

МАССИВТЕР

5.1 Сілтемелік тип туралы мағлұмат

C# тілінде айнымалылардың екі типі бар – мәнді және сілтемелік. Мәнді типтегі айнымалылар сәйкес типте деректер мәнін сақтауға қабілетті. Сілтемелік айнымалылар компьютер жадының физикалық адресін сақтайды, олар арқылы компьютер жадында деректердің сәйкес мәндері орналасады.

Деректерді сақтау үшін мәнді типтегі айнымалыларға жады әдетте бағдарлама компиляциясы кезінде, ал сілтемелік типті айнымалыларға `new` операторының көмегімен бағдарлама орындалу барысында бөлінеді.

Біз бұдан бұрынғы бағдарламаларда сілтемелік типтегі айнымалыны - кездейсоқ сандарды дайындайтын `Random` класының (`rnd` объектісі) айнымалысын қолданғанбыз.

Айта кету керек сілтемелік тип әдетте құрылымы күрделі деректерді, яғни массивтер, жолдар, кластар типіндегі деректерді ұйымдастыру үшін қолданылады. Барлық осы деректер үйінді (куча) деп аталатын компьютердің арнайы жадында сақталады. Мәнді типтегі барлық айнымалылар бағдарламалық стекте орналасады, ал сілтемелік типті айнымалылар - операциялық жүйе басқаратын үйіндіде.

Сондықтан сілтемелік типті айнымалыларға қарағанда мәнді типтегі айнымалыларға қатынау (қол жеткізу) үшін біраз уақыт жұмсалады, бірақ мәнді типті айнымалыларға қарағанда сілтемелік типті айнымалыларға жады көлемі бір деңгейге көбірек бөлінеді.

Кейде мәнді типтегі айнымалылар үшін компьютер жадын үлестіруді жадының статикадық үлестіруі деп атайды, яғни компьютер жадын бөлу үдерісі бағдарлама компиляциясы кезінде орындалады. Ал сілтемелік типті айнымалыларға компьютер жадын үлестіруді жадының динамикалық үлестіру деп атайды, яғни компьютер жадын бөлу үдерісі бағдарламаның орындалуы барысында орын алады.

Ескере кету керек, сілтемелік типті айнымалыларға компьютер жадын бөлу `new` операторы арқылы орындалады.

5.2 Массив ұғымы

Массив дегеніміз – бір айнымалы арқылы бір типтегі айнымалылар тобын сипаттау формасы. Массивке біріктірілетін барлық айнымалылар 0-ден n -ге дейін нөмірленеді және әрбір айнымалыға өз нөмірі - индекс сәйкес қойылған. Нақты бір массив айнымалысын қолдану үшін массив аты мен айнымалы индексін көрсету керек. Сондықтан массив айнымалысын индексті айнымалы деп жиі атайды.

Бір өлшемді массивтер (векторлар) болады, мысалы, атаулар массиві, бір емтихан бағаларының массиві, туған күндер массиві және т.б.

Екі өлшемді массивтер (матрицалар немесе кестелер) болады, мысалы, футбол бойынша чемпионат ойынының нәтижелері, студенттердің емтихандар нәтижелері бойынша қорытынды кестесі және т.б.

Көп өлшемді массивтерге барлық қалған массивтер жатады.

Бағдарламада массивті сипаттау екі кезеңнен тұрады: массивті жариялау және массивтің жұмысын бастау.

массивті жариялауда айнымалы типі мен аты анықталады, мысалы:

```
int[ ] masi;
```

```
double[ ] masf; ,
```

мұнда бірінші жолда біз аты `masi` болатын бүтін айнымалылар массивін қолданамыз, ал екінші жолда аты `masf` болатын нақты айнымалылар массивін жариялаймыз.

Бұл орайда бағдарлама компиляциясы барысында бағдарлама стегіне сілтемелік типті массивтердің аттары жазылады. Массив айнымалыларына компьютер жадысы массив жұмысын сипаттағаннан бастап бөлінеді, мысалы,

```
masi = new int[10];
```

```
masf = new double[20] ; ,
```

мұнда квадрат жақша ішіндегі мәндер массив өлшемін анықтайды (0 элементтер саны -1 – массив элементтерінде қолдануға болатын индекстер мәндерінің ауқымы).

Массивті инициализациялау кезеңінде үйіндіде массив объектісі құрылады және оның барлық айнымалыларына «нөлдік» мәндер меншіктеледі (оның инициализациясы орындалады).

