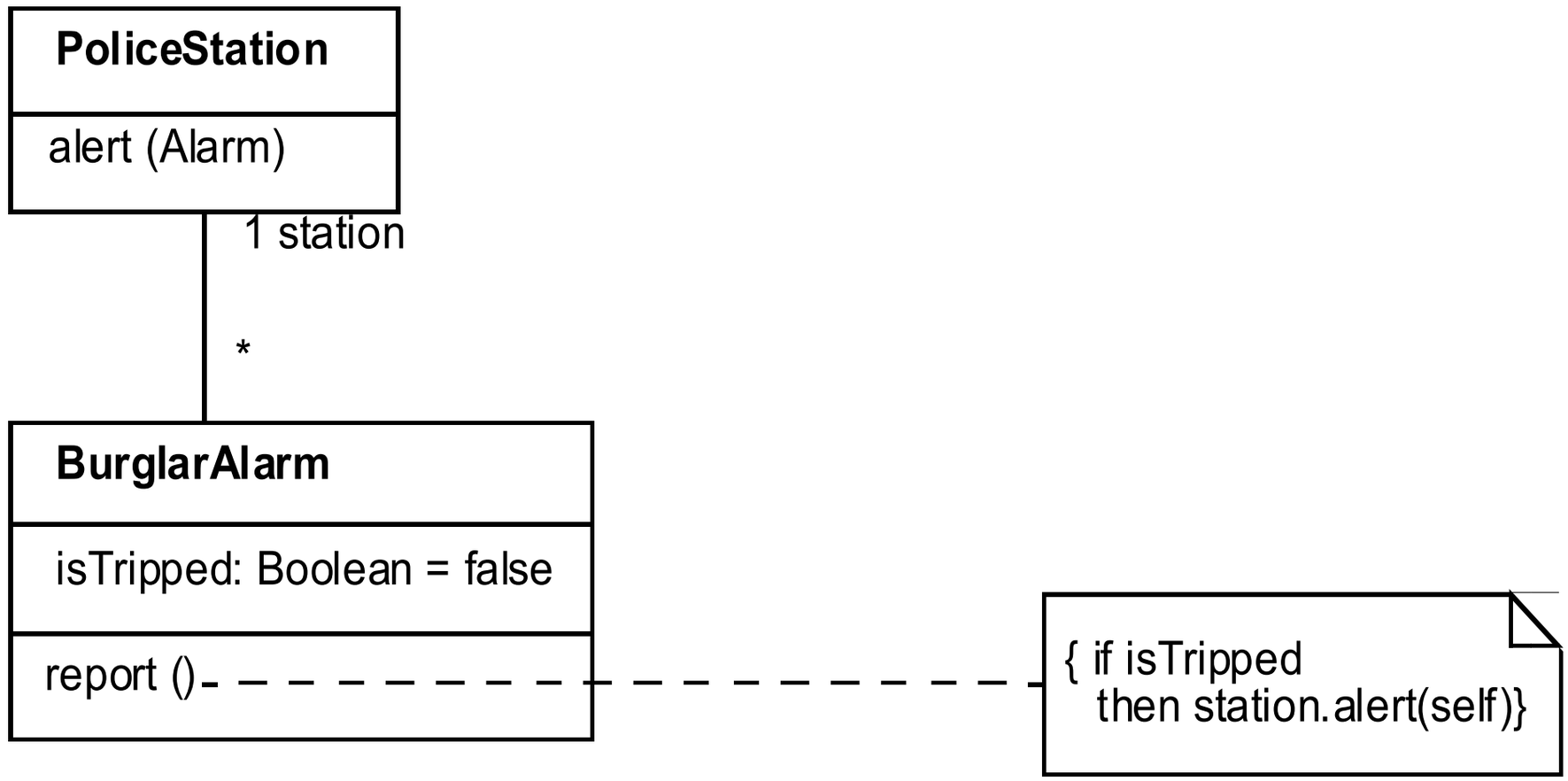
Classes


Window
Window
size: Area
visibility: Boolean
display ()
hide ()
Window
{abstract,
author=Joe,
status=tested}
+size: Area = (100,100)
#visibility: Boolean = true
+default-size: Rectangle
#maximum-size: Rectangle
-xptr: XWindow*
+display ()
+hide ()
+create ()
-attachXWindow(xwin:Xwindow*)

