

3 дәріс. Объектілерді әдістерге беру. ref және out модификаторлары.

Дәрістің мақсаты: студенттерде объектілермен әдістердің құрамында жұмыс істеу ерекшеліктері туралы түсініктерін көрсетуге қабілет қалыптастыру.

Осы дәрісті меңгеру нәтижесінде студенттер келесі қабілеттерге ие болады:

- Әдістерге объектілерді беру тәсілдері бойынша түсініктерін көрсету;
- ref және out модификаторларын пайдалану жағдайларын ажырату.

Бұған дейін әдістерге параметрлер бергенде **int** немесе **double** типтері қолданылды. Бірақ әдістерге сілтемелік типтегі параметрлерді де беруге болады, бұл үрдіс ОБП-да кең таралған. Келесі мысалда объектілерді сілтеме бойынша әдіске беру қарастырылады.

Мысал 1. Әдістерге сілтеме бойынша объектілер беру мысалы.

```
using System;
class Shenber {
    public int xcoord;
    public int ycoord;
    public double radius;
    public Shenber(int x, int y, int r)
    { xcoord = x;
      ycoord = y;
      radius = r;
    }
    public void Shygaru() { Console.WriteLine("shenber centrinin
      koordinatalary: (" + xcoord + ", " + ycoord + "),
      shenber radiusy = " + radius);
    }
    public bool qiylysu(Shenber ob) {
      double d = Math.Sqrt(Math.Pow(xcoord - ob.xcoord, 2) +
        Math.Pow(ycoord - ob.ycoord, 2));
      if (d < radius + ob.radius) return true;
      return false;
    }
}
class Program {
    static void Main() {
        Shenber shenber1 = new Shenber(2,2,1);
        Shenber shenber2 = new Shenber(6,6,2);
        Console.WriteLine("birinshi shenber parametrleri: ");
        shenber1.Shygaru();
        Console.WriteLine("ekinshi shenber parametrleri: ");
        Shenber2.Shygaru();
        bool b = shenber1.qiylysu(shenber2);
        if(b) Console.WriteLine("berilgen shenberler qiylysady");
        else
            Console.WriteLine("berilgen shenberler qiylyspaidy");
        Console.ReadKey();
    }
}
```

Программа орындалу нәтижесі :

```
birinshi shenber parametrleri:
shenber centrinin koordinatalary: (2, 2), shenber radiusy
= 1
ekinshi shenber parametrleri:
shenber centrinin koordinatalary: (6, 6), shenber radiusy
```

= 2

berilgen shenberler qiylyspaidy

Бұл программадағы **qiylysu()** әдісі аргумент ретінде **Shenber** типіндегі объектіге сілтеме алады. Бұл әдіс шақырған объекті мен **ob** параметрі арқылы берілетін объектінің қиылысатынын тексереді. Егер екі объекті қиылысатын болса, бұл әдіс **true** мәнін қайтарады.

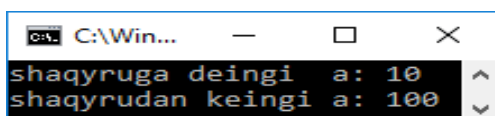
Параметрдің ref модификаторын қолдану

Параметрдің ref модификаторы әдісті мән бойынша емес, мәжбүрлі түрде сілтеме бойынша шақырады. Бұл модификатор әдісті жариялау кезінде де және оны шақырғанда да көрсетіледі.

Келесі программада Sqr() әдісі құрылады, ол оған берілген бүтін аргументтің квадратын қайтарады. Осы мысалдағы ref модификаторының қалай қолданылғанына назар аударыңыздар.

```
// ref модификаторының типті сілтеме бойынша беруі.  
using System;  
class RefTest {  
    // Бұл әдіс ref сөзі арқылы өз аргументін өзгертеді.  
    public void Sqr(ref int i) {  
        i = i * i;  
    }  
}  
  
class RefDemo {  
    static void Main() {  
        RefTest ob = new RefTest(); // ob енді сілтеме бойынша  
  
        int a = 10;  
  
        Console.WriteLine("shaqyruga deingi a: " + a);  
  
        ob.Sqr(ref a); // ref модификаторы қолданылды  
        Console.WriteLine("shaqyrudan keingi a: " + a);  
    }  
}
```

Сонымен, ref модификаторы әдісті жариялаудағы параметр алдында және әдісті шақырғанда да аргумент алдында көрсетіледі екен. Программа нәтижесінде де аргумент мәні Sqr() әдісі арқылы нақты өзгергені көрініп тұр.



```
C:\Win...  
shaqyruga deingi a: 10  
shaqyrudan keingi a: 100
```

