

Chapter 19 – Visibility

What is visibility; give different ways visibility can be achieved from Obj. A to Obj.B?

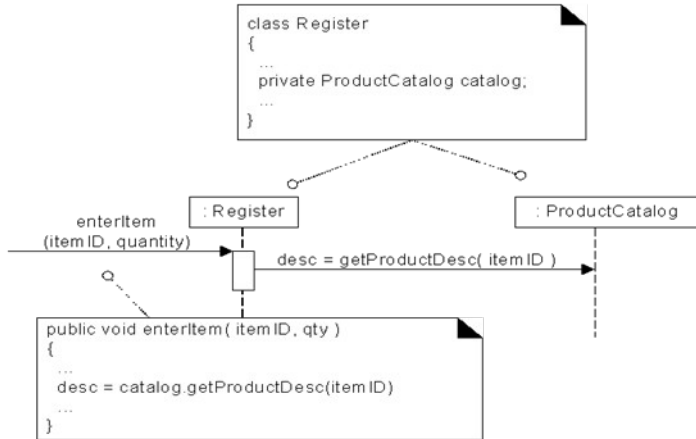
Visibility → is the ability of an object to “see” or have reference to another object. More generally, it is related to the issue of scope: Is one resource (e.g. instance) within the scope of another?

Motivation:

For object A to send a message to object B, A must have a reference to B, B must be visible to A & A must have visibility to B.

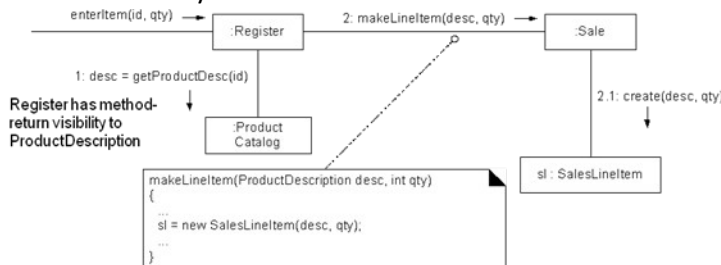
Types of visibility → there are four common ways that visibility can be achieved.

[1] Attribute visibility → from A to B exists when B is an attribute of A (Association). It is a relatively permanent visibility because it persists as long as A and B exists. This is the most common visibility in OO.

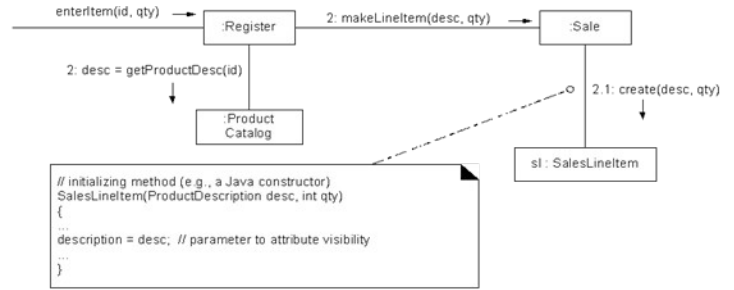


Register needs visibility to ProductCatalog
Attribute visibility: Relatively long-term visibility

[2] Parameter visibility → from A to B exists when B is passed as a parameter to a method of A (Dependency). It is a relatively temporary visibility because it persists as only within the scope of the method. This is the 2nd most common visibility in OO.



Parameter visibility: Relatively short-term
Exists only within scope of method
Parameters used to 'pass around' visibility to objects needing it
Commonly converted to attribute visibility by message receiver
ProductDescription parameter becomes an attribute of SalesLineItem

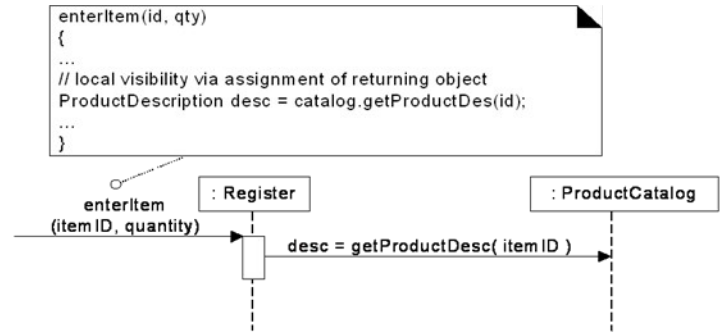


Parameter visibility converted to attribute visibility
SalesLineItem has association with ProductDescription

[3] Local visibility → from A to B exists when B is declared as a local object in a method of A (Dependency). It is a relatively temporary visibility because it persists only within the scope of the method. This is the 3rd most common visibility in OO.

Special case of Local visibility is **Method return**(Dependency)
Two common means by which local visibility is achieved:

- 1] Create a new local instance and assign it to a local variable.
- 2] Assign the returning obj from a method invocation to a local variable.



Local visibility: Register assigns method return to a local variable
Register has local visibility to ProductDescription

[4] Global visibility → from A to B exists when B is globally visible (Dependency). It is a relatively permanent visibility because it persists as long as A and B exists. This is the least common visibility in OO. One way to achieve this, assign an instance to a global variable (possible in C++but not in Java). The preferred method is to use the **Singleton Pattern**.