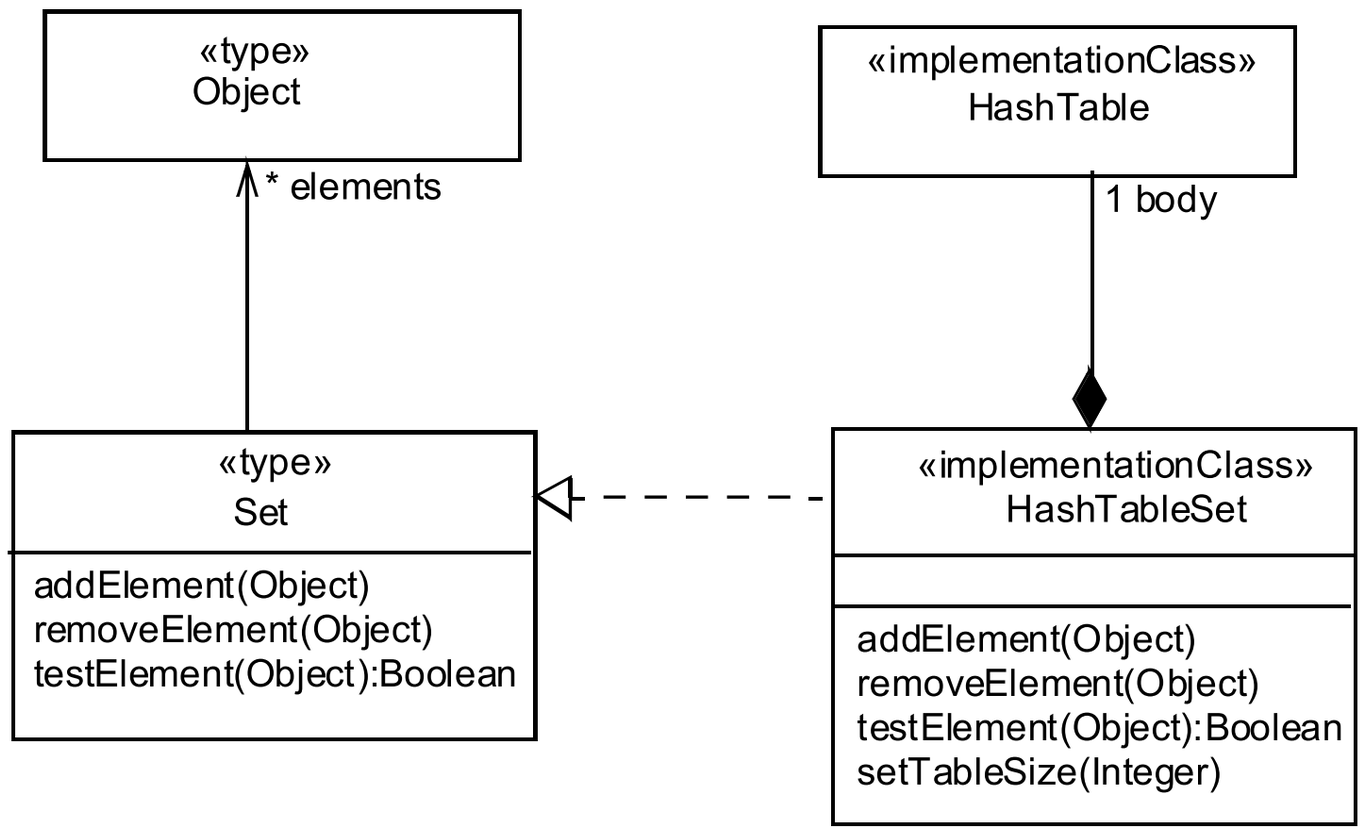
Classes: Compartments with Names



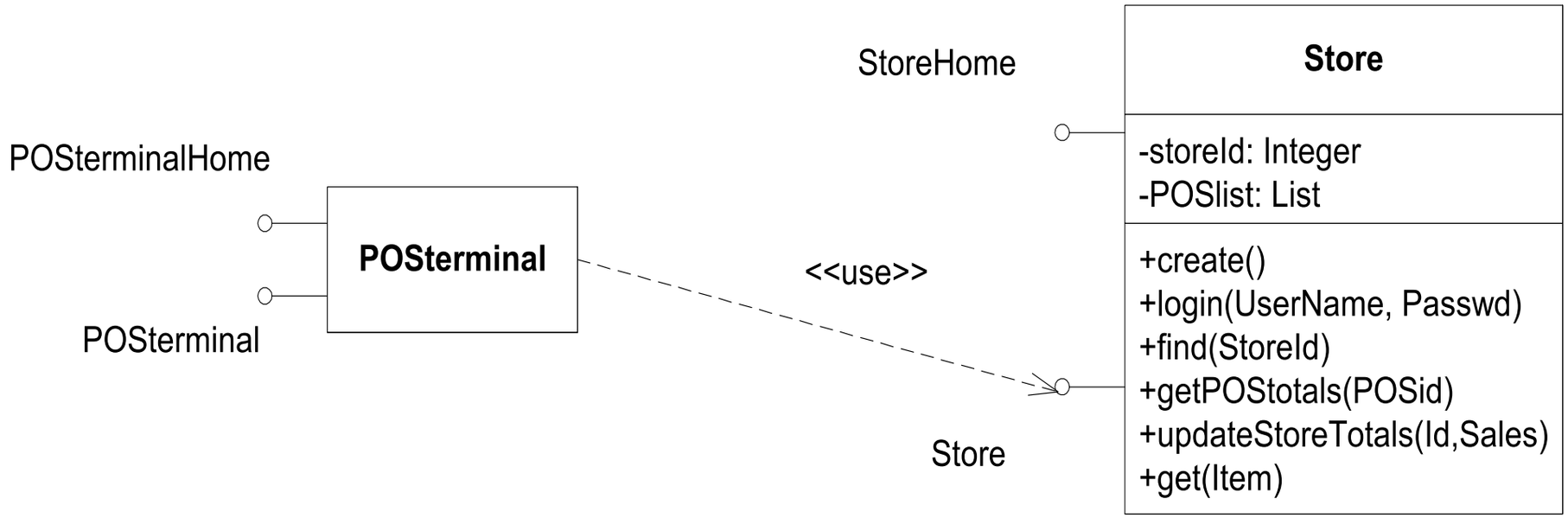
