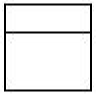
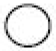
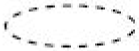
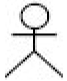
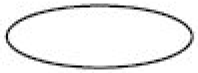
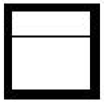
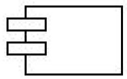
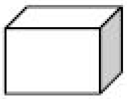
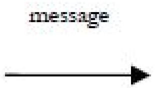
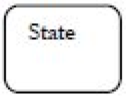
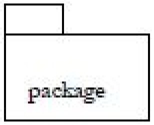
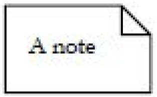
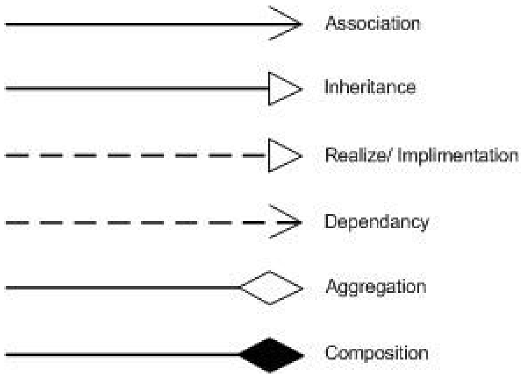
UML ELEMENTS

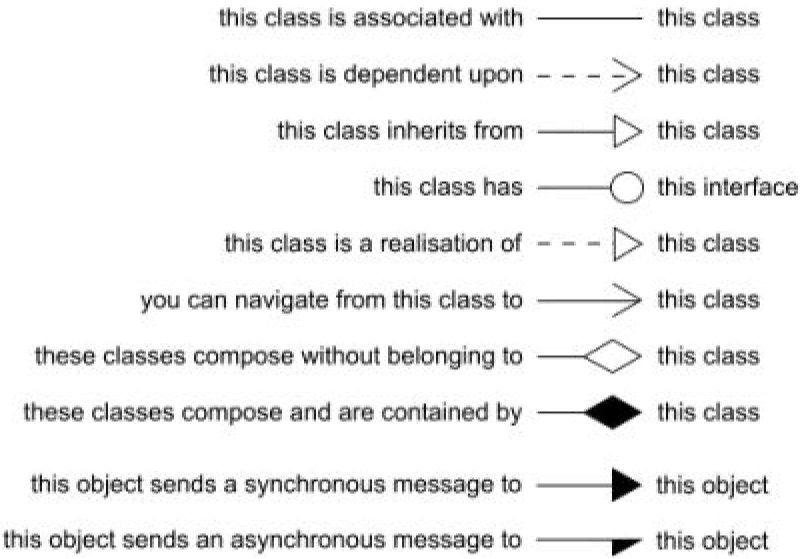
Representations

Sl.	Name	Symbol	Description	Variations/ other related elements/ specialization
1.	**Class**		Description of a set of objects that share the same: attributes, operations, relationships and semantics.	**- actors** **- signals** **- utilities**
2.	**Interface**		A collection of operations that specify a service of a class or component.	
3.	**Collaboration**		An interaction and a society or roles and other elements that work together to provide some cooperative behavior that is bigger than the sum of all the elements. Represent implementation of patterns that make up the system.	
4.	**Actor**		The outside entity that communicates with a system, typically a person playing a role or an external device.	
5.	**Use Case**		A description of set of sequence of actions that a system performs that produces an observable result of value to a particular actor. Used to structure behavioral things in the model.	
6.	**Active class**		A class whose objects own a process or execution thread and therefore can initiate a control activity on their own.	

7.	**Component**		A component is a physical and replicable part that conforms to and provides the realization of a set of interfaces.	
8.	**Node**		A physical resource that exists in run time and represents a computational resource.	
9.	**Interaction**		Set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose.	
10.	**State machine**		A behavior that specifies the sequences of states an object or an interaction goes through during its lifetime in response to events, together with its responses to those events.	
11.	**Packages**		General purpose mechanism of organizing elements into groups.	
12.	**Note**		A symbol for rendering notes and constraints attached to an element or a collection of elements.	

