2025-03-26

# UML Class Diagrams

## General Rules

### Attributes and Properties

| Type of Attribute or Property |  |
| --- | --- |
| Public attribute | + attribute : [data type] |
| Private attribute | - attribute : [data type] |
| Public property | + «property» Property : [data type] |
| Private property | - «property» Property : [data type] |

### Methods with Return Type

| Type of Method |  |
| --- | --- |
| Public method with no arguments | + Method() : [return type] |
| Private method with no arguments | - Method() : [return type] |
| Public method with arguments | + Method(argument1 : [data type 1], …, argumentn : [data type n]) : [return type] |
| Private method with arguments | - Method(argument1 : [data type 1], …, argumentn : [data type n]) : [return type] |

### Methods with Void or No Return Type

| Type of Method |  |
| --- | --- |
| Public method with arguments | + Method(argument1 : [data type 1],…, argumentn : [data type n]) |
| Private method with arguments | - Method(argument1 : [data type 1],…, argumentn : [data type n]) |

### Special Cases

| Case | How to Change UML Formatting |
| --- | --- |
| Static | Underline |
| Abstract | Italicize |
| Protected | # instead of + or - |
| Abstract class | Add «Abstract» above class name |
| Interface | Add «Interface» above class name |

### Relationships (Arrows)

| Arrow Type | What It Means |
| --- | --- |
| Open arrow with solid line (⇽) | Inheritance |
| Open arrow with dashed line (◁┈) | Realization |

## Examples

### Simple Example

For example, in the following UML class diagram:



A UML diagram for the ExampleClass1 class ([text version](https:///princomp.github.io/uml/cla/ExampleClass1.txt), [image version](https:///princomp.github.io/uml/cla/ExampleClass1.png), [svg version](https:///princomp.github.io/uml/cla/ExampleClass1.svg))

#### Attributes

* attr1 is private and of type string,
* attr2 is public and of type int,

#### Properties

* Prop1 is a public property of type char (that may have both a set and a get, or only one of them),
* Prop2 is a public property of type double that must contain both a set and a get,
* Prop3 is a private property of type float that contains only a get and no set,

#### Methods

* Method1 is a public method that takes as an argument a sbyte and returns a short,
* Method2 is a public method that takes as an argument a long and does not return anything (that is, its return type is void, here omitted),
* Method3 is a public *abstract* method that does not take any argument and returns a string,
* Method4 is a public *static* method that does not take any argument and returns a char.

#### Constructors

* The first ExampleClass1 is a public constructor that does not take any argument (and does not have any return type),
* The second ExampleClass1 is a private constructor (we know it even if the indication «constructor» is missing, since it has the same name as the class) that takes as an argument a bype (and does not have any return type)

## Forbidden Combination

Note that the following does not make sense:

* A static method cannot be abstract,
* An attribute cannot be abstract,