Айнымалылардың «нөлдік» мәндері сәйкес:

– сандық айнымалылар үшін нөл болады;

– жолдық айнымалылар үшін бос жолдар болады;

– символдық айнымалыларда символ болмайды.

Массивті инициализациялаудан кейін оның айнымалыларын бағдарламада қолдануға болады.

Массивтерді инициализациялау мен жариялаудың әр түрлі нұсқаларын түрлі есептерді шешу мысалдарында қарастырайық.

5.3 Әр түрлі есептерде массивтерді қолдану мысалдары

Массивті қолдануды қажет ететін көптеген есептер бар. Мәтін символдар массиві болғандықтан есептердің бір бөлігі мәтінді өңдеуге байланысты болады.

Математикалық немесе физикалық есептер, мысалы, эксперименттік мәндер массивімен көрсетілген функцияның орта мәнін есептеу.

Информатикада, мысалы, аттардың тізімін алфавит бойынша реттеу немесе массивте іздеу есептері, т.б.

Массивті қолдануды қажет ететін кейбір есептердің шешімін қарастырайық.

5.1-есеп. Эксперименттік функция 12-кестедегі мәндер арқылы берілген. Осы функцияның барлық максимумын табу керек. Есептеулерді жүргізу үшін функцияның келесі мәндерін қолдану керек: 1.05, 3.17, 5.24, 4.38, 6.42, 4.93, 6.59, 7.84, 5.73, 5.14, 4.87 және 3.18.

Есепті шешу алгоритмі келесі қадамдардан тұрады:

– функцияның бірінші мәнін максимумға тексеру – егер функцияның екінші мәні бірінші мәнінен кіші болса, онда функцияның бірінші мәні максимум болып есептеледі;

– максимумдарды іздестіру - егер функцияның алдыңғы және кейінгі мәндері ағымдағы мәнінен кіші болса, онда функцияның ағымдағы мәні максимум болып есептеледі;

– функцияның соңғы мәнін максимумға тексеру - егер функцияның соңғы мәні соңғының алдында тұрған мәнінен үлкен болса, онда функцияның соңғы мәні максимум болып есептеледі;

Бұл алгоритмде функцияның бірнеше бірдей мәндерінің болу жағдайын қарастырмаймыз. Өмірде мұндай жағдайдың болу мүмкіндігі аз. Есепте осындай жағдайлар орын алатын болса, шешімнің алгоритмін өз бетімен құра аласыз.

Бағдарламаның коды:

```
using System;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            int i;
            double j, n;
            double[] mas = {1.05, 3.17, 5.24, 4.38, 6.42, 4.93,
                6.59, 7.84, 5.73, 7.14, 3.87, 2.18 };
            //әдістемелік тұрғыдан мына түрде жазу дұрыс
            //float[] mas;
            //mas = new float[12] { 1.05, 3.17, 5.24, 4.38, 6.42,
            // 4.93, 6.59, 7.84, 5.73, 6.14, 4.87, 3.18 };
            //массивті жариялауды және инициализациялауды бірге
қатар қолдануға болады, мысалы:
            int[] max = new int[7];
            //максимумдар индекстерінің массиві
            // массивті "графикалық" түрде көрсетейік
            for (i = 0; i <= 11; i++)
            {
                Console.Write(i);
                n = Math.Round(mas[i]);
                j = 0;
                while (j <= n) { Console.Write(" "); j++; }
                Console.WriteLine('*');
            }
            int k; // максимум индексі
            k = 0;
            // функцияның бірінші мәнін максимумға тексеру
            if (mas[0] > mas[1]) { max[k] = 0; k++; }
```

```

//максимумдерді іздеу
for (i = 1; i < 11; i++)
if ((mas[i - 1] < mas[i]) && (mas[i] > mas[i + 1]))
{ max[k] = i; k++; }
//функцияның соңғы мәнін максимумге тексеру
if (mas[10] < mas[11]) { max[k] = 11; k++; }
//функция максимумдерін экранға шығару
Console.WriteLine("Funkzia Maksimumi:");
for (i = 0; i < k; i++)
Console.WriteLine("{0} - {1}", max[i], mas[max[i]]);
Console.ReadLine();
}
}
}

```

Бағдарламаның жұмысы:

```

0 *
1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *

```

Funkzia Maksimumi:

```

2 - 5,24
4 - 6,42
7 - 7,84
9 - 7,14

```

Келтірілген бағдарламада массивтерді жариялау мен инициализациялаудың бірнеше нұсқасы қарастырылған.

