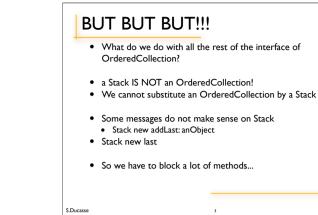


The Problem

- There is not a clean simple relationship between Stack and OrderedCollection
- Stack interface is not an extension or subset of OrderedCollection interface
- Compare with CountingStack a subclass of Stack



Another Appro	pach Revol
By defining the cla OrderedCollection	
Object subclass: St iv: elements	ack
Stack>>push: anEle elements addFir Stack>>pop	rst: anElement
element isEmpt	y ifFalse: [^ element removeFirst]

•	Subclassing Dictionary is a subclass of Set
•	Semaphore is a subclass of LinkedList
•	No relationship between the interfaces of the classes
•	Subclass reuses code from superclass, but has a different specification. It cannot be used everywhere its superclass is used. Usually overrides a lot of code.
•	ShouldNotImplement use is a bad smell

Specification Inheritance

Subtyping

S.Ducasse

S.Ducasse

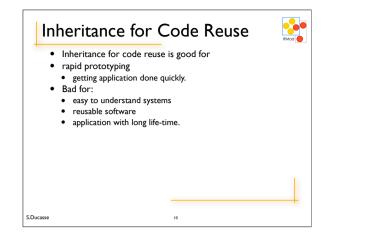
- Reuse of specification
- Fractions.
- A program that works with Collections will work with Arrays.
- A class is an abstract data type (Data + operations to manipulate it)

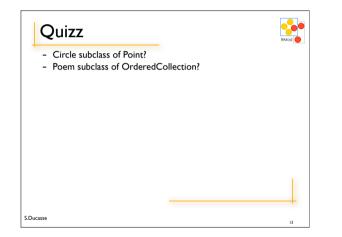
8

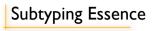
••••

- CountingStack is an extension

• A program that works with Numbers will work with







You reuse specification

S.Ducasse

- You should be able to substitute an instance by one of its subclasses (more or less)
- There is a relationship between the interfaces of the class and its superclass



How to Choose?

Favor subtyping

S.Ducasse

- When you are in a hurry, do what seems easiest.
- Clean up later, make sure classes use "is-a-subtype" relationship, not just "is-implemented-like".
- Is-a-subtype is a design decision, the compiler only enforces is-implemented-like!!!

RMod