**What is a class diagram in UML?**

A**class diagram**is a static structure that is used in software engineering. A class diagram shows the classes, attributes, operations, and the relationship between them. This helps software engineers in developing the code for an application. It is also used for describing, visualizing, and documenting different facets of a system.

Class diagrams are the only UML diagrams that can be mapped directly with object-oriented languages. That is why they are frequently used in the modeling of object-oriented systems and are widely used during the construction of object-oriented systems.

Class diagrams are one of the most important diagrams in coding as they form the basis for component and deployment diagrams and describe the responsibilities in a system. Along with that, they are used for the analysis and design of an application and are also used in forward and reverse engineering.

#### Class Notation

There are three major parts of a class diagram as shown in the image below:

1. Class Name
2. Class Attributes
3. Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

   Mô tả được tạo tự độngClass Operations

A single rectangle is used to represent the class as shown above. The rectangle is divided into three compartments with the topmost being Class Name, then Attributes in the middle, and Operations in the bottom.

##### Class Name

The class name is important for graphical representation. It should be written in bold in the top compartment and start with a capital letter. Moreover, an abstract class should be written in italics.

##### Attributes

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự độngAttributes are written in the middle compartment and list down all the properties of the object being modeled. You can simply add new attributes or derive new attributes from already listed attributes. Attributes must be meaningful and are usually used with the visibility factor that describes the accessibility of an attribute.

##### Operations

Operations are processes that a class knows to carry out. They correspond to the methods of a class. You don’t need to show operations that are similar to attributes because one can already deduce that from the information.

#### Class Relationships

To create a class diagram, the next step is building relationships. There are three main types of relationships here:

1. Generalizations
2. Associations
3. Dependencies

##### Generalizations

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động**Generalizations** are often known as **Inheritance** because it links a subclass to its superclass. The class diagram allows a subclass to inherit from multiple superclasses but it can’t be used to model interface implementation. Checking, Savings, and Credit Accounts are *generalized* by Account

##### Associations

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động**Association** shows a static relationship between two entities. The association between a student and school is “studies”.

The multiplicity factor in association represents how many times an attribute is multiplied. If 100 people work at an organization, then the attribute has multiplied 100 times.

In **[aggregation](https://www.edrawsoft.com/article/uml-aggregation-vs-composition.html" \t "_blank)**, 2 classes have a whole-part relationship. For example, if an employee does not come, the organization will remain there.

Ảnh có chứa văn bản, hàng, Phông chữ, biểu đồ

Mô tả được tạo tự động

Ảnh có chứa văn bản, hàng, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự độngThe aggregation has another special type, called **[composition](https://www.edrawsoft.com/article/uml-aggregation-vs-composition.html" \t "_blank)**. In composition, a class is strongly connected to another class that it will stop functioning without it. For example, if an organization closes, all employees will have to leave.

##### Dependencies

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Phông chữ

Mô tả được tạo tự động**Dependency** shows that one class depends on another. Change in one class will create change in another class. For example, an employee is dependent on the organization.

### **Class Diagram Examples of Common Scenarios**

#### **Class Diagram for Hotel Management System**

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, Song song

Mô tả được tạo tự độngThis hotel management class diagram carefully links all classes joining them together through arrows to show the relationship between them. You can easily customize this hotel management class diagram and add more classes if you like.

#### Class Diagram for Student Registration System

In this class diagram, you can show multiple classes like student, account, course registration manager, course, etc. This class diagram is fairly easy due to its linear design. Registration, course, and account are subclasses of the registration manager and are linked to it using a solid arrow. If your registration system operates a bit differently, then you can add in new classes and easily modify this template.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Hình chữ nhật

Mô tả được tạo tự động