Айнымалылардың аз саны үшін массивтерді жариялау мен инициализациялауды қатарынан орындауға болады, әсіресе егер массив элементтерінің мәндері бағдарламада анықталмаса - массив элементтері массив константаларымен ұсынылса. Мысалы:

```

double[] mas = { 1.05, 3.17, 5.24, 4.38, 6.42, 4.93, 6.59, 7.84, 5.73, 6.14, 4.87, 3.18
};

```

Массивті жариялау мен инициализациялаудың екінші нұсқасы бірінші массив типіндегі айнымалыны жариялауды, ал содан кейін инициализациялауды талап етеді.

Мысалы:

```
float[] mas;  
mas = new float[12] { 1.05, 3.17, 5.24, 4.38, 6.42, 4.93, 6.59, 7.84, 5.73, 6.14, 4.87,  
3.18 };
```

Үшінші нұсқада массивті жариялау мен инициализациялау бір әрекетте (амалда) орындалады. Мысалы:

```
int[] max = new int[7];
```

Келесі есеп ақпараттық есептер тобына жатады.

5.2-есеп. 20 студенттен тұратын топ емтихан тапсырды. Бағалар 2-ден 5-ке дейін кездейсоқ түрде құрылады. 5, 4, 3 және 2 бағаларын алған студенттердің санын анықтау керек.

Есепті шешу алгоритмі келесі қадамдардан тұрады:

–циклде 1-ден 20-ға дейін кездейсоқ түрде 20 студентке емтихан бағаларының массивін құрамыз;

– бағалар массивін қайта қарау барысында 5, 4, 3 және 2 санын есептейміз;

– монитор экранына бағдарлама жұмысының нәтижесін шығарамыз.

Бағдарламаның коды:

```
using System;  
namespace ConsoleApplication1  
{  
    class Program  
    {  
        static void Main()  
        {  
            int i, j;  
            int[] a = new int[20];  
            int[] b = new int[6];  
            Random rnd = new Random();  
            // 20 студенттің бағасын кездейсоқ түрде құрамыз  
            // монитор экранына шығарамыз  
            Console.WriteLine("Studentterdin bagalari: ");  
            for (i=0;i<=19;i++)  
            {  
                a[i] = rnd.Next()%4 + 2;  
                Console.WriteLine(" {0}", a[i]);  
            }  
            Console.WriteLine();  
            //5, 4, 3 және 2 сандарын есептейміз, нәтижелерді b  
            массивіне жазамыз  
            for (i=0;i<=19;i++)  
            {  
                if (a[i]==2) b[2]++;  
                if (a[i]==3) b[3]++;  
                if (a[i]==4) b[4]++;  
                if (a[i]==5) b[5]++;  
            }  
        }  
    }  
}
```

```
// бағдарлама жұмысын экранға шығарамыз
for (i=2;i<=5;i++)
    Console.WriteLine(" {0} bagasin {1} student aldi",i,
b[i]);
    Console.ReadLine();
}
}
```

Бағдарлама жұмысы:

Studentterdin bagalari: 5 2 2 5 2 5 4 2 2 3 5 4 4 3 5 3 4 3 2 2

2 bagasin 7 student aldi

3 bagasin 4 student aldi

4 bagasin 4 student aldi

5 bagasin 5 student aldi

шарттары тексерілетін for циклінің операторын келесі үзіндімен ауыстыруға болады:

```
for (i=0;i<=19;i++)
    for (j = 2; j <= 5; j++)
        if (a[i] == j) b[j]++;
```

немесе

```
for (i=0;i<=19;i++)
    b[a[i]]++;
```

Кездейсоқ санды құрғаннан кейін бірден барлақ тексерулерді бір циклде жазуға болады:

```
for (i=0;i<=19;i++)
{
    a[i] = rnd.Next()%4 + 2;
    b[a[i]]++;
    Console.Write(" {0}", a[i]);
}
```

Осы есептің шешімі бойынша алгоритмді орындаудың бірнеше нұсқа қарастырылды, бірақ алгоритмді орындаудың соңғы нұсқалары айқын емес.

Осы есептің берілген кодын қысқартып жазу өте қызықты, бірақ осы пәннің дәрістерінде ол қарастылмайды.

Оқулықта біз есепті шешу алгоритмінің бағдарламалық кодтарының ішінен ең қысқа нұсқасының орнына алгоритмнің жүзеге асырылуы айқын нұсқасын қарастырамыз.

