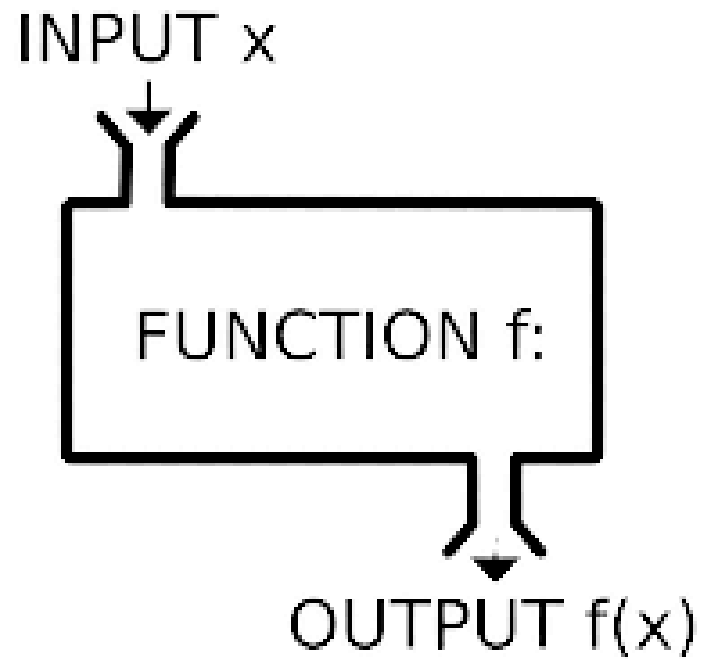


# Metodlar bilan ishlash

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

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



Metod – bu ma’lum bir vazifani bajaruvchi va natijani metodni chaqirgan tomonga qaytaruvchi operatorlar majmuidir.

Metod - aniq bir vazifani bajarib natijani qaytarmasligi ham mumkin.

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

# C# da metod turlari

- Standart kutubxonasi metodlari
- Foydalanuvchilar tomonidan yaratilgan metodlar

# Standart kutubxona metodlari

Standart kutubxona metodlari – C# va .NET ning oʻzida mavjud va foydalanish uchun qulay boʻlgan metodlardir. Ushbu standart kutubxonalar sdk bilar birgalikda oʻrnatiladi. Masalan:

- `System.Console` ning `Console.WriteLine()` methodi turli toifaga ega boʻlgan maʼlumotlarni ekranga chiqarish uchun ishlatiladi
- `System.Math` klassining `Sqrt()` methodi sonning ildizini hisoblash uchun ishlatiladi.



```
using System;
```

```
namespace CSHARP  
{
```

```
    class Methods  
    {
```

```
        static void Main(string[] args)  
        {
```

```
            Console.WriteLine("4 ning kvadrat ildizi: "  
                               + Math.Sqrt(4) + " ga teng")
```

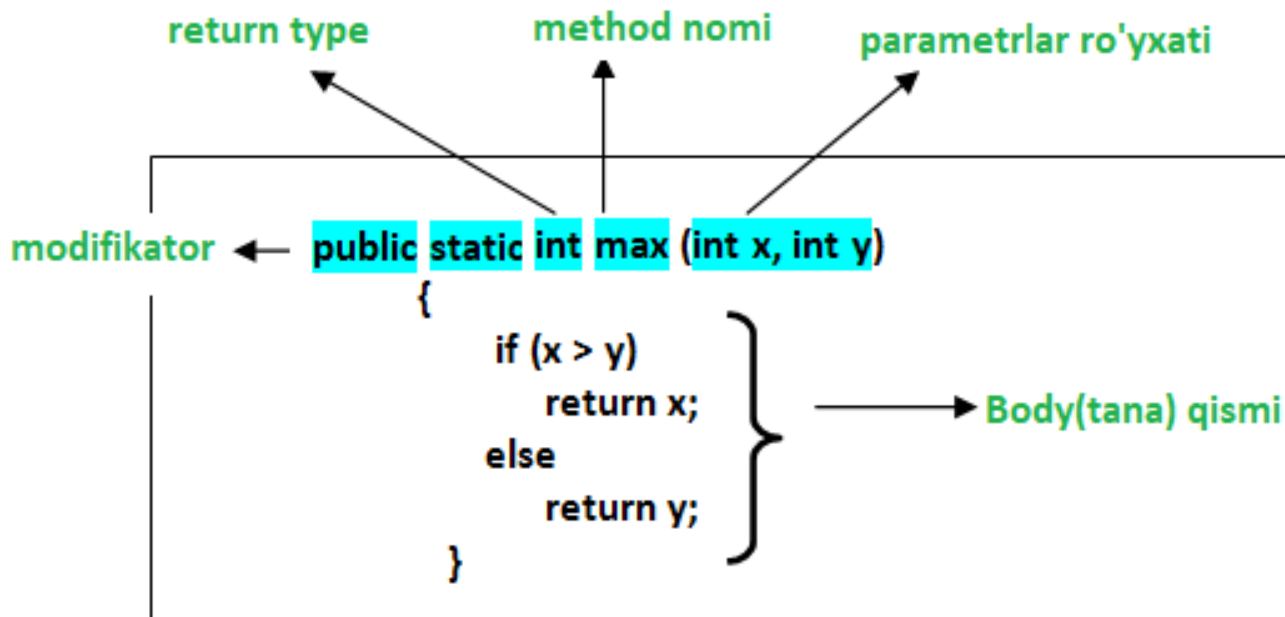
```
        }
```

```
    }
```

```
}
```