Classes: method body



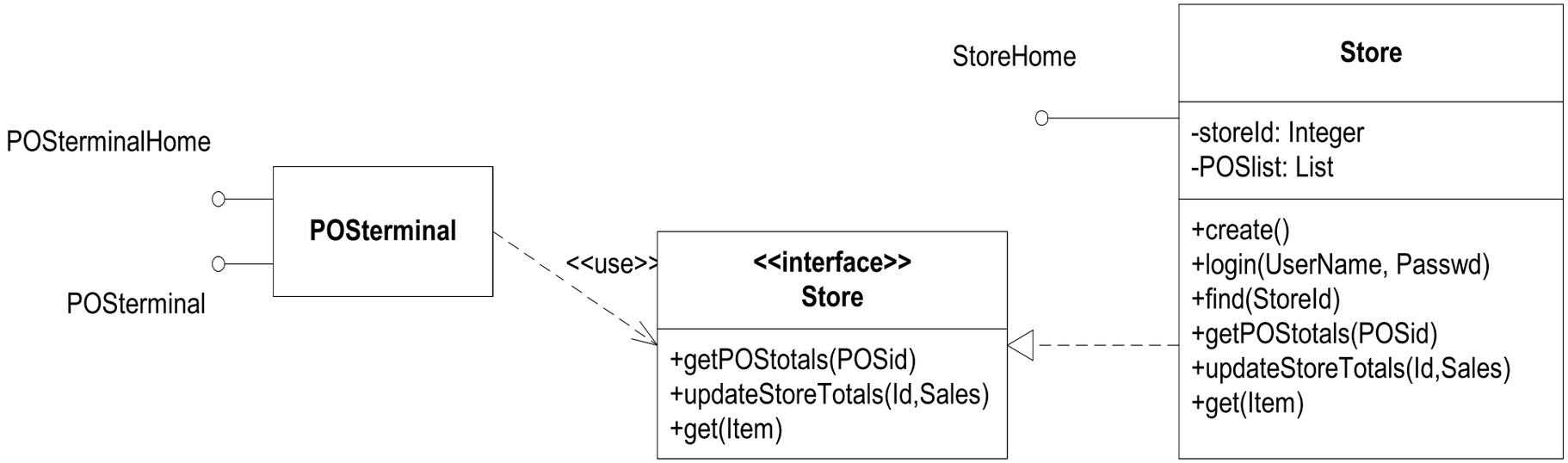
Types & Implementation Classes