Relationships





this class is associated with this class

this class is dependent upon this class

this class inherits from this class

this class has this interface

this class is a realisation of this class

you can navigate from this class to this class

these classes compose without belonging to this class

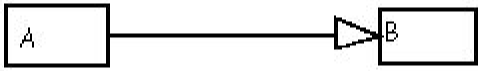
these classes compose and are contained by this class

this object sends a synchronous message to this object

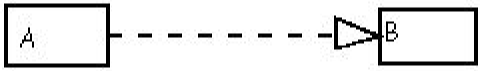
this object sends an asynchronous message to this object

Arrows

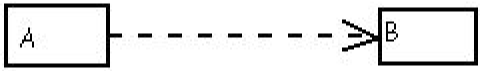
What they explain?



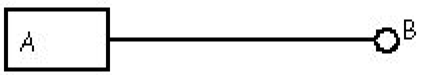
Generalization: A extends B, every A is A KindOf B, A is derived from B, B is the base for A, A is a child of B, A is a subclass of B, B is a superclass of A.



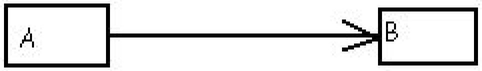
Realization: A implements B, A is derived from abstract B, B is the definition for implementation A, B is an interface and A provides the operations in B, A defines the abstractions listed in B.



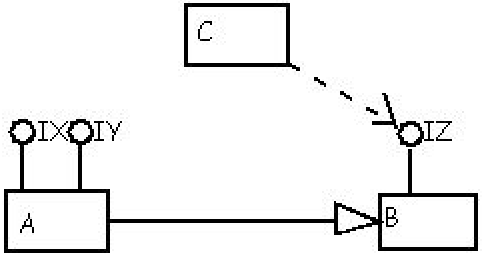
Dependency: A depends B, Changing B can foul up A. Often A uses some of B or A contains B objects, etc.



Realization: A implements B, A is derived from abstract B, B is the definit ion for implementation A, B is an interface and A provides the operations in B, A defines the abstractions listed in B.



Association: A has access to some Bs but B doesn't have access to an A.

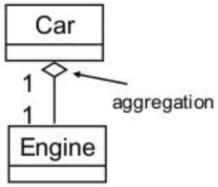


C uses the IZ interface to access B. A extends B and implements IX and IY. Because A extends B it inherits B's IZ interface!

aggregation: "is part of" / “has a”

– symbolized by a clear white diamond

* refers to the formation of a particular class as a result of one class being aggregated or built as a collection.

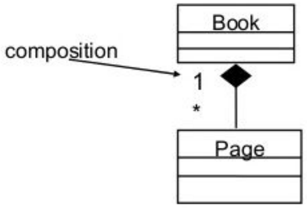


composition: "is entirely made of"

