

8-ші дәріс. Кездейсоқ сандарды пайдалану. Символдар және сөз тіркестері

- Кездейсоқ сандар тізбегін алу;

Жиымдарды пайдаланатын программаларда олардың мәні ретінде кездейсоқ сандарды қолданған ыңғайлы болып табылады.

C# кітапханасында кездейсоқ сандар беретін **Random** класы бар, ол **System** атаулар кеңістігінде анықталған.

Кездейсоқ сандар тізбегін алу үшін алдымен төмендегідей конструктор көмегімен класс экземплярын (данасын) жасау керек:

```
Random a = new Random(); // 1
```

```
Random b = new Random( 1 ); // 2
```

мұндағы 1-жолдағы параметрсіз конструктор кездейсоқ сандарды уақыт көрсеткіші негізінде қрастырады. Сондықтан ол қайталаған сайын жаңа сандар тізбегін беріп отырады. Ал, 2-жолдағы **int** типіндегі параметрі бар конструктор қайталаған сайын тұрақты сандар тізбегін береді.

Кездейсоқ сандар тізбегін алу үшін алдымен төмендегідей конструктор көмегімен класс экземплярын (данасын) жасау керек:

```
Random a = new Random(); // 1
```

```
Random b = new Random( 1 ); // 2
```

мұндағы 1-жолдағы параметрсіз конструктор кездейсоқ сандарды уақыт көрсеткіші негізінде қрастырады. Сондықтан ол қайталаған сайын жаңа сандар тізбегін беріп отырады. Ал, 2-жолдағы **int** типіндегі параметрі бар конструктор қайталаған сайын тұрақты сандар тізбегін береді.

Аты	Сипаттамасы
Next ()	int типіндегі оң сандар диапазонынан бүтін оң сандар тізбегін қайтарады.
Next(макс)	[0, макс] аралығындағы (диапазонынанда-ғы) бүтін оң сандар тізбегін қайтарады.
Next(мин, макс)	[мин, макс] диапазонынан бүтін оң сандар тізбегін қайтарады.
NextBytes(жиым)	[0, 255] диапазонынандағы сандар тізбегін қайтарады.
NextDouble()	[0, 1] диапазонынандағы нақты оң сандар сандар тізбегін қайтарады.

Кездейсоқ сандар тізбегін алу тәсілдерін пайдалану мысалы
using System;

```
namespace Listing6_10
```

```
{ class Class1
```

```
{static void Main()
```

```
{Random a = new Random();
```

```
Random b = new Random(1);
```

```
const int n = 10;
```

```
Console.WriteLine("\n Диапазон [0, 1]:");
```

```
for (int i = 0; i < n; ++i)
```

```
Console.Write("{0 ,6:0.##}“,a.NextDouble());
```

```

        Console.WriteLine("\n Диапазон [0, 1000]:");
        for (int i = 0; i < n; ++i)
            Console.Write(" " + b.Next(1000));
    Console.WriteLine("\n    Диапазон [-10, 10]:");
    for (int i = 0; i < n; ++i)
        Console.Write(" " + a.Next(-10, 10));
    Console.WriteLine("\n    Массив [0, 255]:");
    byte[] mas = new byte[n]; a.NextBytes(mas);
    for (int i = 0; i < n; ++i)
        Console.Write(" " + mas[i]);
}
}}

```

```

C:\WINDOWS\system32\cmd.exe
Диапазон [0, 1]:
0,2 0,26 0,48 0,48 0,97 0,45 0,46 0,7 0,97 0,05
Диапазон [0, 1000]:
248 110 467 771 657 432 354 943 101 642
Диапазон [-10, 10]:
1 8 -4 0 -6 -1 -8 -10 -9 -5
Массив [0, 255]:
241 189 6 89 107 158 42 8 206 41
Для продолжения нажмите любую клавишу . . .

```

Программа жұмысының нәтижесі:

```

Диапазон [0, 1]:
0,02 0,4 0,24 0,55 0,92 0,84 0,9 0,78 0,78 0,74
Диапазон [0, 1000]:
248 110 467 771 657 432 354 943 101 642
Диапазон [-10, 10]:
-8 9 -6 -10 7 49 -5 -2 -1
Массив [0, 255]:
181 105 60 50 70 77 9 28 133 150

```

1. Символдар мен сөз тіркестері

Мәтіндік мәліметтерді өңдеу программа-лаудағы ең бір жиі кездесетін жұмыс түрі болып саналады. Оның мынадай мүмкіндік-тері бар: символдарды, символдар жиымын, сөз тіркестерін, өзгермелі және өзгермейтін тіркестерді өңдеу, т.с.с.

1.1. Символдарды пайдалану

Символдық тип **char** Unicode кодтау жүйе-сінде символдарды сақтау үшін керек. Сим-волдық тип C# тілінің құрамдас типі болып саналады да, **.NET** кітапханасының **System** атаулар кеңістігіндегі **Char** стандартты класы-на сәйкес келеді. Бұл класс символдарды анықтап, олардың категориясын, төменгі немесе жоғарғы регистрге ауыстыруға қатысты статикалық тәсілдерді де береді.

