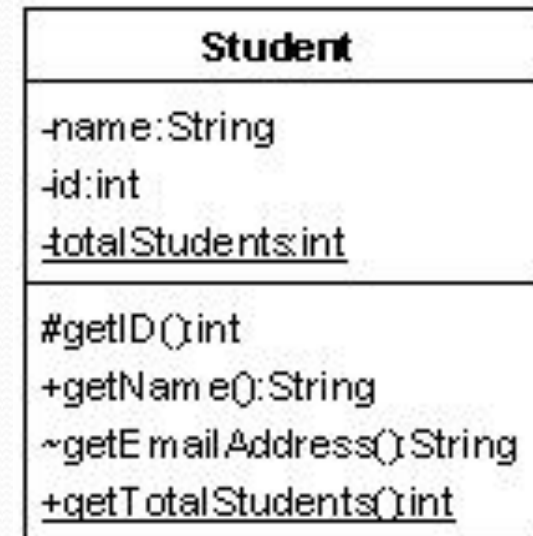
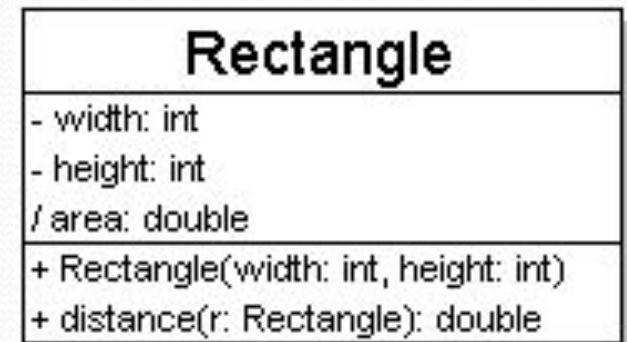


UML class diagrams

- What is a UML class diagram?
 - **UML class diagram:** a picture of the classes in an OO system, their fields and methods, and connections between the classes that interact or inherit from each other

Diagram of one class

- class name in top of box
- attributes
- operations / methods



Class attributes

- attributes (fields, instance variables)
 - visibility: + public
protected
- private
/ derived
 - underline static attributes

Rectangle
- width: int - height: int / area: double
+ Rectangle(width: int, height: int) + distance(r: Rectangle): double

Student
-name:String -id:int <u>-totalStudents:int</u>
#getID():int +getName():String ~getEmailAdress():String <u>+getTotalStudents():int</u>

Class operations / methods

- operations / methods
 - *visibility name (parameters) : return_type*
 - visibility: + public
 # protected
 - private
 ~ package (default)
 - underline static methods
 - parameter types listed as (name: type)
 - omit *return_type* on constructors and when return type is void
 - method example:
+ distance(p1: Point, p2: Point): double

Rectangle
- width: int - height: int / area: double
+ Rectangle(width: int, height: int) + distance(r: Rectangle): double

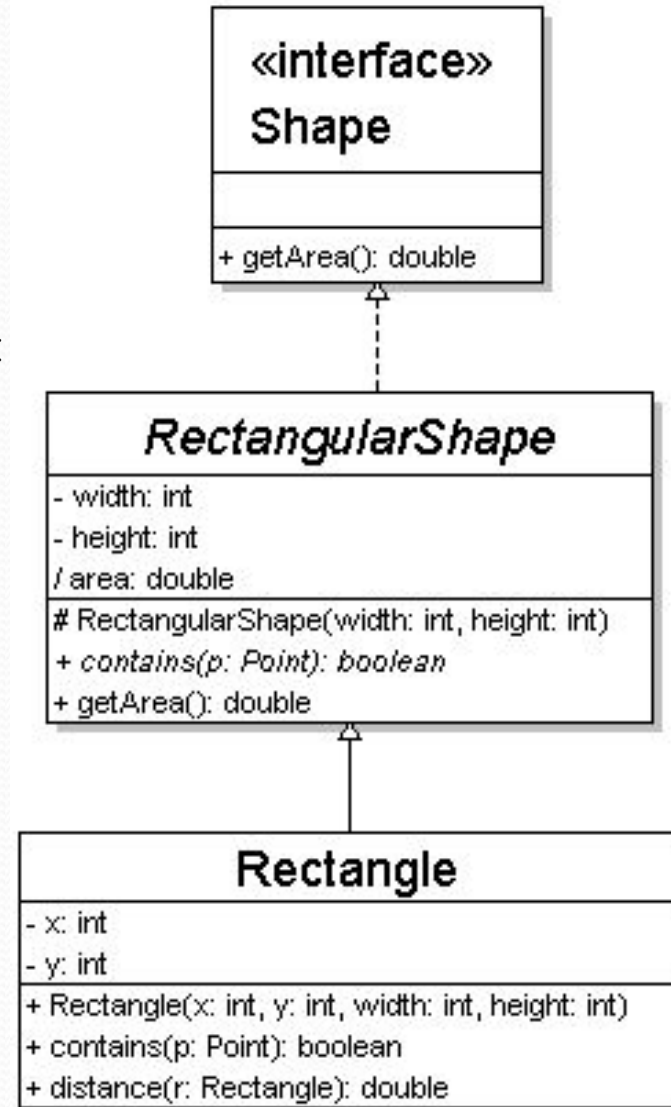
Student
-name:String -id:int <u>-totalStudents:int</u>
#getID()int +getName():String ~getEmailAdress():String <u>+getTotalStudents()int</u>

