

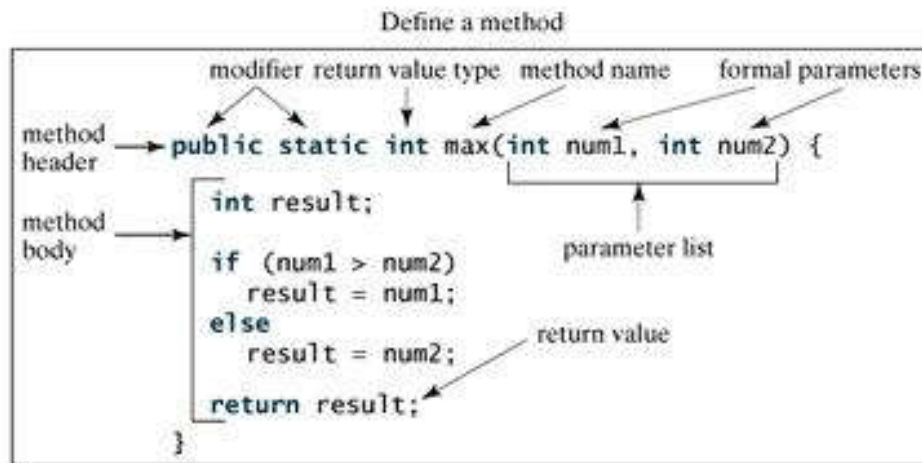
1. Metodi – opšti oblik metoda:

```
modifier returnType methodName(list of parameters)
{
    // Method body;
}
```

2. Ime metoda: početno slovo je malo, a zatim slijede slova, cifre ili simbol ‘_’. Primjeri dobrih imena:

```
run
runFast
getBackground
getFinalData
compareTo
setX
isEmpty
```

3. Primjer:



4. Metod se sastoji od zaglavlja (ili signature, engl. method header) i tijela metoda (engl. method body). U zaglavlju se definišu, pored modifikatora `public` i `static`, tip rezultata, ime metoda i lista argumenata (parametara) metoda. Za svaki argument navodi se tip i ime. U tijelu metoda zadaju se naredbe koje implementiraju šta metod radi.

5. Primjer:

```
/** Vraca maksimum dva broja */
public static int max(int num1, int num2) {
    int result;
    if (num1 > num2)
        result = num1;
    else
        result = num2;

    return result;
}
```

6. Pozivanje metoda koji vraća vrijednost:

```
public class TestMax {
    /** Main metod */
```

```

public static void main(String[] args) {
    int i = 5;
    int j = 2;
    int k = max(i, j);
    System.out.println("Veci do brojeva " + i +
        " i " + j + " is " + k);
    System.out.println(1+max(i+5,12));
}

/** Vraca maksimum dva broja */
public static int max(int num1, int num2) {
    int result;
    if (num1 > num2)
        result = num1;
    else
        result = num2;

    return result;
}
}

```

7. Pri pozivu metoda max, formalni parametri (u primjeru 6. to su num1 i num2) zamjenjuju se stvarnim parametrima (u primjeru 6. to su promjenljive i i k, pa num1 dobija vrijednost 5 a num2 dobija vrijednost 2). Vrijednosti i i k se kopiraju redom u num1 i num2.
8. Ako metod ne vraća vrijednost, tada je njegov tip void.

```

public class TestVoidMethod {
    public static void main(String[] args) {
        printGrade(78.5);
    }

    void printGrade(double score) {
        if (score >= 90.0) {
            System.out.println('A');
        }
        else if (score >= 80.0) {
            System.out.println('B');
        }
        else if (score >= 70.0) {
            System.out.println('C');
        }
        else if (score >= 60.0) {
            System.out.println('D');
        }
        else {
            System.out.println('F');
        }
    }
}

```

9. Pozivanje metoda koji ne vraća vrijednost:

```

public class TestVoidMethod {
    public static void main(String[] args) {
        printGrade(78.5);
    }
}

```

```

public static void printGrade(double score) {
    if (score >= 90.0) {
        System.out.println('A');
    }
    else if (score >= 80.0) {
        System.out.println('B');
    }
    else if (score >= 70.0) {
        System.out.println('C');
    }
    else if (score >= 60.0) {
        System.out.println('D');
    }
    else {
        System.out.println('F');
    }
}
}

```

10. Pri pozivu metoda `printGrade`, formalni parametar (u ovom primjeru to je `score`) zamjenjuje se stvarnim parametrom (u ovom primjeru to je `78.5`).
11. Metod koji je označen kao `static` može direktno pozivati samo druge `static` metode.
12. U jednoj klasi možete definisati više metoda koji imaju isto ime, ali se njihova zaglavlja (signature) moraju razlikovati (vidi primjer 12)
13. Metod `max` sa tri argumenta

```

public class TestMax {

    public static void main(String[] args) {
        int i = 5;
        int j = 2;
        int k = max(i, j);
        System.out.println("veci od brojeva " + i +
            " i " + j + " is " + k);
        System.out.println(1+max(i+5,12));
    }

    /** Vraca veci od brojeva num1 i num2 */
    public static int max(int num1, int num2) {
        int result;
        if (num1 > num2)
            result = num1;
        else
            result = num2;

        return result;
    }

    /** Vraca maksimum tri data cijela broja */
    public static int max(int num1, int num2, int num3) {
        return max(max(num1,num2),num3);
    }
}

```