**Function Categories in java**

In Java, functions are referred to as methods. Methods can be categorized based on various criteria such as their purpose, scope, and usage. Here are some common categories of methods in Java:

**1. Based on Access Modifiers**

* **Public**: Accessible from any other class.
* **Private**: Accessible only within the class they are declared.
* **Protected**: Accessible within the same package and subclasses.
* **Default (Package-Private)**: Accessible only within the same package.

**2. Based on Type**

* **Instance Methods**: Belong to an instance of a class. These methods require an object of the class to be invoked.

public class Example {

public void instanceMethod() {

// Code

}

}

* **Static Methods**: Belong to the class rather than any object instance. They can be called without creating an instance of the class.

public class Example {

public static void staticMethod() {

// Code

}

}

**3. Based on Return Type**

* **Void Methods**: Do not return a value.

public void voidMethod() {

// Code

}

* **Non-Void Methods**: Return a value of a specified type.

public int nonVoidMethod() {

return 42;

}

**4. Based on Parameters**

* **Parameterized Methods**: Take parameters as input.

java

Copy code

public void parameterizedMethod(int param1, String param2) {

// Code

}

* **Parameterless Methods**: Do not take any parameters.

java

Copy code

public void parameterlessMethod() {

// Code

}

**5. Based on Functionality**

* **Accessor Methods (Getters)**: Used to get the value of a private field.

public class Example {

private int value;

public int getValue() {

return value;

}

}

* **Mutator Methods (Setters)**: Used to set the value of a private field.

public class Example {

private int value;

public void setValue(int value) {

this.value = value;

}

}

* **Constructors**: Special methods used to initialize objects. They have the same name as the class and do not have a return type.

public class Example {

public Example() {

// Initialization code

}

}

**6. Based on Use Case**

* **Utility Methods**: Commonly used static methods in utility classes.

public class MathUtils {

public static int add(int a, int b) {

return a + b;

}

}

* **Recursive Methods**: Methods that call themselves to solve a problem.

public int factorial(int n) {

if (n == 0) return 1;

return n \* factorial(n - 1);

}

**7. Based on Lifecycle**

* **Initialization Methods**: Methods that set up the initial state of an object or class.

public void initialize() {

// Code to initialize

}

* **Clean up Methods**: Methods that clean up resources before an object is destroyed.

protected void finalize() throws Throwable {

// Cleanup code

}