Relationships btwn. classes

- **generalization**: an inheritance relationship
 - inheritance between classes
 - interface implementation
- **association**: a usage relationship
 - dependency
 - aggregation
 - composition

Generalization relationships

- generalization (inheritance) relationships
 - hierarchies drawn top-down with arrows pointing upward to parent
 - line/arrow styles differ, based on whether parent is a(n):
 - class:
solid line, black arrow
 - abstract class:
solid line, white arrow
 - interface:
dashed line, white arrow



Associational relationships

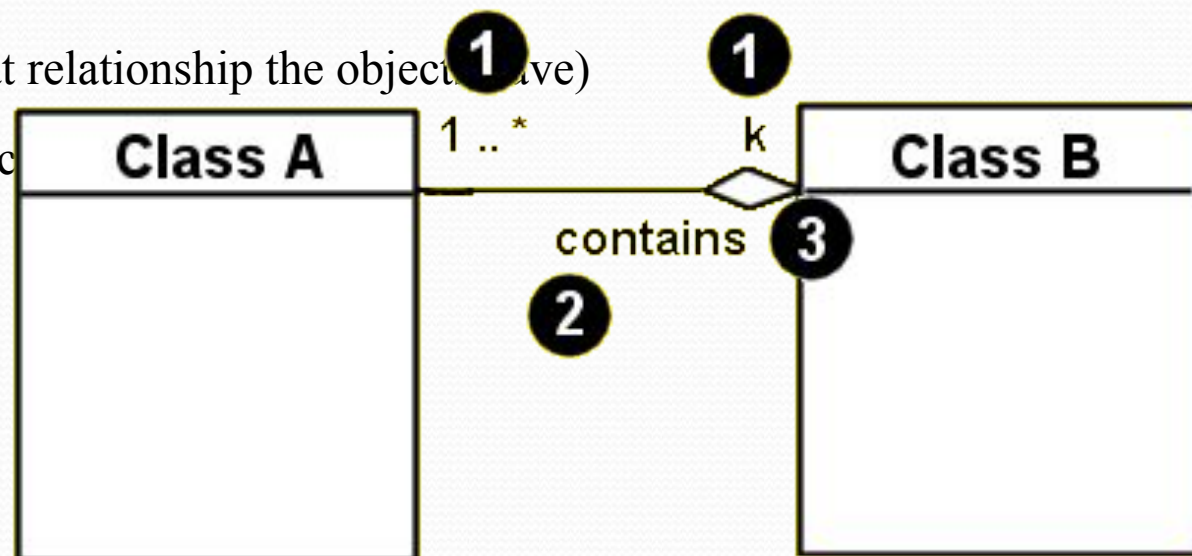
- associational (usage) relationships

1. multiplicity (how many are used)

- * \Rightarrow 0, 1, or more
- 1 \Rightarrow 1 exactly
- 2..4 \Rightarrow between 2 and 4, inclusive
- 3..* \Rightarrow 3 or more

2. name (what relationship the objects have)

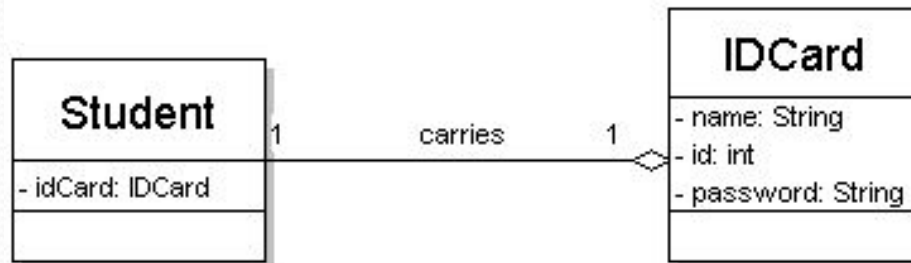
3. navigability (direction)



