



Methodlar bilan ishlash


$$f(x)=x^2$$





Method – bu ma'lum bir vazifani bajaruvchi va natijani methodni chaqirgan tomonga qaytaruvchi operatorlar majmuidir.

Method aniq bir vazifani bajarib natijani qaytarmasligi ham mumkin.

Methodlar asosan vaqtni tejash va kodni qayta-qayta yozmasdan takroran ishlatish imkonini berdi



Javada method turlari

- (Javaning) Standart kutubxonasi methodlari
- Foydalanuvchilar tomonidan yaratilgan methodlar

Standart kutubxona methodlari

Standart kutubxona methodlari- Javaning o'zida mavjud va foydalanish uchun qulay bo'lgan methodlardir. Ushbu standart kutubxona java arxiv fayli (*.jar) shaklida JVM va JRE lar bilan birgalikda beriladi.

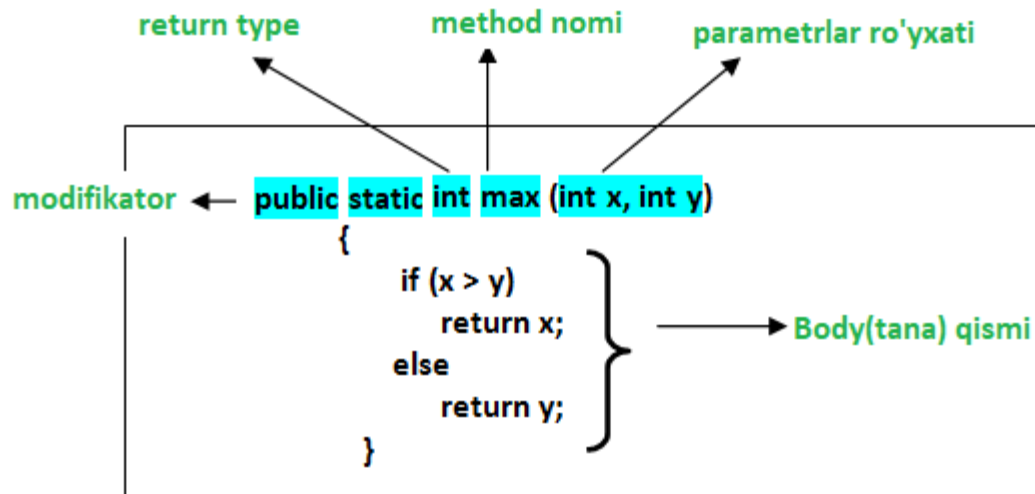
Masalan:

- `java.io.PrintSteam` ning `print()` methodi "" ichidagi matnni ekranga chiqazish uchun ishlatiladi.
- `Math` klassining `sqrt()` methodi sonning ildizini hisoblash uchun ishlatiladi.

```
public class Numbers {  
    public static void main(String[] args) {  
        System.out.print("4 ning kvadrat ildizi: " + Math.sqrt(4)+" ga teng");  
    }  
}
```



Methodning umumiy tuzilishi



Modifikator – ruxsat (доступ) turini aniqlaydi Ya'ni dasturning qaysi qismidan unga murojat qilish mumkinligini belgilaydi. Javada 4 xil modifikator (access modifier) bor:

- **public** – dasturdagi barcha klasslar uchun ruxsat bor.
- **protected** - method e'lon qilingan klass ichida va ostklasslar (voris olgan klass) uchun ruxsat bor.
- **private** - method e'lon qilingan klass ichida ruxsat bor.
- **default** (hech qanday modifikatorsiz e'lon qilingan, default/package) - method e'lon qilingan klass va package uchun ruxsat bor.



- **static**- kalit soʻzi methodni static boʻlishini taʼminlaydi. Static methodlarni klasslardan object olmasdan chaqirish mumkin. Static boʻlmagan methodlar klasslardan olingan objectlar orqali chaqiriladi.
- **Return type** - method qaytaradigan qiymat toifasi, agarda hech narsa qaytarmasa void boʻladi.
- **Method nomi** - kichik harflar bilan yoziladi, agarda ikkita va undan koʻp soʻzdan iborat boʻlsa har bir soʻzning birinchi harfi katta harf bilan(camelCase) yoziladi. Methodga ish harakatni ifodalovchi feʼl soʻz turkumiga oid nom beriladi. Masalan print, write, checkNumber va h.k.
- **Parametrlar roʻyxati(var-args)** - kirish parametrlari toifasi va nomi “,” bilan ajratilgan holda () ichida beriladi.
- **Body(tana) qismi** - belgilangan vazifani bajarish uchun moʻljallangan kod {} ichida yoziladi.

