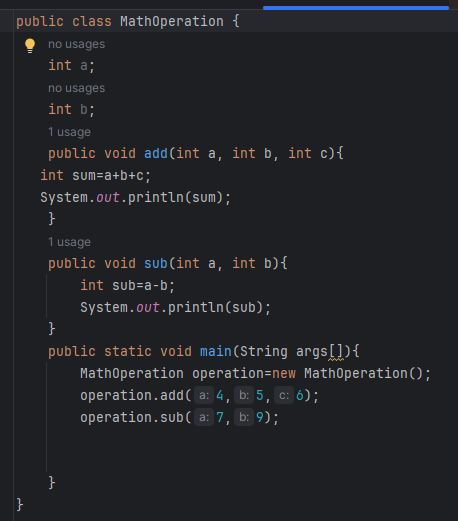
**1. Class in Java**

A **class** is like a blueprint or template for creating objects (instances). It defines the properties (variables) and behaviour’s (methods) that the objects created from the class will have.

* **Properties** (often referred to as **fields** or **attributes**) define the state or data of an object.
* **Methods** define the behaviour of an object by performing operations or actions using the object's data.

**E.g**



Here **MathOperation** is the class name

Int variables are the properties and method defines the behaviour means how operation is being performed, here in e.g add method Is adding 2 number.

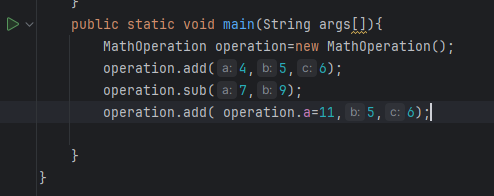
**2. Object in Java**

An **object** is an instance of a class. When you create an object, you are creating a real-world entity based on the blueprint defined by a class.

\*Objects have both **state** (defined by properties) and **behaviour** (defined by methods).

In above program object can access the properties i.e variables a and b, and method add and subtract.

E.g



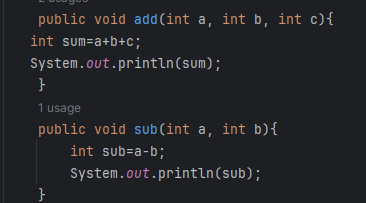
**3. Methods in Java**

A **method** in Java is a block of code that performs a specific task. Methods define the behaviour of objects. They can perform operations, modify object data, and return values.

**Key points about methods:**

* **Method signature**: The method name and parameters.
* **Return type**: The type of value the method returns (e.g., void, int, String).
* **Access modifiers**: Defines the visibility of the method (e.g., public, private).
* **Parameters**: Variables passed into the method to provide input.
* **Method body**: The code inside the method that performs the operations.

e.g :



**Writing the Best Class Names: Best Practices and Tips**

**Class Name:** Should start with upper case letter e.g MathOperation

**Method name:** Should start with lower cases letter e.g addTowNum()

Variable name : should start with lower cases leter