Енді ref модификаторы арқылы аргументтердің мәндерін алмастыратын Swap() әдісін жазып көрсетейік.

```
// Мәндердің орнын ауыстыру.  
using System;
```

```

class ValueSwap {
    // Бұл әдіс өз аргументтерінің мәндерін алмастырады.
    public void Swap(ref int a, ref int b) {
        int t; t = a; a = b; b = t;
    }
}
class ValueSwapDemo {
    static void Main() {
        ValueSwap ob = new ValueSwap();// сілтеме бойынша
        int x = 10, y = 20;
        Console.WriteLine("shaqyruga deingi x pen y: " + x + " " + y);
        ob.Swap(ref x, ref y);
        Console.WriteLine("shaqyrudan keingi x pen y: " + x + " " + y);
    }
}

```

Бұл программа нәтижесі

```

C:\Windows\system32\cmd.exe
shaqyruga deingi x pen y: 10 20
shaqyrudan keingi x pen y: 20 10

```

ref модификаторы жайлы мынаны білу керек:

- бұл модификатор арқылы сілтеме бойынша берілетін аргументке *әдіс шақырылғанға дейін* мән берілуі тиіс.
- параметр ретінде мұндай аргумент алатын әдісте параметр белгілі бір мәнге сілтеме жасайды деп есептеледі.
- сонымен, ref модификаторын қолданған кезде әдіс ішінде аргументтің бастапқы мәнін беруге болмайды.

Параметрдің out модификаторын қолдану

Кейде сілтемелі параметр әдіске мән беру үшін емес, одан мән алу үшін қолданылады. Мысалы, белгілі бір функцияны орындап, тек мән қайтаратын әдіс бар делік (желілік сокетті ашып, одан желі операциясының дұрыс немесе қате орындалғаны жайлы сілтемелік параметр түрінде мәлімет алу керек). Мұнда әдіске мәлімет берілмейді, бірақ ол белгілі бір жұмысты орындап, мән қайтаруы тиіс. Бұған дейінгі ref типіндегі параметр әдісті шақыру үшін оған мән беруді талап ететін болатын. Сондықтан, осы айтылған шектеулерді болдырмас үшін, C# тілінде – параметрдің out модификаторын пайдалану керек.

Параметрдің out модификаторы ref түйінді сөзіне ұқсас, тек бір айырмасы бар: ол әдістен мән қайтарып береді, бірақ мән қабылдап алмайды. Сондықтан out параметрі ретінде қолданылатын айнымалыға мән беру қажет емес. Оған қоса, мұндағы әдісте out параметрі *инициалданбайтын болып саналады*. Бұл әдістің қайтаратын параметріне ол *жұмысын аяқтағанша мән берілуі тиіс*. Сонымен, әдіс шақырылған соң, out параметріне белгілі бір мәні берілуі керек.

Келесі программа мысалында нақты санның бүтіні мен бөлшегін жеке-жеке бөліп алу үшін Decompose класындағы GetParts() әдісі пайдаланылады. Берілген санның әрбір бөлігінің әдістен қалай қайтарылатынына назар аударыңыздар.

```

// Параметрдің out модификаторын қолдану.
using System;
class Decompose {

```

```
// Нақты санның бүтіні мен бөлшегін ажыратып алу.
public int GetParts(double n, out double frac) {
    int whole;

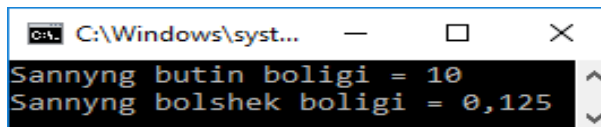
    whole = (int)n;
    frac = n - whole; // frac параметрі санның бөлшегін береді
    return whole;    // санның бүтін бөлігін қайтару
}
}
```

```
class UseOut {
    static void Main() {
        Decompose ob = new Decompose();
        int i; double f;

        i = ob.GetParts(10.125, out f);

        Console.WriteLine("Sannyng butin boligi = " + i);
        Console.WriteLine("Sannyng bolshek boligi = " + f);
    }
}
```

Бұл программа нәтижесі.



```
C:\Windows\syst...
Sannyng butin boligi = 10
Sannyng bolshek boligi = 0,125
```

GetParts() әдісі n санының бүтін бөлігін return операторы арқылы қайтарады. Ал санның бөлшегін out типіндегі frac параметрі арқылы бөліп алады. Мысал көрсеткендей, out модификаторын қолдана отырып, әдістен екі мән қайтарып алдық. Осы тәсілмен әдістерден екі, үш, төрт, т.с.с. мәндер қайтара аламыз, оған шек қойылмайды.

Енді out арқылы екі параметр қолданатын мысал келтірейік. Мұндағы HasComFactor() әдісі екі түрлі функцияны орындайды.

