4 дәріс. ref және out модификаторлары.

Дәрістің мақсаты: студенттерде ref және out модификаторларын пайдалану ерекшеліктері бойынша түсініктерін көрсетуге қабілет қалыптастыру.

Осы дәрісті меңгеру нәтижесінде студенттер келесі қабілеттерге ие болады:

- ref және out модификаторларының қызметі бойынша түсініктерін көрсету;
- ref және out модификаторларын пайдалану жағдайларын ажырату.

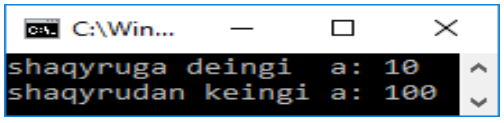
Параметрдің ref модификаторын қолдану

Параметрдің refмодификаторы 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.

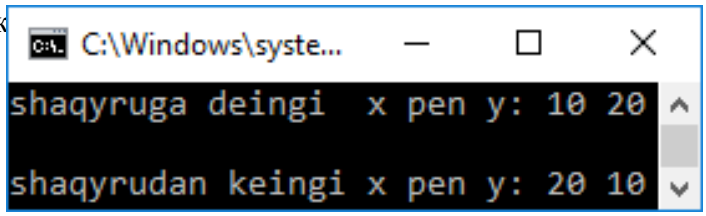


Енді ref модификаторы арқылы аргументтердің мәндерін алмастыратын Swap() әдісін жазып көрсетейік.

// 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 нәтиж



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 немес 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.

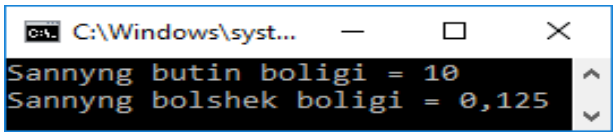
Параметрдің out модификаторы ref түйінді сөзіне ұқсас, тек бір айырмасы бар: ол әдістен мән қайтарып береді, бірақ мән қабылдап алмайды. Сондықтан out параметрі ретінде қолданылатын айнымалыға мән беру қажет емес. Оған қоса, мұндағы әдісте out параметрі *инициалданбайтын болып саналады*. Бұл әдістің 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.



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

Бұл программадағы Main() әдісінде lcf және gcf айнымалыларына мәндер HasComFactor() әдісін шақырғанға дейін меншіктеледі. Егер HasComFactor() әдісінің параметрлері out болмай, ref 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 сандардаң ең кіші және ең үлкен ортақ көбейткіштері қайтарылады.

Бұл программаның орындалу нәтижесі: