

UML Diagrams

Dr. Buddhadeb Pradhan

UML Diagrams

UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system.

The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.

UML includes the following nine diagrams, the details of which are described in the subsequent chapters.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

UML - Architecture

Any real-world system is used by different users. The users can be developers, testers, business people, analysts, and many more. Hence, before designing a system, the architecture is made with different perspectives in mind. The most important part is to visualize the system from the perspective of different viewers. The better we understand the better we can build the system.

UML plays an important role in defining different perspectives of a system. These perspectives are –

- Design
- Implementation
- Process
- Deployment

The center is the **Use Case** view which connects all these four. A **Use Case** represents the functionality of the system. Hence, other perspectives are connected with use case.

Design of a system consists of classes, interfaces, and collaboration. UML provides class diagram, object diagram to support this.

Implementation defines the components assembled together to make a complete physical system. UML component diagram is used to support the implementation perspective.

Process defines the flow of the system. Hence, the same elements as used in Design are also used to support this perspective.

Deployment represents the physical nodes of the system that forms the hardware. UML deployment diagram is used to support this perspective.

UML - Modeling Types

It is very important to distinguish between the UML model. Different diagrams are used for different types of UML modeling. There are three important types of UML modeling.

Structural Modeling

Structural modeling captures the static features of a system. They consist of the following –

- Classes diagrams
- Objects diagrams
- Deployment diagrams
- Package diagrams
- Composite structure diagram
- Component diagram

Structural model represents the framework for the system and this framework is the place where all other components exist. Hence, the class diagram, component diagram and deployment diagrams are part of structural modeling. They all represent the elements and the mechanism to assemble them.

The structural model never describes the dynamic behavior of the system. Class diagram is the most widely used structural diagram.

Behavioral Modeling

Behavioral model describes the interaction in the system. It represents the interaction among the structural diagrams. Behavioral modeling shows the dynamic nature of the system. They consist of the following –

- Activity diagrams

- Interaction diagrams

- Use case diagrams

All the above show the dynamic sequence of flow in a system.

Architectural Modeling

- Architectural model represents the overall framework of the system.
- It contains both structural and behavioral elements of the system.
- Architectural model can be defined as the blueprint of the entire system.
- Package diagram comes under architectural modeling.

UML - Basic Notations

- UML is popular for its diagrammatic notations. We all know that UML is for visualizing, specifying, constructing and documenting the components of software and non-software systems. Hence, visualization is the most important part which needs to be understood and remembered.
- UML notations are the most important elements in modeling. Efficient and appropriate use of notations is very important for making a complete and meaningful model. The model is useless, unless its purpose is depicted properly.

Structural Things

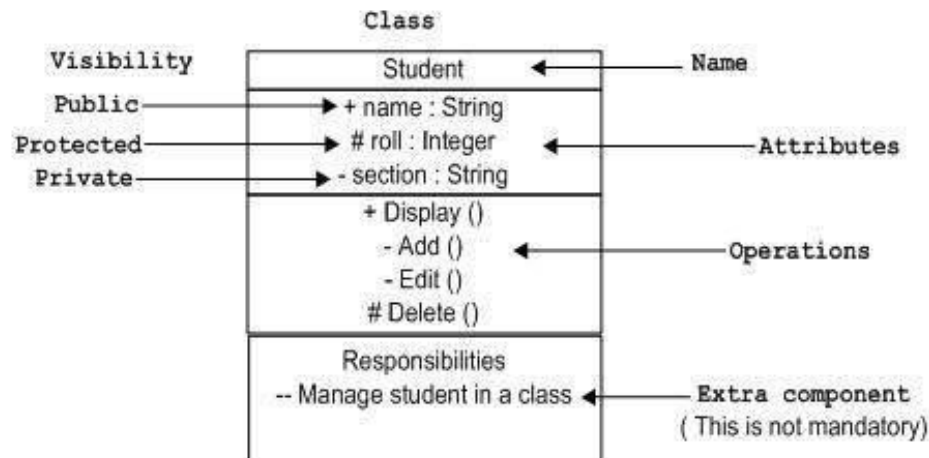
Graphical notations used in structural things are most widely used in UML. These are considered as the nouns of UML models. Following are the list of structural things.

- Classes
- Object
- Interface
- Collaboration
- Use case
- Active classes
- Components
- Nodes

Class Notation

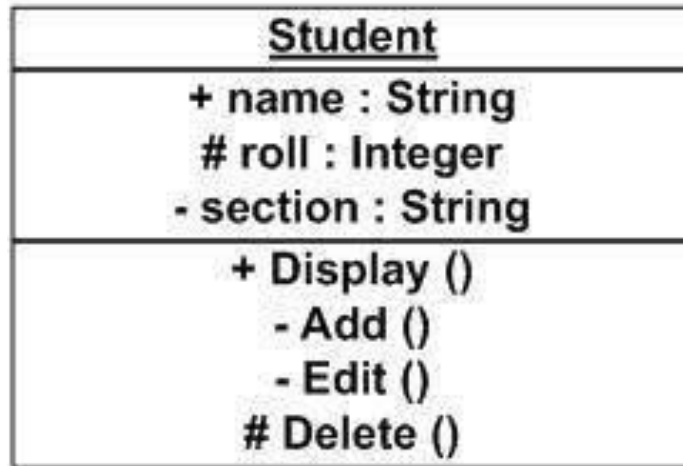
UML *class* is represented by the following figure. The diagram is divided into four parts.