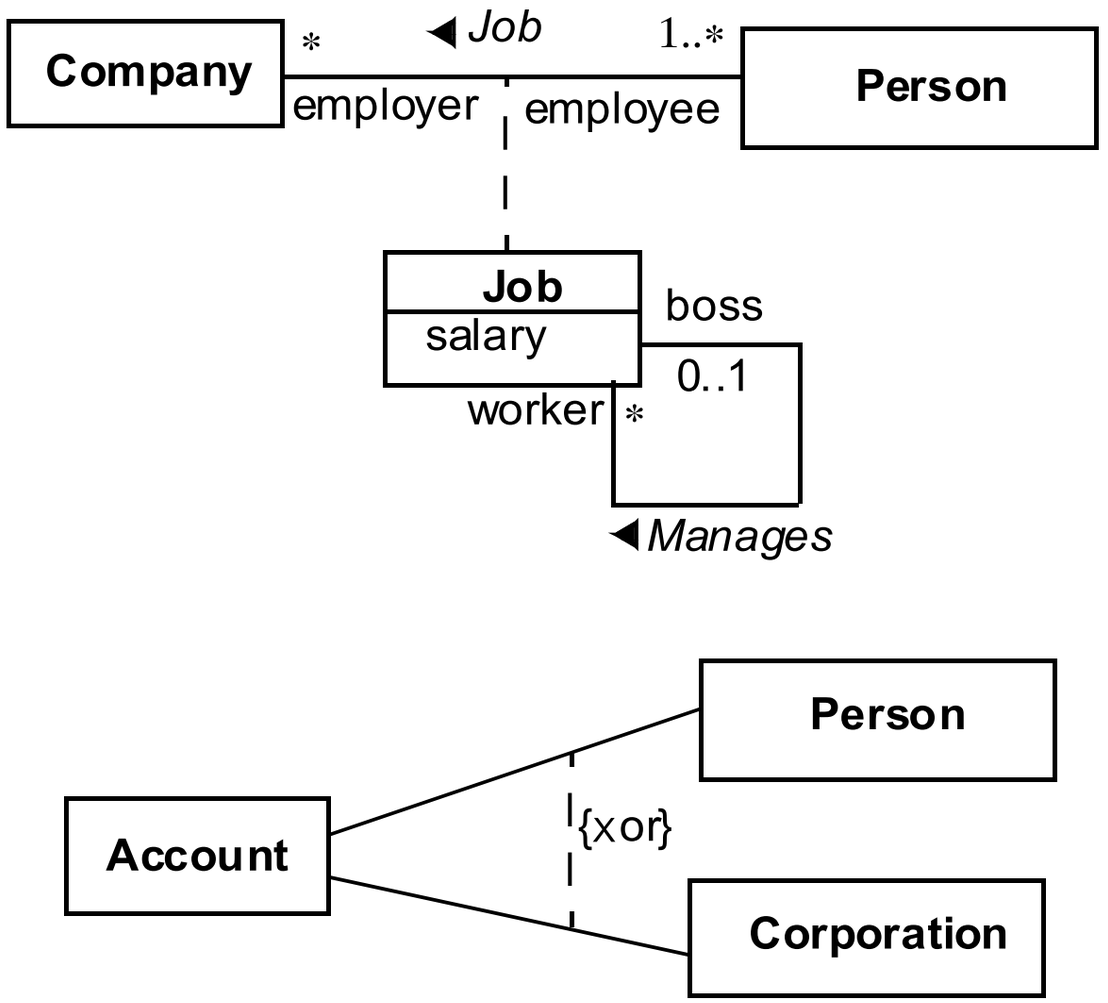
Interfaces: Shorthand Notation



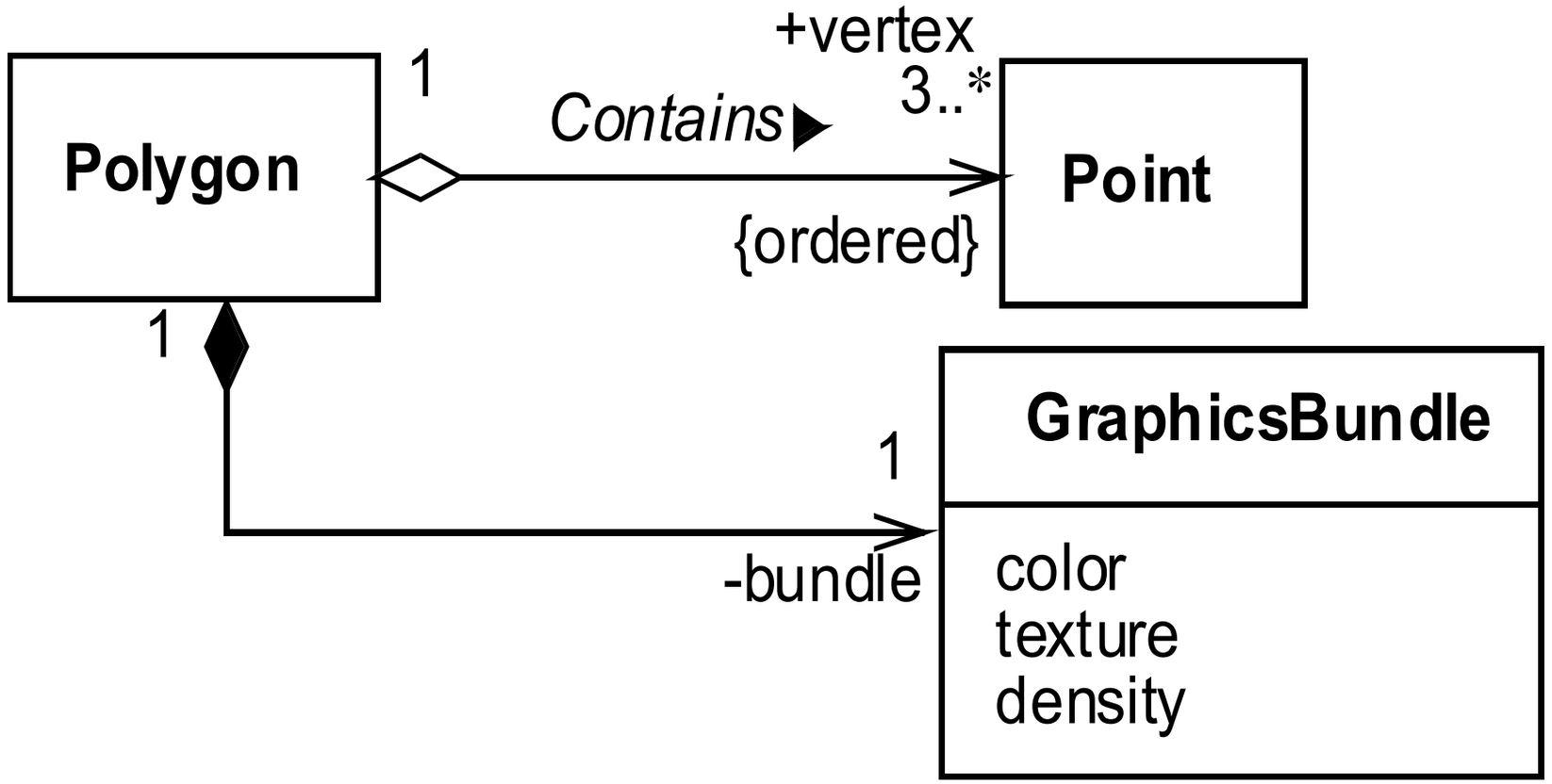
