

Тел: +7 (707) 900 92 67

Почта: saken.yan@yandex.com

4 ЛЕКЦИЯ

C-SHARP АҚПАРАТТЫҚ БАҒДАРЛАМАЛАУ ТІЛІ**§3. C-Sharp тіліндегі әдістер (методы).****Жалпы түсінік.**

Егер айнымалылар мәндерді сақтауға арналған болса, әдістер белгілі бір әрекеттерді орындайтын кодтардың жиынтығы болып табылады. Негізінде C# тілінде әдіс дегеніміз - бұл белгілі бір әрекеттерді немесе басқаша айтқанда процедураларды орындайтын кодтар жиынтығының бөлігі.

Әдістерді жариялаудың жалпы синтаксисі мынадай:

```
[модификатор] қайтаратын_айнымалы_типі әдістің_аты ([параметрлар])
{
    // әдістің денесі
}
```

Мысал.

```
using System;
namespace Project
{
    class Program
    {
        //Print әдісі ақпаратты консолға шығарады:
        public static void Print(string s)
        {
            Console.WriteLine("You got a new message:");
            Console.WriteLine(s);
        } //Print әдісінің соны

        //Main әдісі. Программа Main әдісінен басталады:
        static void Main(string[] args)
        {
            //Print әдісін Main әдісіннен шақыру:
            Print("This message from method Print");

            Console.ReadKey();
        } //Main әдісінің соны
    }
}
```

Мысалда Program атты класс ішінде Print және Main деп аталатын екі статикалық әдіс бар. Программа Main әдісінен басталып, Main әдісінің аяқталуымен тоқтайды.

Мысалда көрсетілген Print әдісін талқылайық. Бұл әдістің модификаторлары **public** және **static**. Қайтаратын типтің түрі **void** болып табылады. яғни бұл тип ешнәрсе қайтармайды. Print әдісі **string** типті бір ғана параметр қабылдайды да, нәтиже ретінде еш нәрсе қайтармайды. Бұл әдіс екі әрекетті орындайды: Біріншісі, **"You got a new message:"** деген ақпаратты консолға шығарады. Екіншісі, Print әдісін шақыру кезінде осы әдіске берген параметрді консолға шығарады. Print әдісін біз Main әдісі ішінен шақырып тұрмыз (яғни осы әдісті компиляторға орында деп айтып отырмыз).

Әдістердің қабылдайтын параметрлері.

Егер де әдіс ешқандай параметрлерді қабылдасын десек, онда әдіс атауынан кейін бос жақшаларды көрсетеміз. Мысалы:

```
public static void Method()//←пустые скобки
{ ... }
```

Егер біз әдіс екі немесе одан да көп параметрлерді қабылдасын десек, онда осы параметрлерді үтір арқылы бөліп жазуымыз керек. Жақша ішінде параметрлерді жазған кезде алдымен параметрдің типін, содан соң параметрдің атын көрсетуіміз қажет. Мысалы:

```
public static void Method(string s, int m)//← параметрлер
{ ... }
```

Бұл жерде әдіс екі параметрден тұрады. Бұл **s** деген атты **string** типті параметр және **m** деген атты **int** типті параметрі.

Параметр ретінде айнымалылар, массивтер, тізімдер, объектілер, тіпті әдістердің өздері де болуы мүмкін. Мысалы, әдіске параметр ретінде **string** типті массивті берейік және де әдіс осы массивтің элементтерін консольға шығарсын.

```
public static void Method(string[] array)
{
    for(int i = 0; i < array.Length; i++) Console.WriteLine(array[i]);
}
```

Әдістің нәтиже ретінде қайтаратын мәні.