- The top section is used to name the class.
- The second one is used to show the attributes of the class.
- The third section is used to describe the operations performed by the class.
- The fourth section is optional to show any additional components.



Object Notation

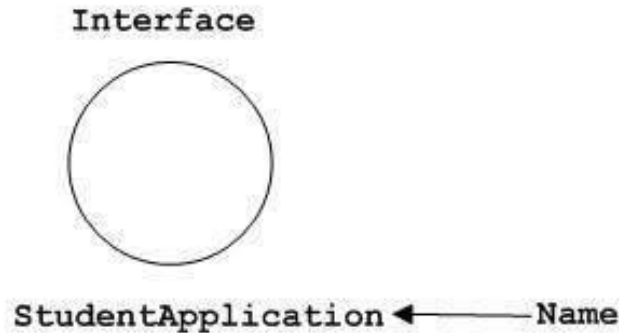
The *object* is represented in the same way as the class. The only difference is the *name* which is underlined as shown in the following figure.



As the object is an actual implementation of a class, which is known as the instance of a class. Hence, it has the same usage as the class.

Interface Notation

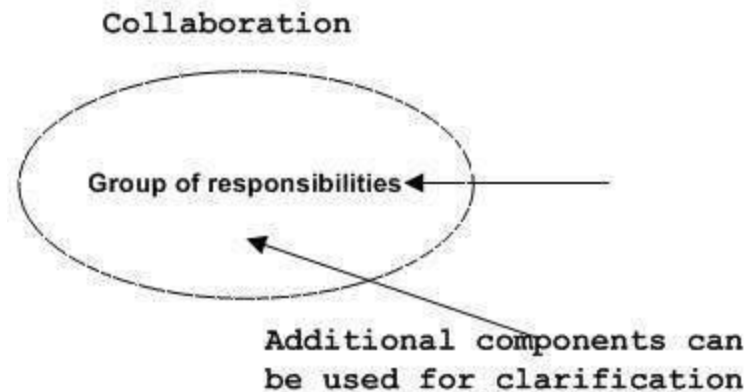
Interface is represented by a circle as shown in the following figure.



Interface is used to describe the functionality without implementation. Interface is just like a template where you define different functions, not the implementation. When a class implements the interface, it also implements the functionality as per requirement.

Collaboration Notation

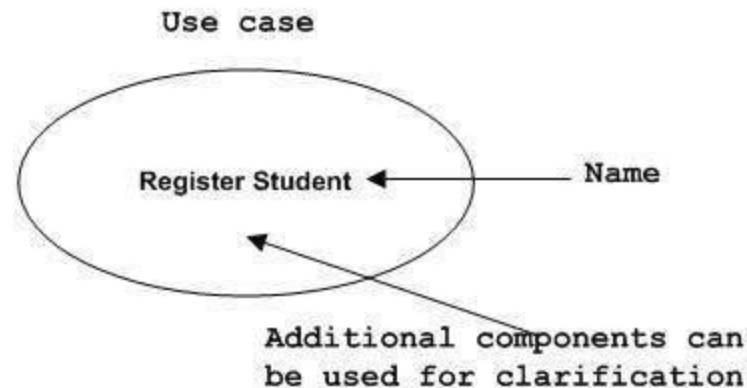
- Collaboration is represented by a dotted ellipse as shown in the following figure. It has a name written inside the ellipse.



- Collaboration represents responsibilities. Generally, responsibilities are in a group.

Use Case Notation

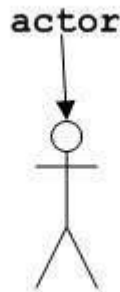
- Use case is represented as an eclipse with a name inside it. It may contain additional responsibilities.



- Use case is used to capture high level functionalities of a system.

Actor Notation

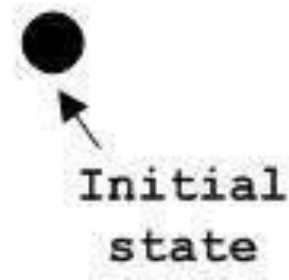
- An actor can be defined as some internal or external entity that interacts with the system.



- An actor is used in a use case diagram to describe the internal or external entities.

Initial State Notation

- Initial state is defined to show the start of a process. This notation is used in almost all diagrams.



- The usage of Initial State Notation is to show the starting point of a process.