Interfaces: Longhand Notation



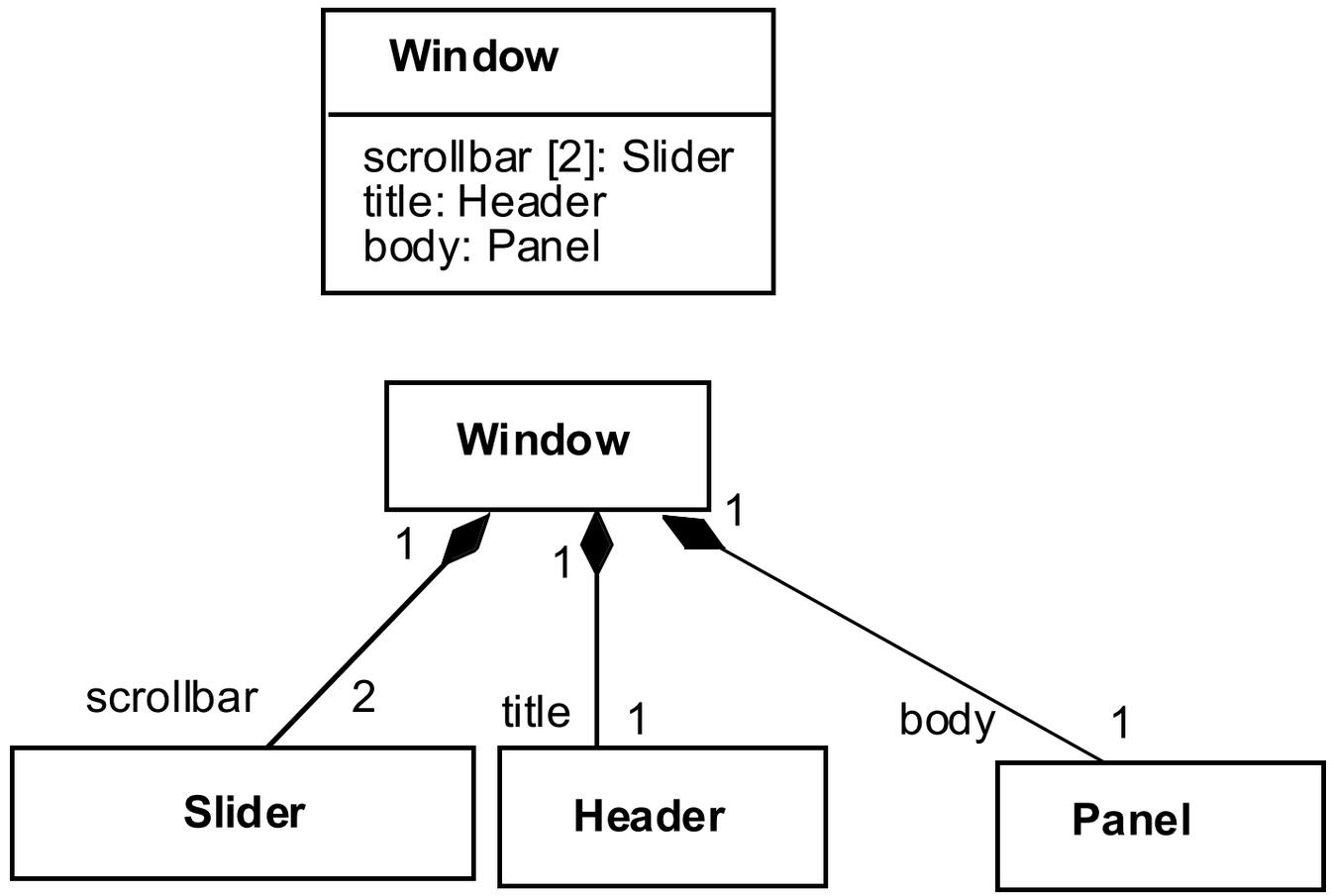
Associations



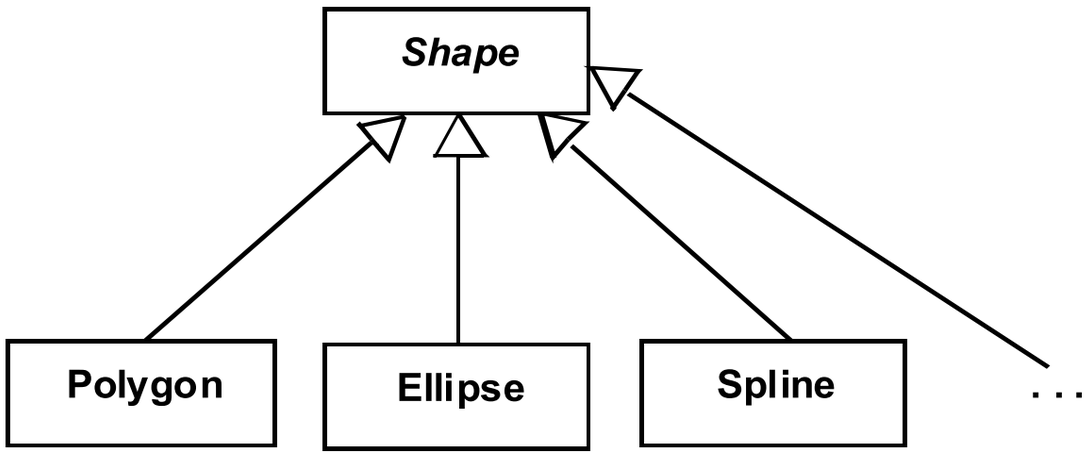