– stronger version of aggregation

– the parts live and die with the whole

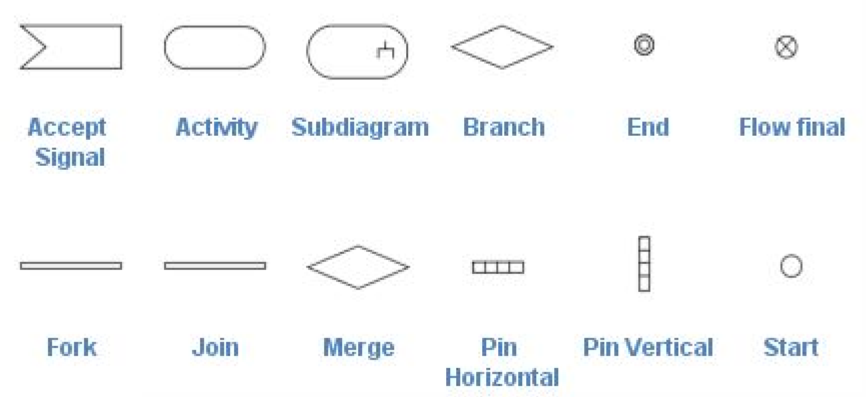
– symbolized by a black diamond



Symbols by UML Diagrams

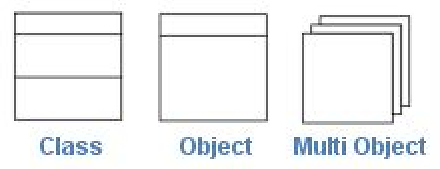
UML Activity Diagram

UML Activity Diagram Shapes include Accept Signal, Activity, Sub diagram, Branch, End, Flow Final, Fork, Join, Merge, Pin Horizontal, Pin Vertical, and Start.



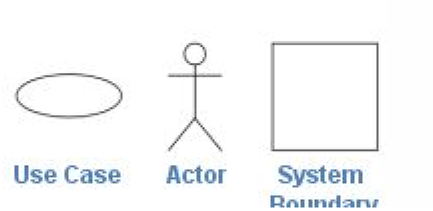
UML Class Diagram

UML Class Diagram Shapes include Class, Object, and Multi Object.



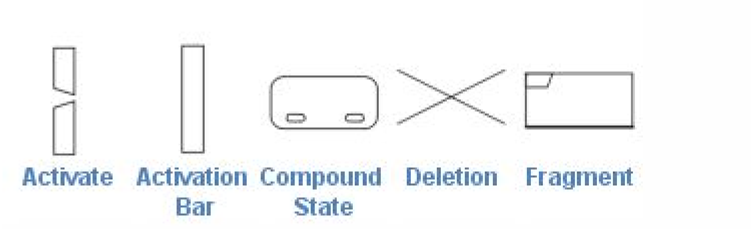
UML Use Case Diagram

UML Use Case Diagram Shapes include Use Case, Actor and System Boundary.



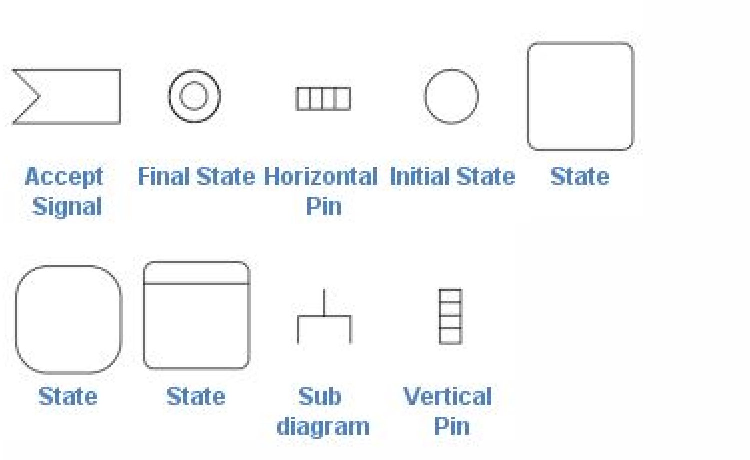
UML Sequence Diagram

UML Sequence Diagram Shapes include Activate, Activation Bar, Compound State, Deletion, and Fragment.



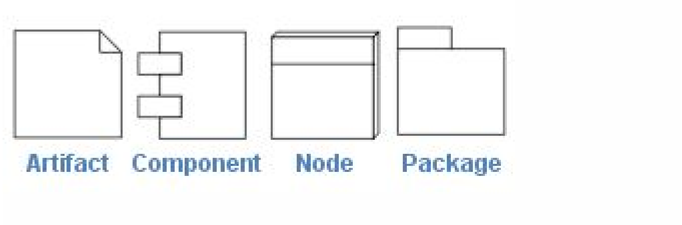
UML State Diagram

UML State Diagram Shapes include Accept Signal, Final State, Horizontal Pin, Vertical Pin, Initial State, State, and Sub Diagram.