Төменде **System.Char** класының негізгі тәсілдері (функциялары) келтірілген

Тәсіл	Сипаттамасы
-------	-------------

GetNumericValue	егер символ цифр болса, оның сандық мәнін, қалған жағдайда -1 мәнін қайтарады.
GetUnicodeCategory	Unicode-символының категориясын қайтарады.
IsControl	егер символ басқару символы болса, true мәнін қайтарады.
IsDigit	егер символ ондық цифр болса, true мәнін қайтарады.
IsLetter	егер символ әріп болса, true мәнін қайтарады.
IsLetterOrDigit	егер символ әріп немесе цифр болса, true мәнін қайтарады.
IsLower	егер символ төменгі регистрде берілсе, true мәнін қайтарады.
IsNumber	егер символ сан (ондық не онақтылық) болса, true мәнін қайтарады.
IsPunctuation	егер символ тыныс белгісі болса, true мәнін қайтарады.
IsSeparator	егер символ ажырату таңбасы (разделитель) болса, true мәнін қайтарады.
IsUpper	егер символ жоғарғы регистрде берілсе, true мәнін қайтарады.
IsWhiteSpace	егер символ босорын белгісі (босорын, келесі жолға көшу, каретканы қайтару) болса, true мәнін қайтарады.
Parse	Тіркесті символға айналдырады (тіркес бір символдан тұруы тиіс)
ToLower	Символды төменгі регистрге түрлендіреді
ToUpper	Символды жоғарғы регистрге түрлендіреді
MaxValue, MinValue	Максимал және минимал кодтары бар символдарды (бұлар экранда көрінбейтін символдар) қайтарады

Енді мысалдар келтірейік

```

System.Char класы тәсілдерін пайдалану
using System;
namespace Listing6_6
{ class Class1
    { static void Main()
        { try
            {
                char b = 'B', c = '\x63', d = '\u0032';    // 1
                Console.WriteLine("{0} {1} {2}", b, c, d);
                Console.WriteLine("{0} {1} {2}", char.ToLower(b),
char.ToUpper(c), char.GetNumericValue(d));
                char a;

                do
                    { Console.Write("Введите символ: ");
                    a = char.Parse(Console.ReadLine());
                    Console.WriteLine("Введен символ {0}, его код - {1}", a,
(int)a);

                    if (char.IsLetter(a)) Console.WriteLine("Буква");
                    if (char.IsUpper(a)) Console.WriteLine("Верхний рег.");
                    if (char.IsLower(a)) Console.WriteLine("Нижний рег.");

```

// 2

```

        if (char.IsControl(a)) Console.WriteLine("Управляющий");
        if (char.IsNumber(a)) Console.WriteLine("Число");
        if (char.IsPunctuation(a)) Console.WriteLine("Разделитель");
    } while (a != 'q');
}
catch
{
    Console.WriteLine("Возникло исключение");
    return;
}
}
}
}
}

```

Мұндағы 1-операторда 3 символдық айнымалы сипатталған. Олар әр түрлі бейнелеу формаларын-дағы символдармен инициалданған. Сонан соң символдарды шығару және түрлендіру әрекеттері орындалады.

2-операторда пернелерден енгізілген символ талданады. Мұнда Ctrl пернесі мен латын әріптерін пайдаланып, басқару символдарын да енгізуге болады. Енгізу кезінде тіркесті бір ғана символ болуға тиіс char символына түрлендіретін Parse тәсілі пайдаланылған. Онда тіркес енгізілетіндіктен, символ соңынан Enter пернесі басылады. Цикл q символы енгізілгенше орындала береді.

Олар әр түрлі бейнелеу формаларындағы символдармен инициалданған. Соңынан символдарды шығару және түрлендіру әрекеттері орындалады.

Символды шығару ондық түрдегі оның кодымен беріледі. Кодтарды экранға шығару үшін олар бүтін типке түрлендіріледі. C# тілінде символдарды тіркеске және кері өздігінен тікелей түрлендіру орындалмайды, ол үшін оларды арнайы әрекеттер арқылы түрлендіру керек, мысалы:

```
string s = 'к' + 'о' + 'т'; // нәтижесі "кот"
```

Орыс әріптерінің Unicode ортасында басқа кодталу аймағында орналасатынына назар аударыңдар.

1.2. Символдар жиымдары

Символдар жиымдары басқа типтегі массивтер тәрізді бұрын айтылған базалық Array класы негізінде құрылады.

Оның тәсілдерін мұнда қолдану бірсыпыра есептерді тиімді түрде шығару мүмкіндігін береді.

Енді бір мысал келтірейік.

Символдар жиымын пайдалану мысалы

```

using System;
namespace Listing 6_7
{ class Class1
{ static void Main( )
{
    char [ ] a = { 'm', 'a', 's', 's', 'i', 'v' }; // 1
    char [ ] b = "a роза упала на лапу
азора".ToCharArray(); // 2
    PrintArray( "Исходный массив a:", a );
    int pos = Array.IndexOf( a, 'm' );
    a[pos] = 'M';

```

```

        PrintArray( "Измененный массив a:", a );
        PrintArray( "Исходный массив b:", b );
        Array.Reverse( b );
        PrintArray( "Измененный массив b:", b );
    }
    public static void PrintArray( string header, Array a )
    {
        Console.WriteLine( header );
        foreach ( object x in a ) Console.Write( x );
        Console.WriteLine( "\n" );
    }
}

```

Программа жұмысы нәтижесі:

Символдық жиымды оның мәндерін бере отырып (1-оператор) немесе string класының ToCharArray функциясын пайдалану жолымен тіркесті символ-дарға бөле отырып инициалдауға болады.

```

C:\WINDOWS\system32\cmd.exe
Исходный массив a:
massiv

Измененный массив a:
Massiv

Исходный массив b:
а роза упала на лапу азора

Измененный массив b:
ароза упал ан алапу азор а

Для продолжения нажмите любую клавишу . . .

```