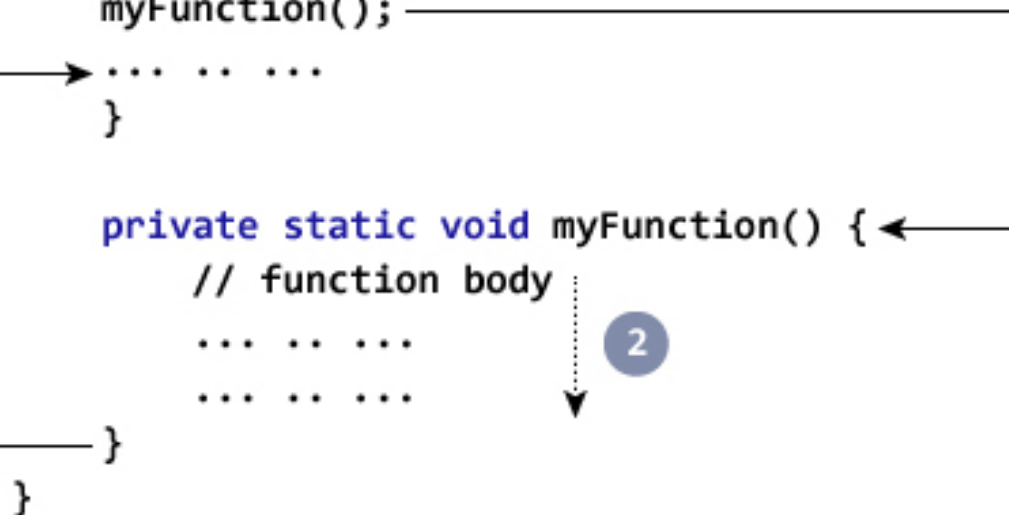


# Metodning umumiy tuzilishi

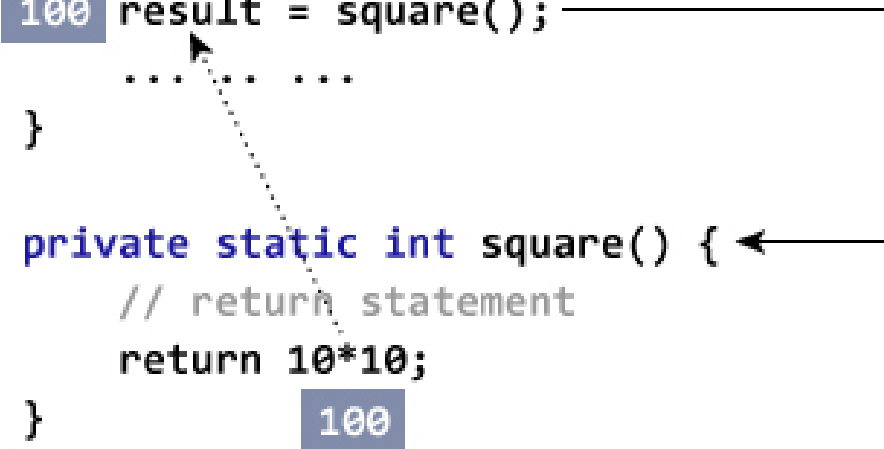


- **static** - kalit so'zi metodni static bo'lishini ta'minlaydi. Static metodlarni klasslardan object olmasdan chaqirish mumkin. Static bo'lmagan metodlar klasslardan olingan objectlar orqali chaqiriladi.
- **Return type** - metod qaytaradigan qiymat toifasi, agarda hech narsa qaytarmasa **void** bo'ladi.
- **Method nomi** - katta harflar bilan yoziladi, agarda ikkita va undan ko'p so'zdan iborat bo'lsa har bir so'zning birinchi harfi katta harf bilan(PascalCase) yoziladi. Metodga ish harakatni ifodalovchi fe'l so'z turkumiga oid nom beriladi. Masalan Print, Write, CheckNumber va h.k.
- **Parametrlar ro'yxati(var-args)** - kirish parametrlari toifasi va nomi “,” bilan ajratilgan holda () ichida beriladi.
- **Body(tana) qismi** - belgilangan vazifani bajarish uchun mo'ljallangan kod {} ichida yoziladi.

```
class Main {  
    public static void main(String[] args) {  
        ... ..  
        myFunction();  
        ... ..  
    }  
  
    private static void myFunction() {  
        // function body  
        ... ..  
        ... ..  
    }  
}
```

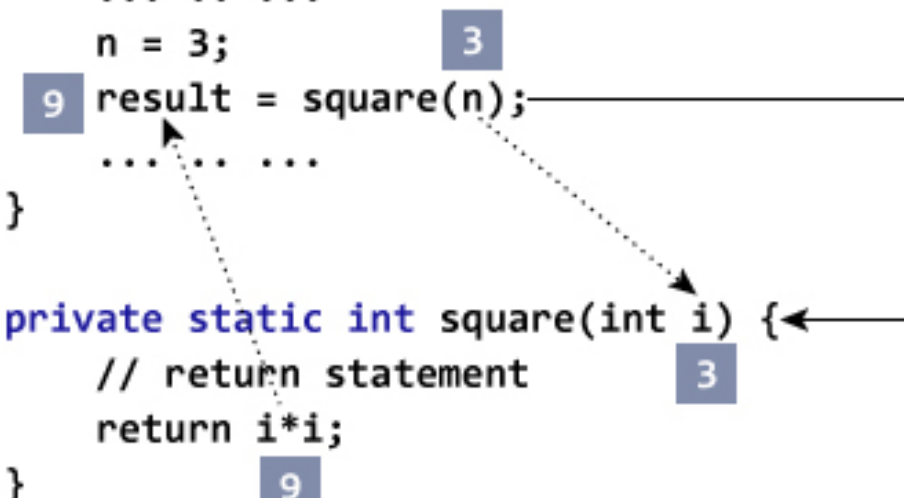


```
class SquareMain {  
    public static void main(String[] args) {  
        ... ..  
        100 result = square();  
        ... ..  
    }  
  
    private static int square() {  
        // return statement  