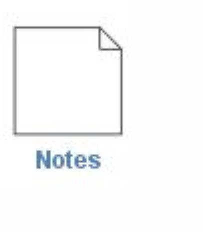
UML Deployment Diagram

UML Deployment Diagram Shapes include Component, Artifact, Node and Package.



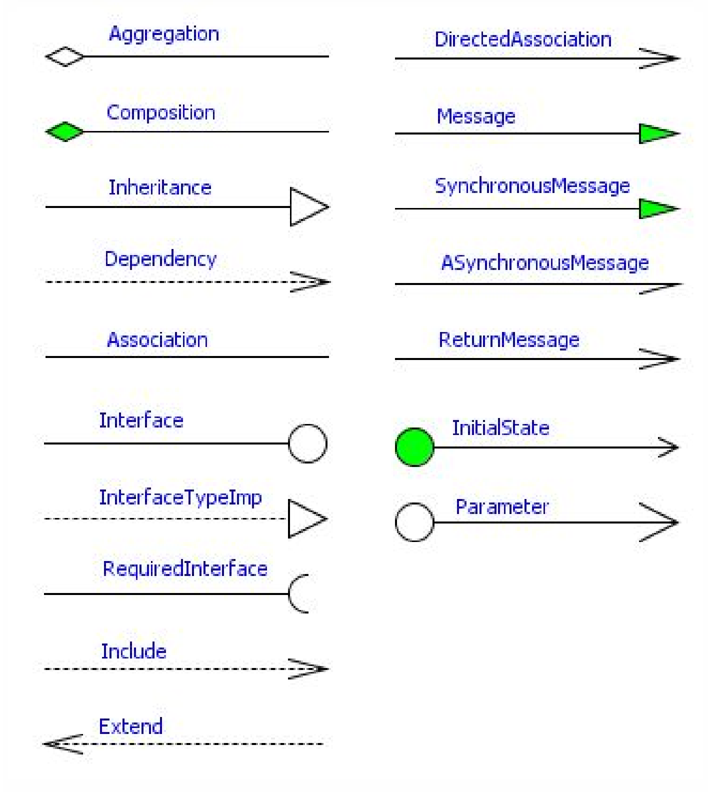
UML Notes

UML Notes Shapes include Notes.



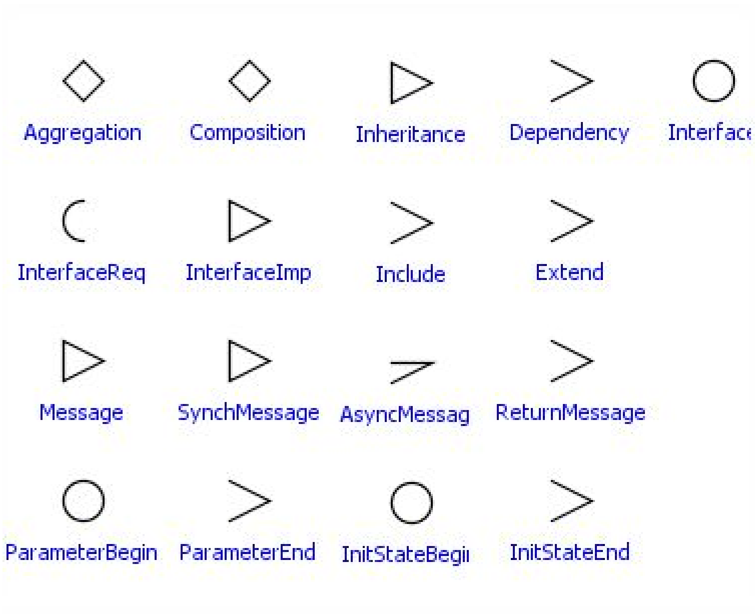
UML Relationships

UML Relationships Shapes include Aggregation, Composition, Inheritance, Dependency, Association, Interface, Interface Implementation, Interface Required, Include, Extend, Direct Association, Message, Synchronous Message, Asynchronous Message, Return Message, Initial State, and Parameter.



