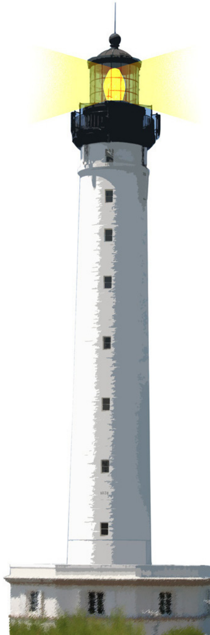
Learning Object-Oriented
Programming and Design with TDD



About visibility

S. Ducasse

http://stephane.ducasse.free.fr

http://www.pharo.org

Extra

Objectives

- Private/protected... in different languages
- And the cost of all of it

Protected in Pharo/CPP/CSharp

- Protected attributes are not accessible from outside the class and its subclasses.
- Subclasses can access instance variables and invoke/override methods.

Visibility in Java

Two levels

- Top level (e.g., class): public or package-private (no explicit modifier)
- Member level: public, private, protected, package-private (no explicit modifier)

Top level

- A public class is visible to all the classes everywhere.
- A package-private class is only to its own package

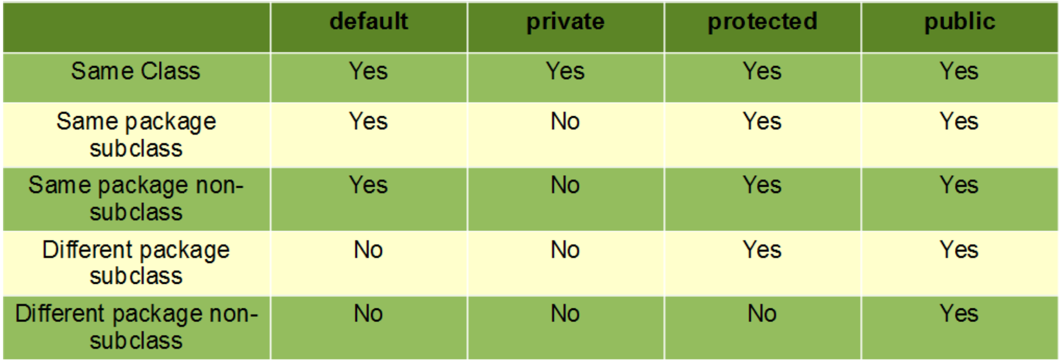
package-private in Java

default or package-private means that the elements are only accessible from the classes in the exact same package.

Protected in Java

- Inside a package you can access to protected methods of any class
- Protected: can only be accessed within its own package and by subclasses defined in another packages

Member Level Visibility in Java



	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

Member Level Visibility in CSharp

- public: accessed by any other elements of any (assembly)
- private: accessed only by code in the **same** class/struct
- protected: accessed only by code in the same **class or subclasses** (derived)
- internal: accessed any code in the **same package** (assembly)
- protected internal: accessed by any code in the package in which it's declared, or from within a subclass in another package
- private protected: accessed only its declaring assembly, by code in the same class or in a type that is derived from that class

About private in Java

- A method in a subclass can be made "more" private
- Instances of subclasses could not be used in place of instance of superclass

Final in Java

- To a class: cannot be extended
- To a method: cannot be redefined
- To an initialized variable: cannot be changed

About Final

- Pay attention because you are not the Kwisatz Harach
- You cannot correctly predict the future

You do not know the future

Avoid premature decisions

- Remember an application average timelife is 15 to 20 years
- You have always two clients: your users and your extender

A course by Stéphane Ducasse
http://stephane.ducasse.free.fr

Reusing some parts of the Pharo Mooc by

Damien Cassou, Stéphane Ducasse, Luc Fabresse
http://mooc.pharo.org