- 1) Ол екі бүтін санның ортақ көбейткішін анықтап (ондайлар бар болса), логикалық true мәнін қайтарады, әйтпесе – false мәнін береді.
- 2) out типіндегі параметрлер арқылы екі санның ең кіші және ең үлкен ортақ бөлгіштерін (ондай сандар бар болса) анықтайды.

// out типіндегі екі параметрді пайдалану.

```
using System;

class Num {
    /* x және y айнымалыларының ортақ көбейткішін анықтау.
    * Егер олар бар болса, солардың ішіндегі ең үлкен
    * және ең кіші көбейткіштерін out типіндегі параметр
    * арқылы анықтап қайтару. */
    public bool HasComFactor(int x, int y, out int least, out int greatest) {
        int max = x < y ? x : y;    int i;
        bool first = true;    least = 1;    greatest = 1;
        // Ең кіші және ең үлкен көбейткіштерді табу.
        for(i=2; i <= max/2 + 1; i++) {
            if( ((y%i)==0) & ((x%i)==0) ) {
                if (first)    { least = i; first = false; }
                greatest = i;
            }
        }
    }
}
```

```

        if(least != 1) return true;
        else return false;
    }
}

class DemoOut {
    static void Main() {
        Num ob = new Num();
        int lcf, gcf;

        if(ob.HasComFactor(231, 105, out lcf, out gcf) {
            Console.WriteLine("231 jane 105 sandarynyng " +
                "eng kishi ortaқ kobeitkishi = " + lcf) ;
            Console.WriteLine("231 jane 105 sandarynyng " +
                "engulkenortaқ kobeitkishi = " + gcf);
        }
        else
            Console.WriteLine("35 jane 49 sandarynyng ortaқ kobeitkishi joq.");
        if(ob.HasComFactor(35, 51, out lcf, out gcf) {
            Console.WriteLine("35 jane51sandarynyng eng kishi " +
                "ortaқ kobeitkishi = " + lcf);
            Console.WriteLine(" 35 jane 51 sandarynyng eng ulken " +
                "ortaқ kobeitkishi = " + gcf);
        }
        else Console.WriteLine(" 35 jane 51sandarynyng ortaқ kobeitkishi joq.");
    }
}

```

Бұл программадағы Main() әдісінде lcf және gcf айнымалыларына мәндер HasComFactor() әдісін шақырғанға дейін меншіктеледі. Егер HasComFactor() әдісінің параметрлері out болмай, gcf болғанда, бұл қате болар еді. Мұндағы әдіс екі бүтін санның ортақ көбейткішінің болуы/болмауына байланысты true не false мәндерінің бірін қайтарады.

Егер ортақ көбейткіш табылса, out типіндегі параметр арқылы осы сандардан ең кіші және ең үлкен ортақ көбейткіштері қайтарылады.

Бұл программаның орындалу нәтижесі:

```

C:\Windows\system32\cmd.exe
231 jane 105 sandarynyng eng kishi ortaқ kobeitkishi = 3
231 jane 105 sandarynyng eng ulken ortaқ kobeitkishi = 21
35 jane 51 sandarynyng ortaқ kobeitkishi joq.

```

Аргументтердің айнымалы санын пайдалану

Әдіс құрған кезде оған берілетін параметрлер саны алдын ала белгілі болады, бірақ кейде олай болмай қалады. Әдіске берілетін аргументтер саны айнымалы болып келетін жағдайлар туындағанда, арнайы типтегі параметр қолданылады. Мысалы, бірсыпыра мәндер ішінен кішісін табатын әдіс құрылған болса, ол берілген мәндер санына қарамай, бәрі бір солардың ең кішісін анықтауы тиіс.

Осындай кездерде қолданылатын арнайы тип params модификаторы арқылы жасалады. Мұнда аргумент ретінде жиым қолданылады да, әдіске берілетін параметрлер саны жиым элементтерінің санымен анықталады. Ал программадағы аргументтерді анықтап алу үшін жиымға қатынас құру, яғни оны пайдалану ісі ұйымдастырылады.

Төмендегі мысалда params модификаторы MinVal() әдісін құруда қолданылып, ол

берілген мәндер ішінен ең кішісін анықтайды.