UML Arrow Heads

UML Arrow Shapes include Aggregation, Composition, Inheritance, Dependency, Association, Interface, Interface Implementation, Interface Required, Include, Extend, Direct Association, Message, Synchronous Message, Asynchronous Message, Return Message, Initial State, and Parameter.



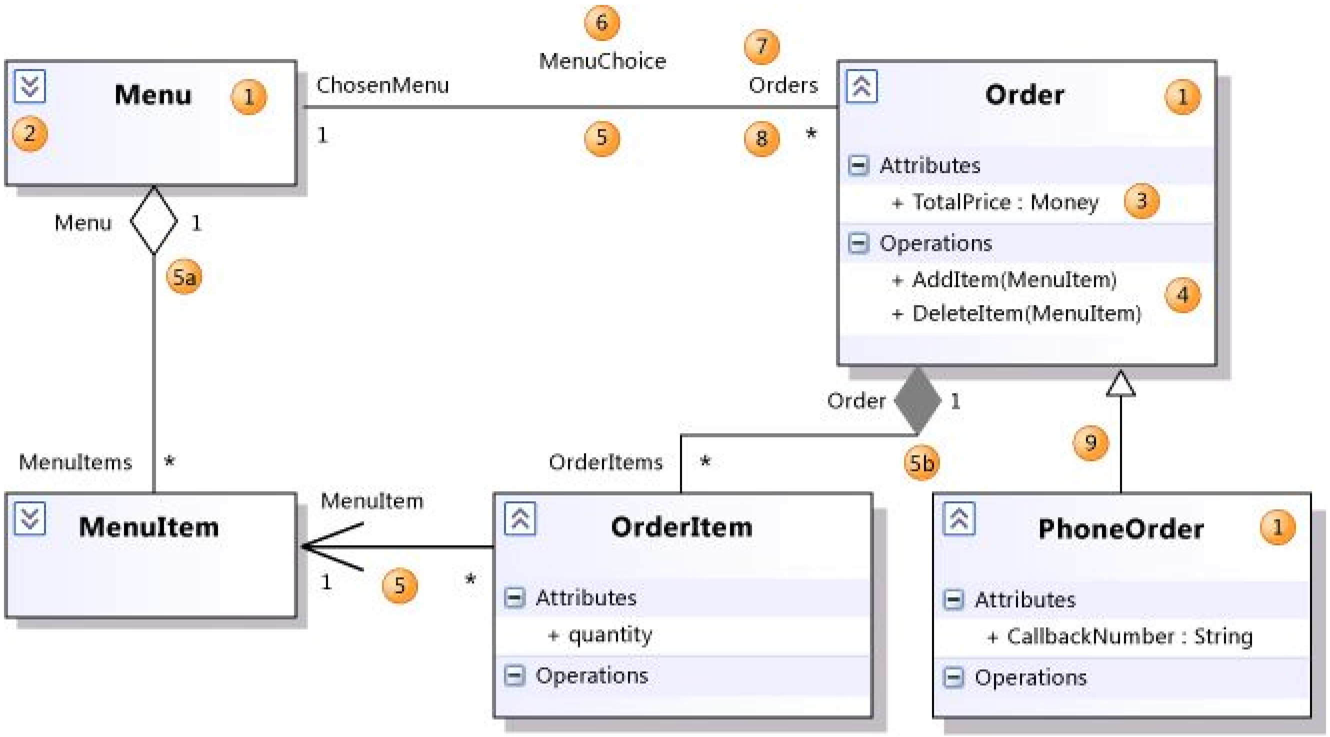
Member visibility (UML class diagram)

+	**Public**
-	**Private**
#	**Protected**
/	**Derived (can be combined with one of the others)**

