

ADVANCED CLASSES I

A classifier is a mechanism that has structural features (in the form of attributes), as well as behavioral features (in the form of operations). Classifiers include classes, interfaces, datatypes, signals, components, nodes, use cases, and subsystems. Those modeling elements that can have instances are called classifiers. Every instance of a given classifier shares the same features. The most important kind of classifier in UML is class. The other kinds of classifiers are given in Table: 1.

• Interface	A collection of operations that are used to specify a service of a class or a component
• Datatype	A type whose values have no identity, including primitive built-in types (such as numbers and strings), as well as enumeration types (such as Boolean)
• Signal	The specification of an asynchronous stimulus communicated between instances
• Component	A physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces
• Node	A physical element that exists at run time and that represents a computational resource, generally having at least some memory and often processing capability
• Use case	A description of a set of a sequence of actions, including variants, that a system performs that yields an observable result of value to a particular actor
• Subsystem	A grouping of elements of which some constitute a specification of the behavior offered by the other contained elements

Table:1 Classifier Description

Classifiers represented graphically are shown in Figure: 1.

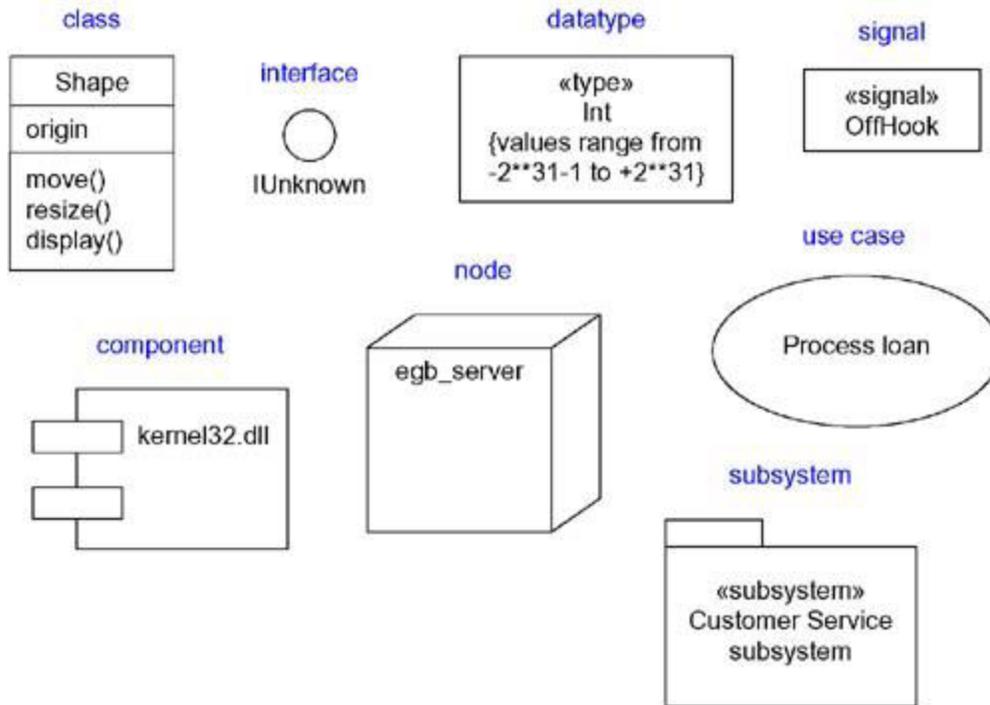


Figure 1: Graphical representation of classifiers

Visibility of a Classifier

Visibility indicates whether the attributes and operations of a classifier can be used by any other classifiers. There are three levels of visibility in UML public, protected and private. A classifier can see another classifier if it is in scope and if there is an explicit or implicit relationship to the target. *Default visibility of a feature is public* in UML. Table 2 shows the various visibilities of the classifiers.

1. public	Any outside classifier with visibility to the given classifier can use the feature; specified by prepending the symbol +
2. protected	Any descendant of the classifier can use the feature; specified by prepending the symbol #
3. private	Only the classifier itself can use the feature; specified by prepending the symbol -

Table:2 Possible Visibilities of Classifier

Figure: 2 indicates the various visibilities of class Toolbar

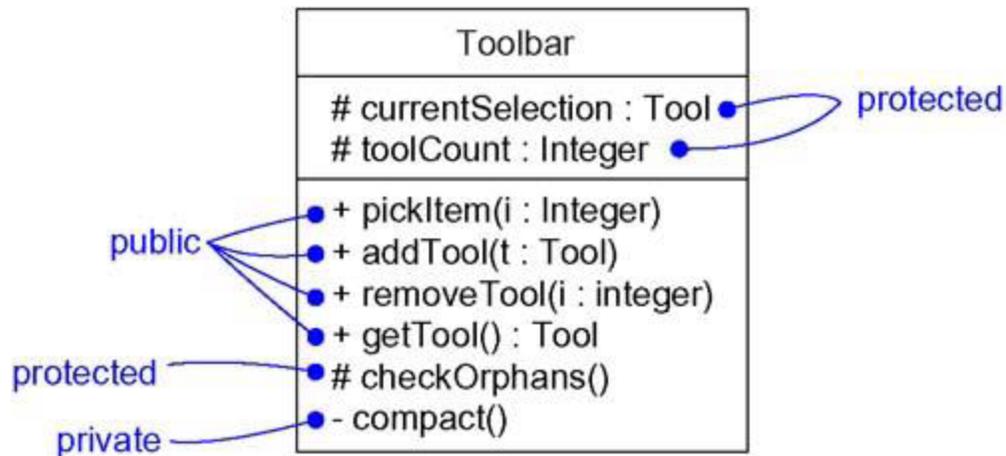


Figure:2 Class showing visibility

Scope(Owner Scope) of a Classifier

The owner scope of a feature(attribute/operations) specifies whether the feature appears in each instance of the classifier or whether there is just a single instance of the feature for all instances of the classifier. *Two kinds of owner scope* – classifier scope and instance scope. *An instance scope* is an owner scope in which each instance of the classifier holds its own value for the feature whereas *classifier scope* is the one which have just one value of the feature for all instances of the classifier. classifier scope is rendered (shown) by underlining the feature's name. No adornment means that the feature is instance scoped. Figure:3 shows the scope of a classifier.

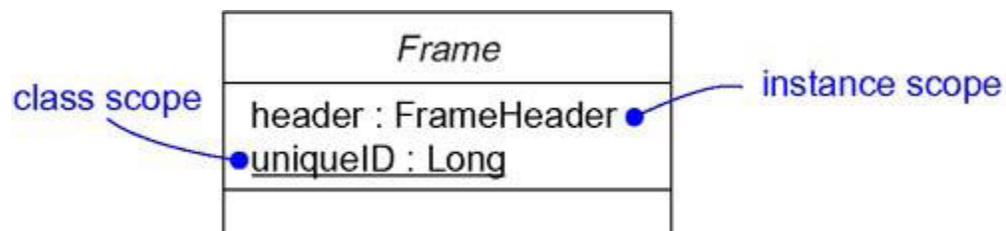


Figure:3 Class showing its Scope

Source : <http://praveenthomasln.wordpress.com/2012/02/25/advanced-classes-under-construction/>