

5.3 UML Odds and Ends

- Just a few things we didn't have time to cover before
- Additional elements of class diagrams
- Extension mechanisms
- Packages
- Notes
- Deployment diagrams

Aggregation



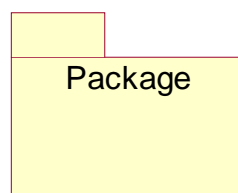
- Aggregation is a kind of association
- "is part of" relation
 - a book is part of a library catalog
 - a library catalog has any number of books
- No really clear meaning
 - may make a diagram more readable
 - maps to the same code as any other association

Composition



- Composition is a kind of aggregation
- But with a well-defined meaning!
- Example
 - class Rectangle
 - `private Point[] p = new Point[4];`
 - //the elements of p are never given out
 - `Rectangle(int x, int y, int height, int width) {`
 - `p[0] = new Point(x, y);`
 - ... }
- Meaning: each part belongs to exactly one composite
 - When the composite is destroyed, the part is destroyed
 - Parts cannot be shared among composites

Packages



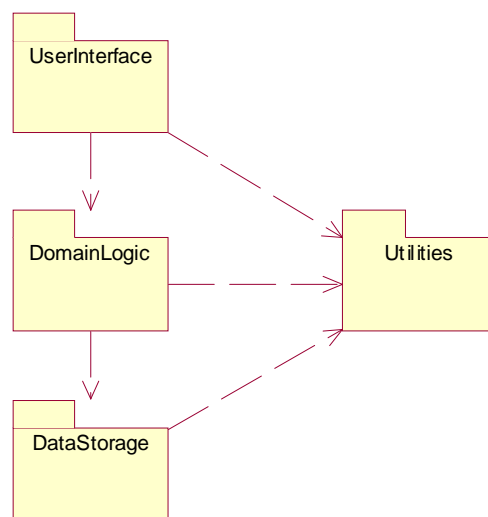
- A grouping mechanism
- A package can contain any kind of UML elements
- Two uses:
 - to hide part of a diagram
 - to show the relationship between large numbers of elements

Dependency

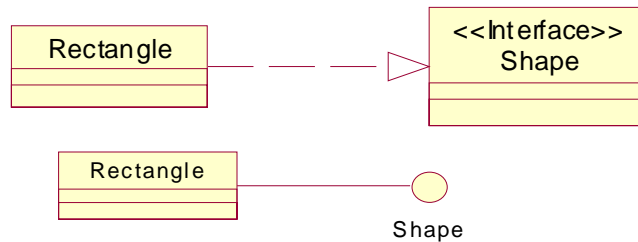


- A depends on B
- Any kind of “uses” relation between classes that is not an association or a generalization
- Examples:
 - A instantiates B (but does not keep the reference)
 - A has operations that have objects of type B as parameters
- Generally:
 - Class A cannot be used without class B
- Can also be used with packages

Example: Three-Tier Architecture

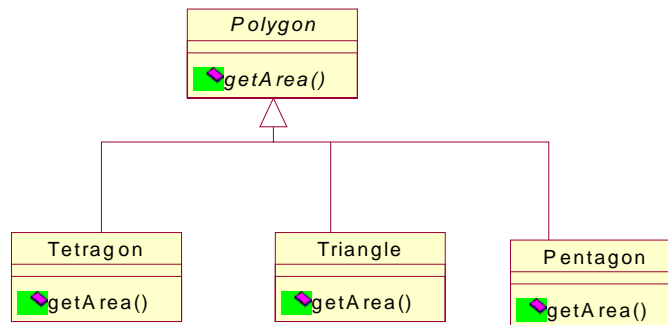


Interfaces



- Class Rectangle implements interface Shape
- Top: full representation
- Bottom: elided representation
 - no operations of the interface can be shown
- The relation between class and interface is a realization
- Interfaces are a kind of class in UML

Abstract Classes



- Abstract classes and operations
 - have names in italics
- By the way...
 - this is an other way to show generalizations
 - several arrows made into one

UML Extension Mechanisms

- Stereotypes
 - allow user-defined UML elements
 - for example:
 - «interface»
 - «user interface class»
 - those thingies are called guillemets
 - can also be represented through their own icons
 - such as a circle for interfaces
- Profiles
 - set of stereotypes for a purpose
 - for example: USDP profile
- Constraints
 - we already discussed those
 - {abstract}