~	**Package**
*	**Random**

Explaining symbols in UML class diagram

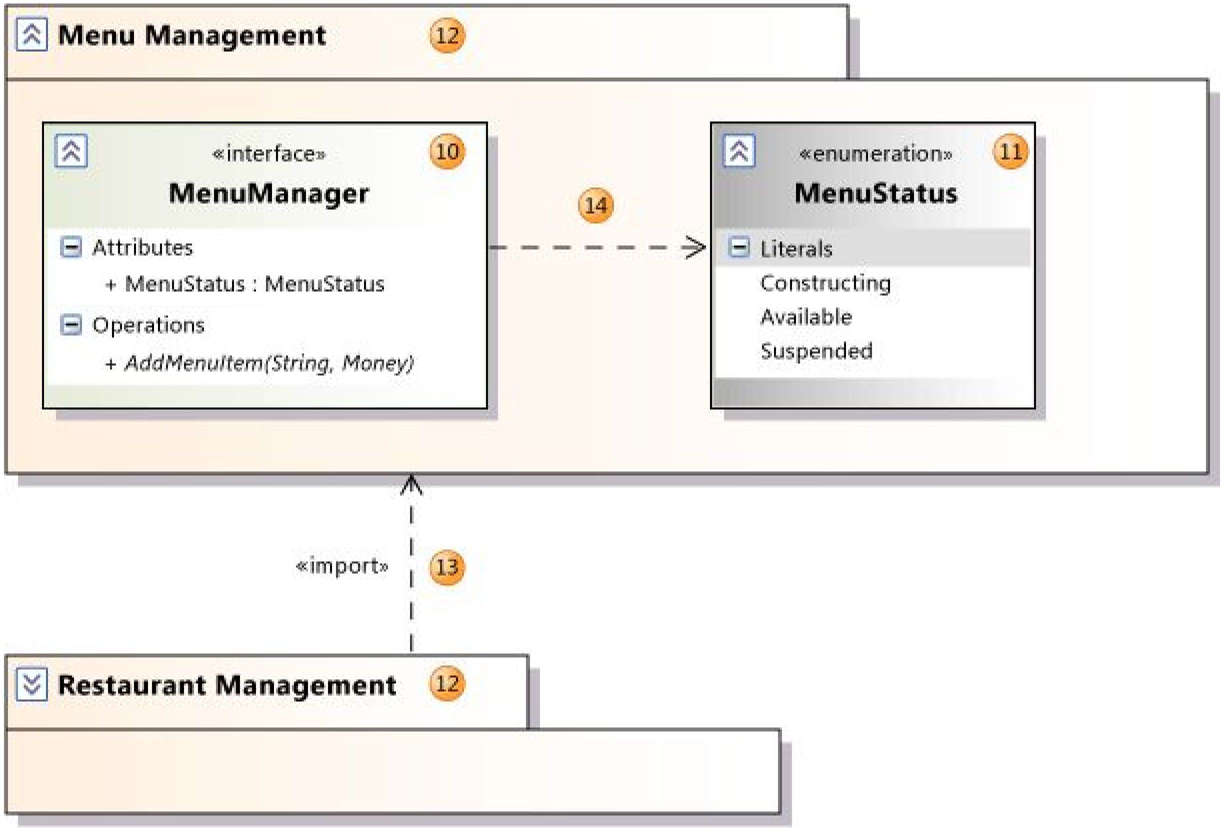
The table in this section describes the elements that you can see on a UML class diagram. For information about the properties of these elements, see the following topics:



Shape	Element	Description
1	**Class**	A definition of objects that share given structural or behavioral characteristics. For more information, see Properties of types on UML class diagrams.

