

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

```
public class MathOperation {  
    no usages  
    int a;  
    no usages  
    int b;  
    1 usage  
    public void add(int a, int b, int c){  
        int sum=a+b+c;  
        System.out.println(sum);  
    }  
    1 usage  
    public void sub(int a, int b){  
        int sub=a-b;  
        System.out.println(sub);  
    }  
    public static void main(String args[]){  
        MathOperation operation=new MathOperation();  
        operation.add(a: 4, b: 5, c: 6);  
        operation.sub(a: 7, b: 9);  
    }  
}
```

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

```
public static void main(String args[]){
    MathOperation operation=new MathOperation();
    operation.add( a: 4, b: 5, c: 6);
    operation.sub( a: 7, b: 9);
    operation.add( operation.a=11, b: 5, c: 6);
}
```

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 :

```
public void add(int a, int b, int c){
    int sum=a+b+c;
    System.out.println(sum);
}

1 usage
public void sub(int a, int b){
    int sub=a-b;
    System.out.println(sub);
}
```

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

RameshwarChavan