        return 10*10;  
    }  
}
```




The diagram illustrates the execution flow. A solid line connects the `square()` call in the `main` method to the `square()` method definition. A solid arrow points from the `square()` method to the `100` value, which is then passed back to the `result` variable in the `main` method via a dotted arrow.

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



# Method overloading (Metodni qayta yuklash)

Bitta klassda bir xil nomdagi va parametrlari turlicha bo'lgan bir nechta method bo'lsa yoki qaytaradigan toifasi turlicha bo'lsa – metodni qayta yuklash (method overloading) deyiladi



```
public class Methods
{
    static int Min (int son1, int son2)
    {
        return son1 > son2 ? son1 : son2;
    }

    static double Min (double son1, double son2)
    {
        return son1 > son2 ? son1 : son2;
    }
}
```

# ref va out

ref – bu metoddagi argumentga reference type kabi murojaat qilish imkonini beradi. Argumentni qabul qilishda, aynan undan nusxa olish emas, balki kelayotgan manzilning aynan o'zini egallashga yordam beradi.

out – ham ref kabi ishlaydi, faqatgina unda metodga qiymat yuborishdan oldin, unga qiymat yuborish shart emas. Odatda outdan bir nechta qiymatlar qaytarishda foydalaniladi.



# ref vs out

- refda parametrlar avval yuborilishdan oldin qiymat olishligi kerak. out uchun shart emas
- refda qiymat qaytarishdan oldin, unga qiymat berish shart emas. out uchun shart