1	Classifier	The general name for a class, interface, or enumeration. Components, use cases, and actors are also classifiers.
2	Collapse/ Expand control	If you cannot see the details of a classifier, click the expander at upper-left of the classifier. You might also have to click the [+] on each segment.
3	**Attribute**	A typed value attached to each instance of a classifier. To add an attribute, click the **Attributes** section and then press **ENTER**. Type the signature of the attribute. For more information, see Properties of attributes on UML class diagrams.
4	**Operation**	A method or function that can be performed by instances of a classifier. To add an operation, click the **Operations** section and then press **ENTER**. Type the signature of the operation. For more information, see Properties of operations on UML class diagrams.
5	**Association**	A relationship between the members of two classifiers. For more information, see Properties of associations on UML class diagrams.
5a	**Aggregation**	An association representing a shared ownership relationship. The **Aggregation** property of the owner role is set to **Shared**.
5b	**Composition**	An Association representing a whole-part relationship. The **Aggregation** property of the owner role is set to **Composite**.
6	**Association Name**	The name of an association. The name can be left empty.
7	**Role Name**	The name of a role, that is, one end of an association. Can be used to refer to the associated object. In the previous illustration, for any Order O, O.ChosenMenu is its associated Menu. Each role has its own properties, listed under the properties of the association.
8	**Multiplicity**	Indicates how many of the objects at this end can be linked to each object at the other. In the example, each Order must be linked to exactly one Menu. ***** means that there is no upper limit to the number of links that can be made.

9	**Generalization**	The *specific* classifier inherits part of its definition from the *general* classifier. The general classifier is at the arrow end of the connector. Attributes, associations, and operations are inherited by the specific classifier. Use the **Inheritance** tool to create a generalization between two classifiers.



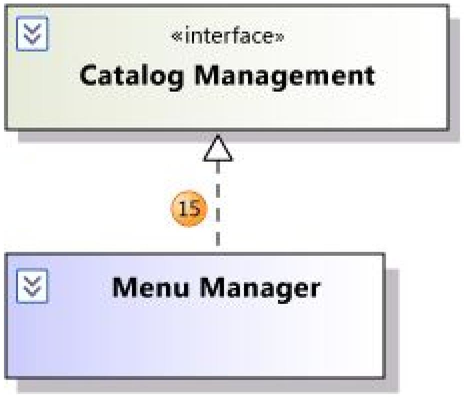
Shape	Element	Description
10	**Interface**	A definition of part of the externally visible behavior of an object. For more information, see [Properties of types on UML class diagrams]().
11	**Enumeration**	A classifier that consists of a set of literal values.
12	**Package**	A group of classifiers, associations, actions, lifelines, components and packages. A logical class diagram shows that the member classifiers and packages are contained within the package.

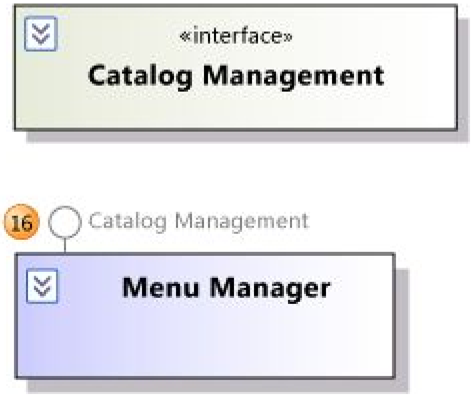
Names are scoped within packages so that **Class1** within **Package1** is distinct from **Class1** outside that package. The name of the package appears as part of the **Qualified Name** properties of its contents.

You can set the **Linked Package** property of any UML diagram to refer to a package. All the elements that you create on that diagram will then become part of the package. They will appear under the package in **UML Model Explorer**.

13 **Import** A relationship between packages, indicating that one package includes all the definitions of another.

14 **Dependency** The definition or implementation of the dependent classifier might change if the classifier at the arrowhead end is changed.





Shape	Element	Description
15	**Realization**	The class implements the operations and attributes defined by the interface. Use the **Inheritance** tool to create a realization between a class and an interface.
16	**Realization**	An alternative presentation of the same relationship. The label on the lollipop symbol identifies the interface. To create this presentation, select an existing realization relationship. An action tag appears near the association. Click the action tag, and then click **Show as Lollipop**.