// params модификаторын пайдалану мысалы.

```
using System;

class Min {
    public int MinVal(params int[] nums) {
        int m;

        if(nums.Length ==0) {
            Console.WriteLine("Kate: argumentter joq.");
            return 0;
        }

        m = nums[0];
        for(int i=1; i < nums.Length; i++)
            if(nums[i] < m) m = nums[i];
        return m;
    }
}
```

```
class ParamsDemo {
    static void Main() {
        Min ob = new Min();
        int min;
        int a = 10, b = 20;

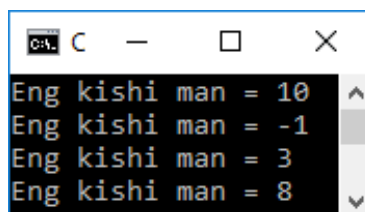
        // Екі мәні бар әдісті шақыру.
        min = ob.MinVal(a, b);
        Console.WriteLine("Eng kishi man = " + min);

        // Үш мәні бар әдісті шақыру.
        min = ob.MinVal(a, b, -1);
        Console.WriteLine("Eng kishi man = " + min);

        // Бес мәні бар әдісті шақыру.
        min = ob.MinVal(18, 23, 3, 14, 25);
        Console.WriteLine("Eng kishi man = " + min);

        // Бүтін сандық жиымнан тұратын әдісті шақыру.
        int[] args = { 45, 67, 34, 9, 112, 8 };
        min = ob.MinVal(args);
        Console.WriteLine("Eng kishi man = " + min);
    }
}
```

Бұл программа нәтижесі:



```
cmd C - □ ×
Eng kishi man = 10
Eng kishi man = -1
Eng kishi man = 3
Eng kishi man = 8
```

Бұл программада MinVal() әдісі шақырылғанда, оған nums жиымынан аргументтер беріледі. Жиым ұзындығы берілетін аргументтер санына тең. Сондықтан MinVal() әдісі арқылы кез келген көлемдегі мәндер арасынан ең кішісін таба аламыз.

MinVal() әдісінің ең соңғы шақырылуына назар салыңыздар. Бұл жолы жеке мәндер орнына көптеген элементтері бар жиым беріледі. Аргументтерді осындай түрде беру дұрыс болып саналады. Params типіндегі параметр құрылғаннан кейін, ол ұзындығы айнымалы аргументтер тізімін немесе жиым элементтерін қабылдайды.

Params типіндегі параметрге кез келген аргументтер санын беруге болады, олардың барлығы да осы параметр көрсетіп тұрған жиым типінде болуы тиіс.

Мысалы, MinVal() әдісін мынадай түрде шақыру қате:

```
min = ob.MinVal(1, 2.2); // Дұрыс емес!
```

өйткені double типін(2.2) автоматты түрде MinVal() әдісінде көрсетілген nums жиымының int типіне айналдыра алмаймыз.

Params модификаторын байқап пайдалану керек, өйткені ол кез келген аргументтер санын, тіпті *нөлді* де қабылдай береді. Мысалы, төмендегідей түрде MinVal() әдісін шақыру синтаксис бойынша дұрыс болғанымен айтылған ережеге қайшы келіп тұр.

```
min = ob.MinVal(); // аргументтер жоқ
```

```
min = ob.MinVal(3); // 1 аргумент
```

Осындай себептерге байланысты MinVal() әдісінде оны пайдалану алдында nums жиымының кем дегенде, бір элементінің болуы тексеріледі. Егер мұндай тексеру болмаса, MinVal() әдісін аргументтерсіз шақырғанда, аластама (ерекше жағдай) туындайтын еді.

MinVal() әдісінде бір ғана элемент те болуы мүмкін. Әдісте қарапайым параметр де, айнымалы параметр де бола береді. Келесі мысалда ShowArgs() әдісі String типіндегі бір параметр қабылдайды және де params типіндегі параметр ретінде бүтін сан түріндегі ды қабылдайды.

```
// params типіндегі ұзындығы айнымалы параметрмен бірге  
// қарапайым параметрді де қолдану.
```

```
using System;
```

```
class MyClass {
```

```
    public void ShowArgs(String msg, params int[] nums) {  
        Console.WriteLine(msg + " ");
```

```
        foreach (int i in nums)  
            Console.WriteLine(i + " ");  
        Console.WriteLine();  
    }  
}
```

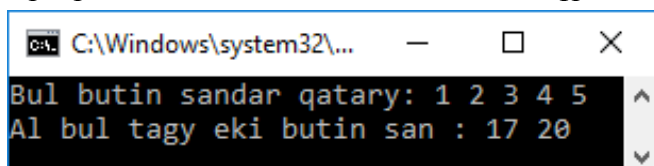
```
class ParamsDemo2 {
```

```
    static void Main() {  
        MyClass ob = new MyClass(); // конструктор
```

```
        ob.ShowArgs("Bul butin sandar qatary", 1, 2, 3, 4, 5);
```

```
        ob.ShowArgs("Al bul tagy eki butin san ", 17, 20);  
    }  
}
```

Бұл программаның нәтижесі төмендегідей түрде болады.



```
C:\Windows\system32\...  
Bul butin sandar qatary: 1 2 3 4 5  
Al bul tagy eki butin san : 17 20
```

Әдісте қарапайым параметрлермен бірге `params` типіндегі ұзындығы айнымалы параметр бар болса, ол параметрлер тізімінде ең соңғы болып көрсетілуі керек. Бірақ кез келген жағдайда `params` типіндегі параметр жалғыз болуы тиіс.