Әдіс нәтиже ретінде мәнді қайтара алады. Жоғарыдағы мысалда әдістер нәтиже ретінде **void** типті нәтиже қайтарады яғни еш нәрсе қайтармайды. Бұндай әдістер тек кейбір әрекеттерді жасайды. Егер әдіс **void** емес кез-келген басқа типті болса, онда мұндай әдіс осы типтің мәнін қайтаруы керек. Ол үшін **return** операторы қолданылады **return** операторынан кейін қайтару керек айнымалы жазылады:

```
return қайтару_керек_айнымалы_аты;
```

Мысал.

```
using System;
namespace Project
{
    class Program
    {
        //GetSum әдісі екі бүтін санның қосындысын табады:
        public static int GetSum(int x, int y)
        {
```

```

    int z = x + y;
    return z;
}

//Main әдісі.
static void Main(string[] args)
{
    //GetSum әдісін 5 және 10 деген параметрлермен Main-нан шақыру:
    int Sum = GetSum(5,10);

    //GetSum әдісінің нәтижесін Sum деген айнымалыға меншіктеп,
    //консолға шығару:
    Console.WriteLine("The sum is:" + Sum);
    Console.ReadKey();
}
}
}

```

Әдісті үзу.

return операторы тек мәнді қайтарып қана қоймайды, сонымен қатар әдісті үзе алады.

Мысалы:

```

public static void Convert(int hour)
{
    if(hour > 24) {
        Console.WriteLine("Wrong value!");
        return;//осы жерде әдіс сонына жетпей үзіледі.
    }

    if(hour < 12) Console.WriteLine("AM");
    if(hour > 12) Console.WriteLine("PM");
}

```

Әдістерді қолдану мысалы. 3-ші дәрістегі мысалды еске түсірейік, ол жерде программаны пайдаланушы b_1 геометриялық қатардың бірінші мүшесін, қатарының $[0,1]$ аралықтағы q өсімшесін және соңғы мүшенің реттік нөмерін n еңгізетін.

Программа консолға қатардың барлық элементтерін және n мүшесінің қосындысын шығаруы керек болатын. Бұл мысалда b_1, q, n дерді қауіпсіз енгізу үшін біз бір біріне ұқсас тексеріс кодтарын жазған болатынбыз. Алайда бір кодты қайта-қайта жаза бермей, оларды арнайы бір әдіске бөліп шығарайық:

```

// Input_db әдісі double типті айнымалыны консолдан қауіпсіз оқиды.
public static double Input_db(string message)
{
    while (true)
    {
        Console.WriteLine(message);
        try{
            double d = double.Parse(Console.ReadLine());
            return d;
        }
        catch(Exception e){ Console.WriteLine(e.Message); }
    }
}

```

```

//Input_ui әдісі uint типті айнымалыны консолдан қауіпсіз оқиды.
public static uint Input_ui(string message)
{
    while (true)
    {
        Console.WriteLine(message);
        try{
            uint u = uint.Parse(Console.ReadLine());
            return u;
        }
        catch (Exception e){ Console.WriteLine(e.Message); }
    }
}

static void Main(string[] args)
{
    //b1 Input_db әдісі арқылы қауіпсіз консолдан еңгізу
    double b1 = Input_db("Enter the first term of the series:");

    double q = 0; // q ды қауіпсіз консолдан еңгізу
    while(true){
        q = Input_db("Please, Enter the q");
        if(q > 0 && q < 1) break;
    }
    //n Input_ui әдісі арқылы қауіпсіз консолдан еңгізу
    uint n = Input_ui("Please, Enter the n");

    double[] b = new double[n+1]; b[0] = 1;

    for (int i = 1; i < b.Length; i++){
        b[i] = b[i-1]*q; Console.WriteLine("b[" + i + "]=" + b[i]);
    }

    double sum = 0;
    for (int i = 1; i < b.Length; i++){ sum = sum + b[i]; }

    Console.WriteLine("s = " + sum);
    Console.ReadLine();
} // Конец Main

```

Студент өз бетімен орындайтын тапсырма. Есеп №1.

1.1. Аты *RandArray* деп аталатын `int` типтес 100 элементке арналған массивті жариялау.

1.2. **Complete** деген әдісті жариялаңыз. Бұл әдіс параметр ретінде `int` типтес массивті қабылдайды, **Complete** әдісі ішінде параметр ретінде келген массив мәндері 0 мен 1000 аралығында жататын кездейсоқ шамалармен толтырыуы тиіс, содан соң толтырылған массивті нәтиже ретінде қайтарып беруі керек.

1.3. **Average** деген әдісті жариялаңыз. бұл әдіс параметр ретінде `int` типті массивті қабылдап, `double` типті айнымалыны нәтиже қайтару керек. Осы қайтарылатын айнымалы параметр ретінде келген массивтің элементтерінің орташасына тең.