Association Ends



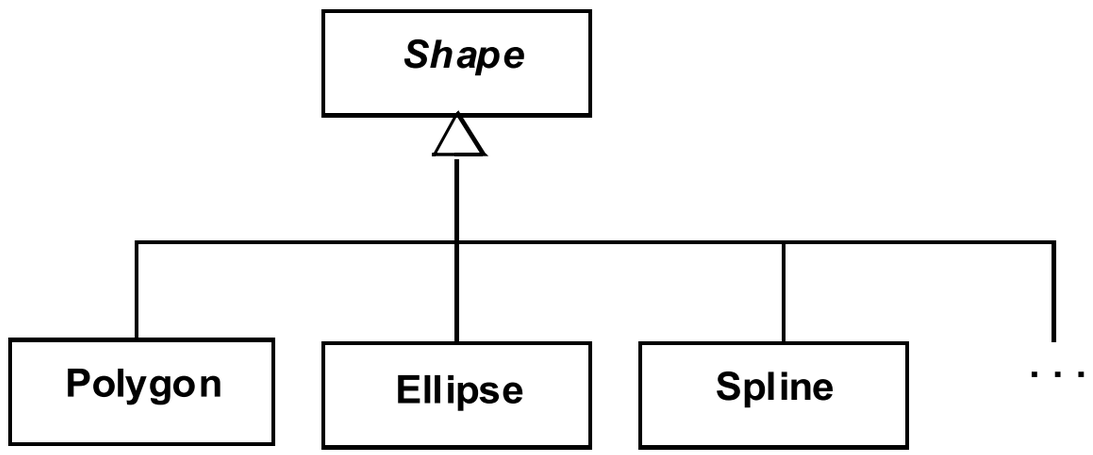
Composition



Generalization (Inheritance)