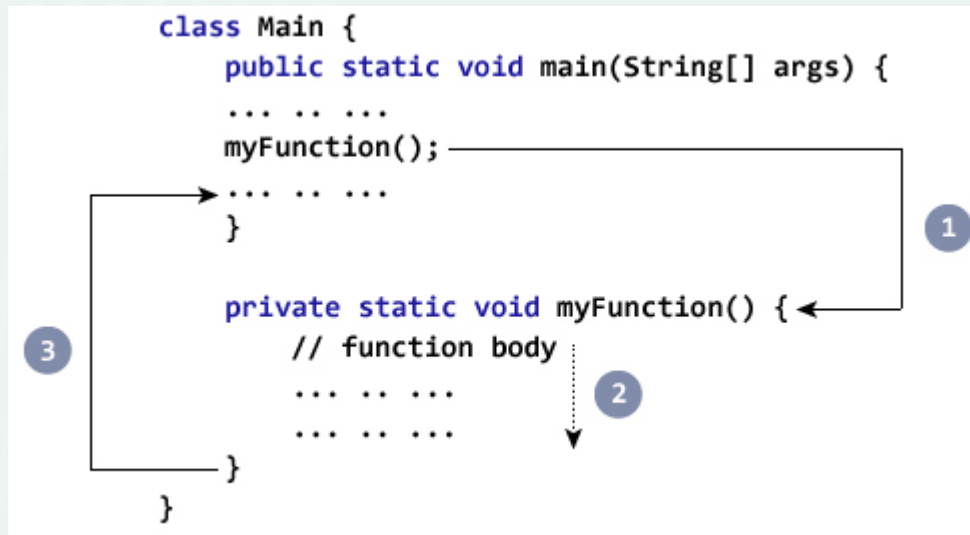
Methodni chaqirish

Method ishlashi uchun uni chaqirish kerak. Methodni chaqirganda 3 xil holat yuz berishi mumkin:

- Methodni ichidagi barcha amallar bajariladi;
- return gacha bajariladi;
- Istisno (Exception) chaqiradi

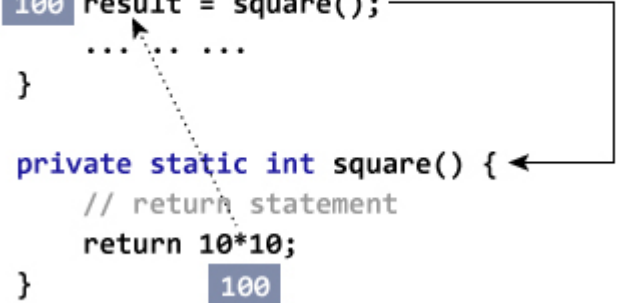


Return type void bo'lgan (hech narsa qaytarmaydigan) methodni chaqirish



Return type void bo'lmagan methodni chaqirish

```
class SquareMain {  
    public static void main(String[] args) {  
        ... ..  
        100 result = square();  
        ... ..  
    }  
  
    private static int square() {  
        // return statement  
        return 10*10;  
    }  
}
```



Return type void bo'lmagan va kirish parametri bor bo'lgan methodni chaqirish

```
class SquareMain {  
    public static void main(String[] args) {  
        ... ..  
        n = 3; 3  
        9 result = square(n);  
        ... ..  
    }  
  
    private static int square(int i) {  
        // return statement 3  
        return i*i;  
    } 9  
}
```



Method overloading (Methodni qayta yuklash)

Bitta klassda bir xil nomdagi va parametrlari turlicha bo'lgan bir necha method bo'lsa –methodni qayta yuklash (method overloading) deyiladi



```
public class Test{
    // for integer
    public static int min (int n1, int n2) {
        int min;
        if (n1 > n2){
            min = n2;
        }
        else {
            min = n1;
        }
        return min;
    }
    // for double
    public static double min (double n1, double n2) {
        double min;
        if (n1 > n2){
            min = n2;
        }
        else {
            min = n1;
        }
        return min;
    }
}
```

