Java methods,Types of methods,Recursion,Calling methods and Examples

-K.L.Madhavi

Java Methods

- A method is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation.We can also easily modify code using methods.
- It is used to achieve the reusability of code. We write a method once and use it many times.We do not require to write code again and again.
- The method is executed only when we call or invoke it.

Method Declaration:

Syntax:

```
<access_modifier> <return_type> <method_name>(list_of_parameters)
     {
       //method body
     }
```

Example:

```
public int sum(int a,int b)
     {
       //method body
     }
```

Naming of a Method

- While defining a method,the method name must be start with a lowercase letter.In the multi-word method name, the first letter of each word must be in uppercase except the first word.

  For example:

  i.Single-word method name: sum(), area()

  ii.Multi-word method name: areaOfCircle(), stringComparision()

- Keywords cant be used as method names.
- It is also possible that a method has the same name as another method name in the same class.

Types of Methods

There are two types of methods in Java:

1.Predefined Method

2.User-defined Method

1.Predefined Method: In Java, predefined methods are the methods that is already defined in the Java class libraries. It is also known as the standard library method or built-in method. We can directly use these methods just by calling them in the program at any point.Some predefined methods are length(),sqrt(),equals(), etc…

Each and every predefined method is defined inside a class. Such as print() method is defined in the java.io.PrintStream class.

Example:

```
public class Demo
{
    public static void main(String[] args)
    {
        System.out.print("The maximum number is: " + Math.max(9,7));
     }
}
```

2.User-defined Method: The method written by the user or programmer is known as a user-defined method. These methods are modified according to the requirement.

Example:

i.Creating user defined method

```
public static void findEvenOdd(int num)  {
     if(num%2==0)
        System.out.println(num+" is even");
     else
        System.out.println(num+" is odd");
}
```

ii.Calling the above user defined method

```
import java.util.Scanner;
public class EvenOdd  {
        public static void main (String args[])  {
        Scanner scan=new Scanner(System.in);
        System.out.print("Enter the number: ");
        int num=scan.nextInt();
        findEvenOdd(num);
        } }
```

```
import java.util.Scanner;
public class EvenOdd  {
        public static void main (String args[])
        {
        Scanner scan=new Scanner(System.in);
        System.out.print("Enter the number: ");
        int num=scan.nextInt();
        findEvenOdd(num);
        }
        public static void findEvenOdd(int num)
        {
        if(num%2==0)
        System.out.println(num+" is even");
        else
        System.out.println(num+" is odd");
        }
}
```

Recursion

Recursion is a process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called a recursive function. Using a recursive algorithm, certain problems can be solved quite easily.

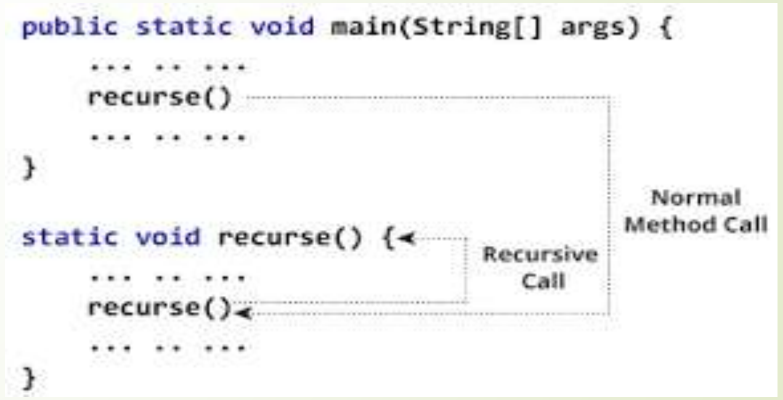
Syntax:

returntype methodname(){

//code to be executed

methodname();//calling same method

}



```
public static void main(String[] args) {
    ... .. ...
    recurse()
    ... .. ...
}

static void recurse() {
    ... .. ...
    recurse()
    ... .. ...
}
```

Recursion Example Program

```
public class RecursionExample {
    static int factorial(int n){
        if (n == 1)
            return 1;
        else
            return(n * factorial(n-1));
        }

public static void main(String[] args) {
    System.out.println("Factorial of 5 is: "+factorial(5));
}
}
```

Calling Methods

How to Call a Method in Java

The method is a collection of statements that performs a specific task or operation. It is widely used because it provides reusability of code means that write once and use it many times. Each method has its own name by which it is called. When the compiler reads the method name, the method is called and performs the specified task.

1. Calling Static Method in Java

A static method is a method that is invoked or called without creating the object of the class in which the method is defined.All the methods that have static keyword before the method name are known as static methods.We can call a static method by using the ClassName.methodName.

The best example of the static method is the main() method. It is called without creating the object.

Example of Calling a Static Method

i.Single class

```
class StaticDemo
{
     static void display()
    {
          System.out.println(“Static method”);
     }
    public static void main(String args[])
    {
     display();
    }
}
```

ii.Multiple classes

```
class StaticDemo
{
     static void display()
     {
          System.out.println(“Static method”);
     }
 }
class Main
{
     public static void main(String args[])
     {
          StaticDemo.display();
     }
 }
```

2.Calling the Abstract Method in Java

An abstract method is a method that is declared with an abstract keyword. The abstract method only has a method declaration.The body of the abstract method defined in the derived class.The abstract method must be declared in the abstract class.

Abstract class:

i.Class contains atleast one abstract method.

ii. It contain abstract methods and normal methods.

We can define an abstract method as follows:

```
abstract public void findArea();
```

Example Program:

```
abstract class AbstractMethodExample
{
        abstract void show();
}
public class AbstractMethodCalling extends AbstractMethodExample
{
        void show()
        {
                System.out.println("The abstract method called.");
        }
        public static void main(String args[])
        {
        AbstractMethodCalling obj = new AbstractMethodCalling();
        obj.show();
        }
}
```