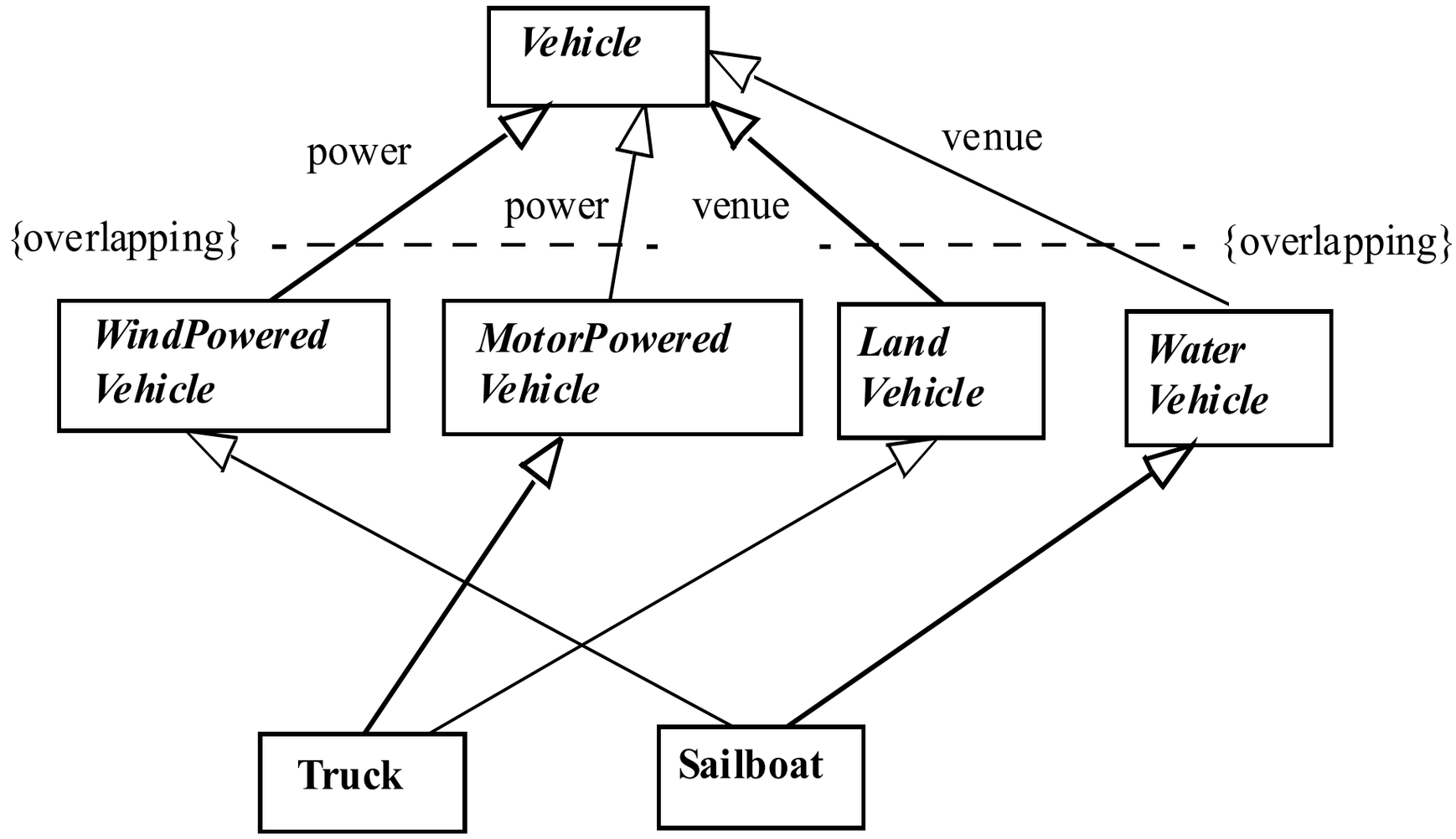


Separate Target Style

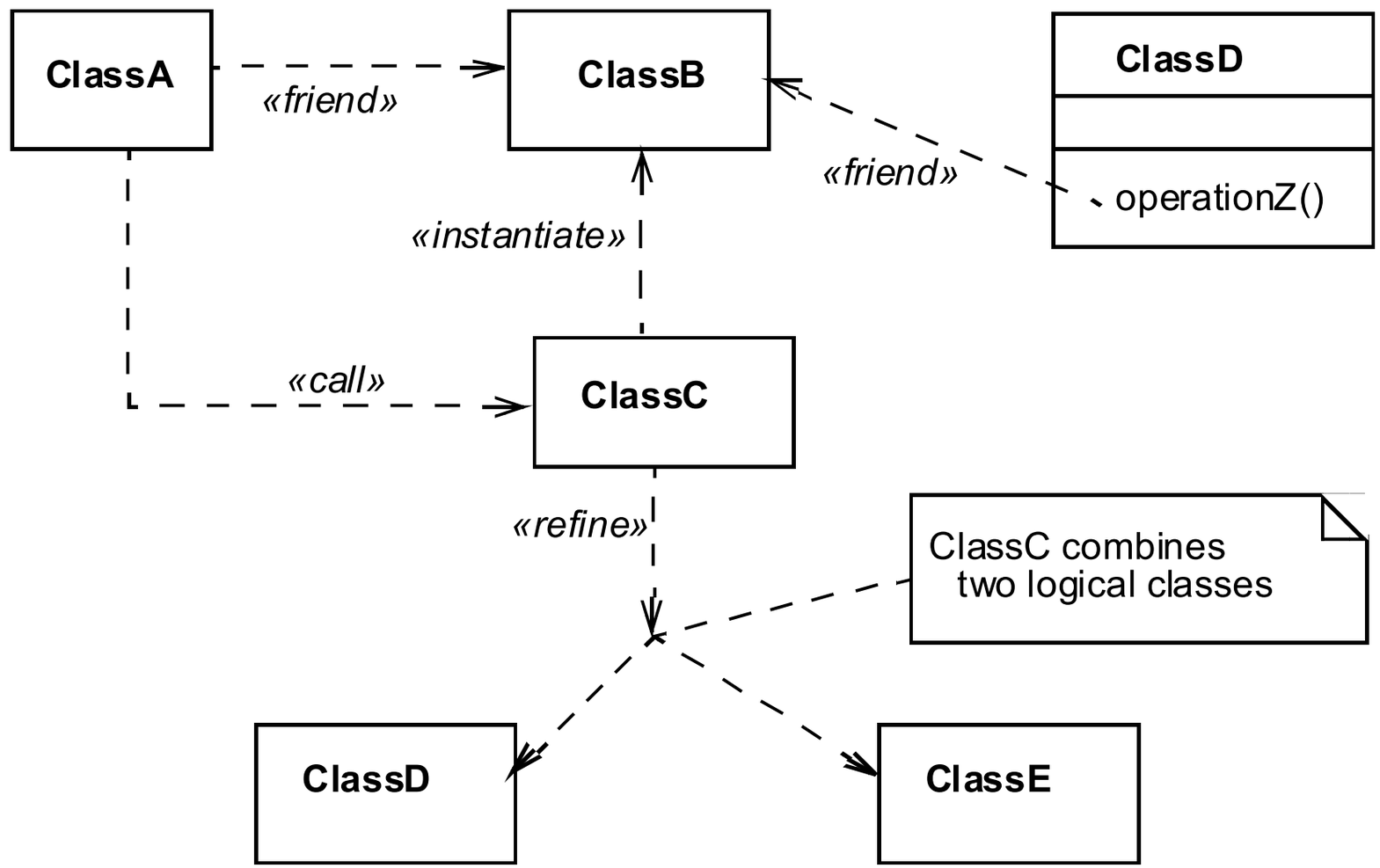


Shared Target Style

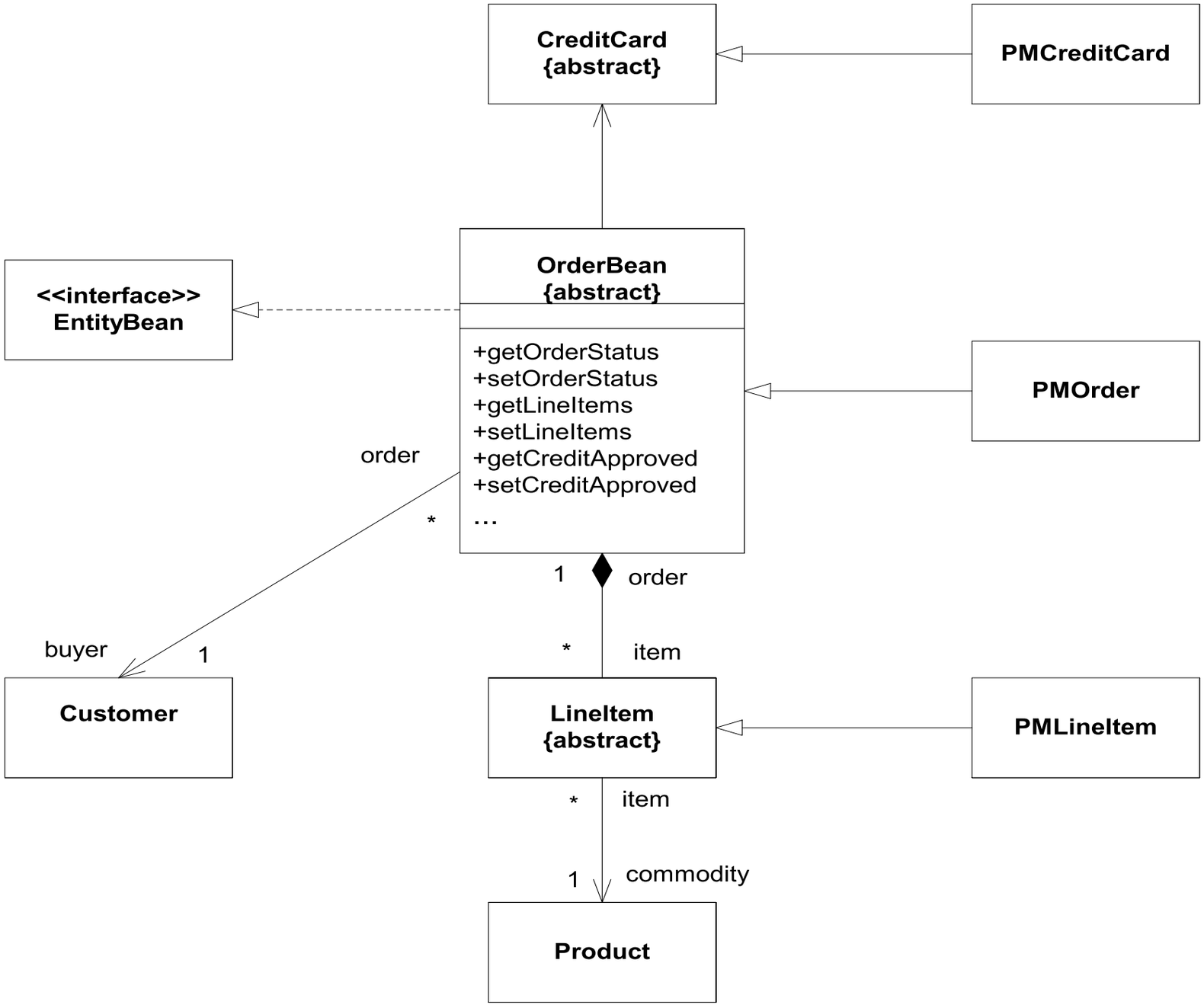
Generalization (Inheritance)



Dependencies



Class Diagram Example



Some of these slides were adapted from a presentation by Cris Kobryn

Co-Chair UML Revision Task Force

+ the Visual Paradigm Online Help

http://www.visual-paradigm.com/product/vpuml/provides/umlmodeling.jsp

Waterford Institute *of* Technology

INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit