

Thursday 10/2/14

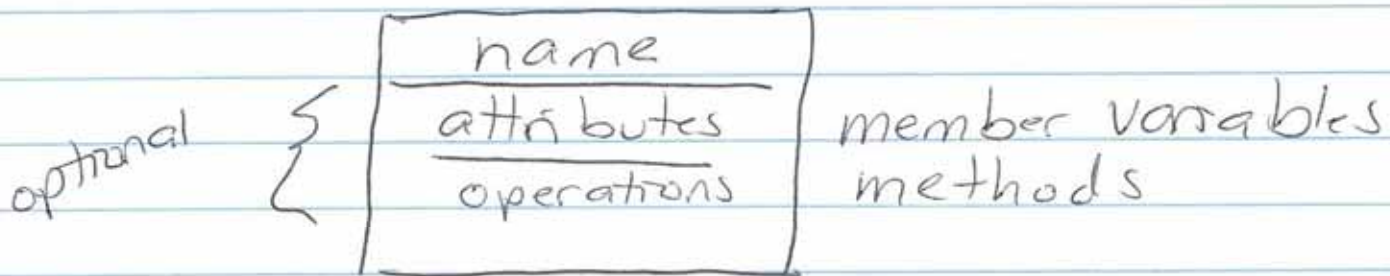
don't <sup>new</sup> show <sup>cssm</sup> yet

page 1

4156

I have covered the 1st few chapters of the Head First Design Patterns book in class, need to learn rest on your own - many online resources

for the examples on the board, I've been using class diagrams w/o ever explaining them - so will do so now



variables

name : type = initial value  
optional

methods

name ( params ) : return type  
↑  
name : type, ~

visibility

+ public  
# protected  
- private

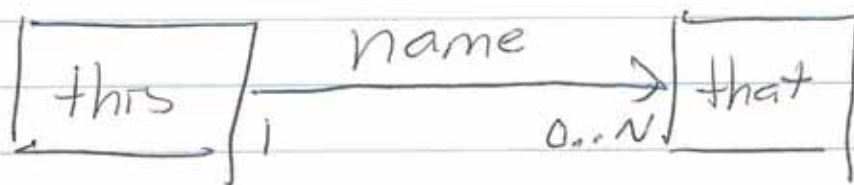
scope - classifier (static)  
or instance

Thursday 10/2/14

4156

class diagram often depicts static relationships among multiple classes

association - instances of one class must "know" about or communicate with an instance of another class to do its work



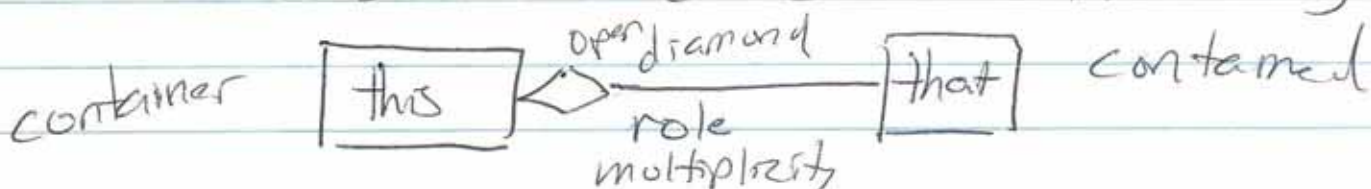
0..\* 1..N multiplicity

arrow shows direction if uni-directional  
no arrow if bi-directional

ends of line might also show role names, visibility, etc.

aggregation - HAS-A relationship, part-whole, collection or container

does not have strong life-cycle dependency (deleting container does not delete contained)



Thursday 10/2/14

4156

composition - Stronger HAS-A  
 usually Strong like cycle dependency  
 altho sometimes child might  
 be explicitly removed before  
 destroying parent

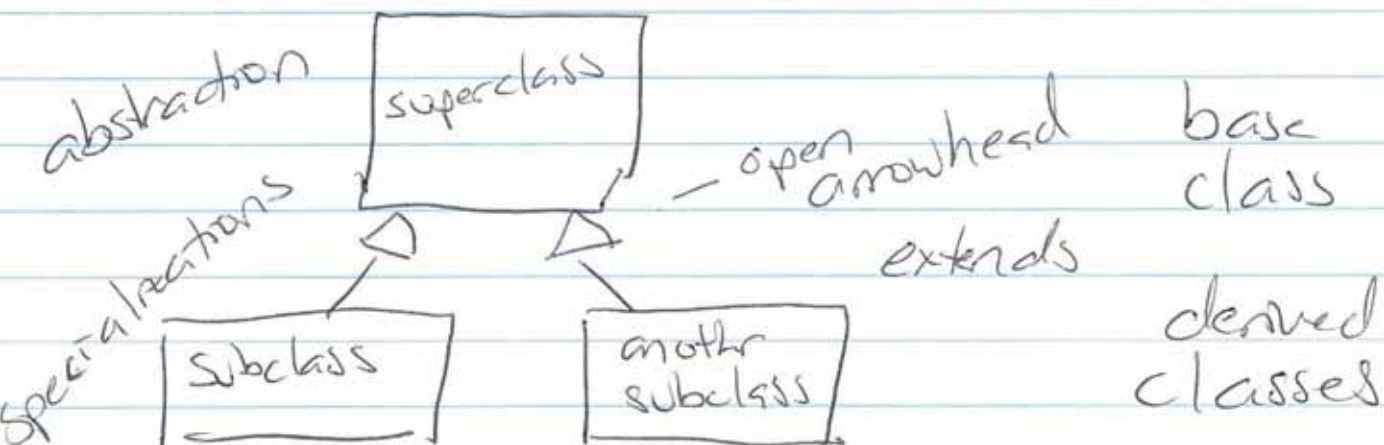


catalog vs. physical containment  
 (software, database) ← design pattern use  
 often "catalog"

association & aggregation/composition  
 are instance level relationships

also class level relationships

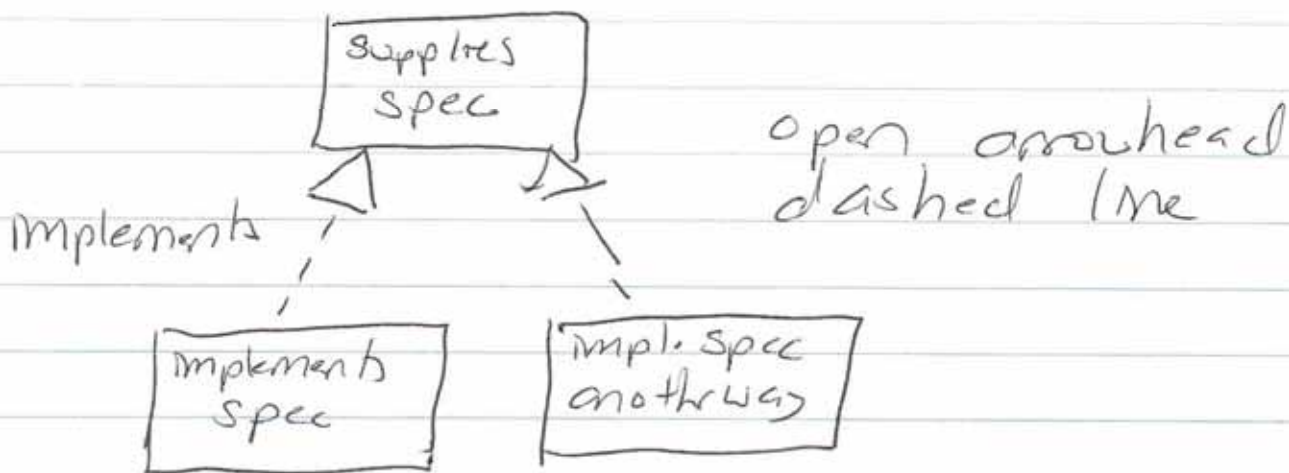
generalization - IS-A or inheritance





Thursday 10/2/14

realization - implements an interface or behavior specified



the reason I'm giving you all this detail now is because you will need for the next assignment

- show assignment now

besides class diagrams, you'll need sequence diagrams

dynamic - shows how instances of classes work together at runtime

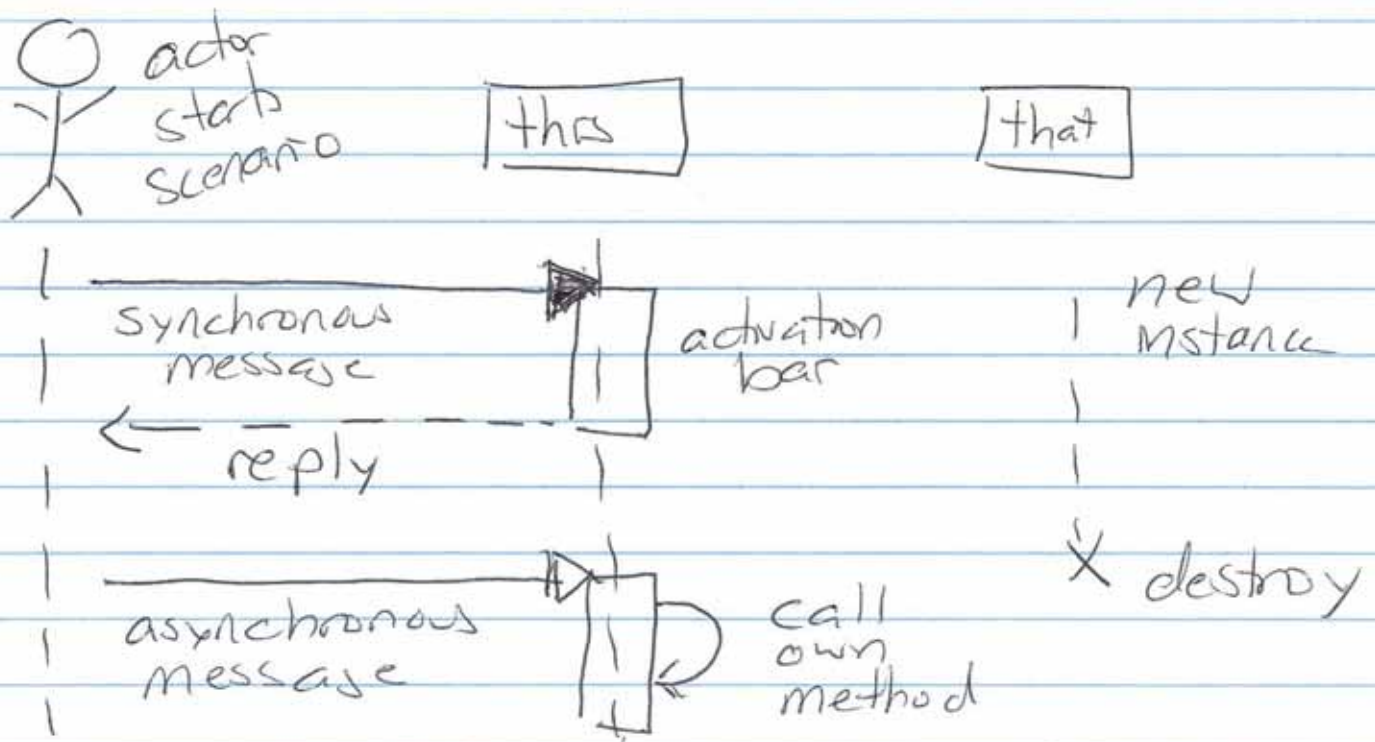
visual way to show order of events that happen, such as invoking methods

- interaction
- often correspond to use cases

Thursday 10/2/14

page 5

4156



lifeline  
(time progress)

message of form

[condition] name (params) : return type

all optional except name

class name of form

instance of class  
name a name

or role name

~~class~~

~~instance of class~~

Thursday 10/2/14

Where do the classes & messages  
come from?

first rough design -

classes - find nouns in  
User stories/use cases

messages - find verbs

subject/object of verb should be  
actors, whole system,  
or classes

focus on essentials, not inner workings

look for opportunities to use  
design patterns

animated walkthroughs with  
CRC cards

index cards with

class responsibilities collaborations