Multiplicity of associations

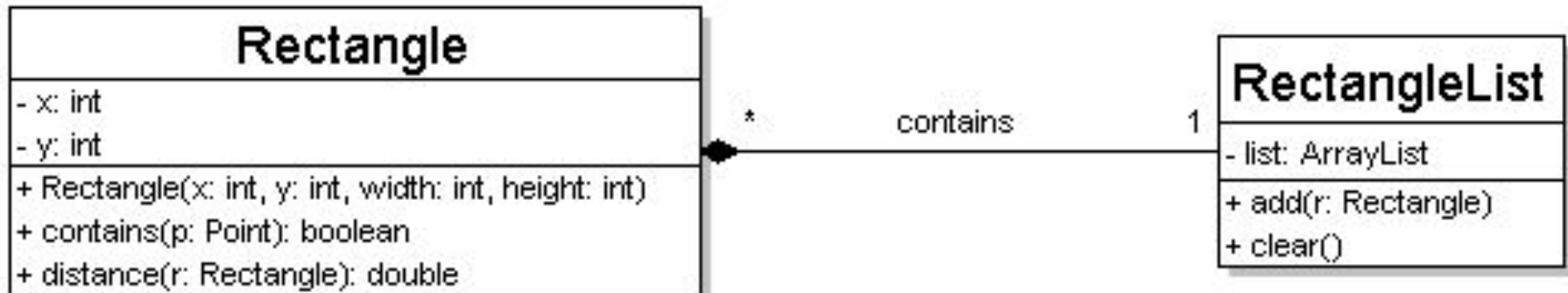
one-to-one

- each student must carry exactly one ID card



one-to-many

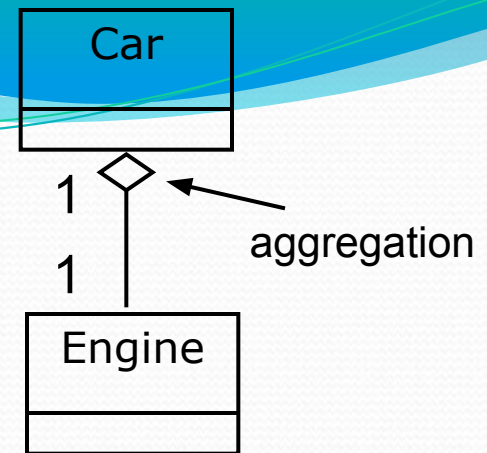
- one rectangle list can contain many rectangles



Association types

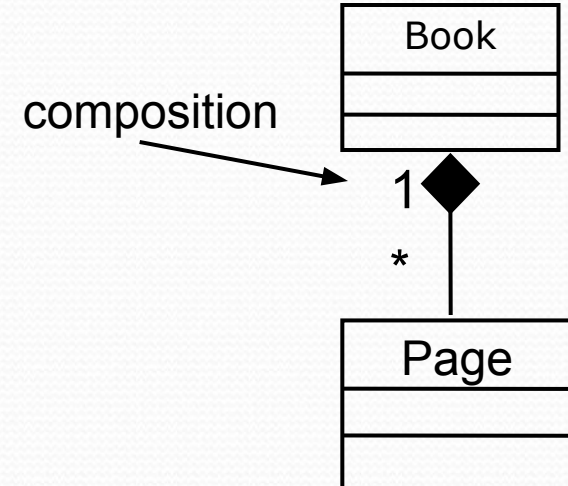
- **aggregation:** "is part of"

- symbolized by a clear white diamond



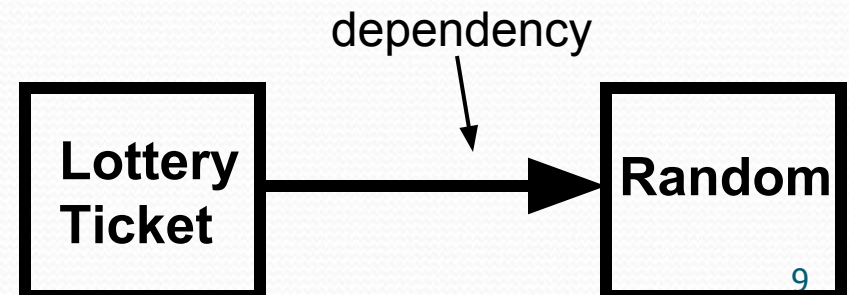
- **composition:** "is entirely made of"

- stronger version of aggregation
- the parts live and die with the whole
- symbolized by a black diamond



- **dependency:** "uses temporarily"

- symbolized by dotted line
- often is an implementation detail, not an intrinsic part of that object's state



Class diagram example