Бағдарлама кодының соңғы нұсқасында b массивінің индекс мәні студенттің емтиханда алған бағасымен анықталады.

Егер массив индексі күрделі алгоритм немесе кездейсоқ құрылатын болса, онда массивті осы түрдегі қолдану ассоциативті қолдану деп аталады.

Математикада массивтер полином коэффициенттерін сипаттағанда қолданылады.

Мысалы, $P_n(x)$ полиномының n -і дәрежелі коэффициенті әдетте мына функция түрінде жазылады

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (5)$$

$a[n+1]$ массивімен көрсетуге болады, онда массив индексі 0-ден n -ге дейін ауысады.

математиканың міндеттерінің бірі – x айнымалысының мәндерінің белгілі бір үзіндісінде n -і дәрежелі полином түрінде көрсетілген функция өзгерісін зерттеуі.

5.3-есеп. 10-дәрежелі полином коэффициенттерінің массиві берілген.

($n = 10$) – 1.05, 3.17, 5.24, 4.38, 6.42, 4.93, 6.59, 7.84, 5.73, 7.14, 3.87, 2.18.

Индексі 0 болатын массивтің мәні полиномның нөлінші коэффициентіне тең және т.б.

Полиномның тәртібін 0-ден 1-ге дейін 0.1 қадамымен зерттеу (x айнымалысы 0-ден 1-ге дейін 0.1 қадамымен өзгереді).

Монитор экранына полином өзгерісін кесте және график түрлерінде шығарыңыз.

Бағдарламаның коды:

```
using System;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            double j, n, x, y, p;
            int i, k;
            double[] mas = {1.05, 3.17, 5.24, 4.38, 6.42, 4.93,
                6.59, 7.84, 5.73, 7.14, 3.87, 2.18 };
            double[] masf = new double[11]; //полином мәндерінің
массиві
            Console.WriteLine("Polinomdi          grafikalik          tyrde
korsety:");
            k = 0;
            for (x = 0; x < 1; x = x + 0.1)
            {
                y = x; p = mas[0];
                for (i = 1; i <= 11; i++)
                {
                    p = p + y * mas[i];
                    y = y * x;
                }
                masf[k] = p; k++;
                Console.Write(x);
                n = Math.Round(p);
                j = 0;
                while (j <= n) { Console.Write(" "); j++; }
                Console.WriteLine('*');
            }
        }
    }
}
```

```

    }
    Console.WriteLine("Polinomdi keste tyrinde korsety:");
    i = 0;
    for (x = 0; x < 1; x = x + 0.1)
    {
        Console.WriteLine("x = {0} P = {1}", x, masf[i]);
        i++;
    }
    Console.ReadLine();
}
}
}

```

Бағдарламаның жұмысы:

Polinomdi grafikalik tyrde korsety:

```

0 *
0,1 *
0,2 *
0,3 *
0,4 *
0,5 *
0,6 *
0,7 *
0,8 *
0,9 *
1 *

```

Polinomdi keste tyrinde korsety:

```

x = 0 P = 1,05
x = 0,1 P = 1,4244787388488
x = 0,2 P = 1,9410304774144
x = 0,3 P = 2,6619038136876
x = 0,4 P = 3,6975170516992
x = 0,5 P = 5,253203125
x = 0,6 P = 7,7263077961728
x = 0,7 P = 11,9008032176404
x = 0,8 P = 19,3213982427136
x = 0,9 P = 32,9819936534232
x = 1 P = 58,54

```

Полиномды есептеу алгоритмінде n дәрежедегі x айнымалысын рекуррентті есептеу қолданылады. Ол үшін y айнымалысына x мәні меншіктелген және 1 индексті қосылғыш полиномды есептеу (вычисление слагаемого полинома с индексом 1) орындалған. Одан кейін y x -ке көбейтіліп (x^2 мәнін аламыз), екінші қосылғыштың мәні есептеледі, т.б..

5.4 Динамикалық массивтер

C# тілінде массивтер элементтерінің мәндері динамикалық болып табылады, яғни оларға жады бағдарлама жұмысының үдерісінде new операциясының көмегімен «үйіндіде» бөлінеді. C# тілінің осындай ерекшелігі бағдарламаларда динамикалық массивтер құруға мүмкіндік берді, яғни олардың элементтер саны бағдарлама жұмысының үдерісінде анықталатын массивтер. Көптеген есептердің тобы бар, онда массив өлшемі бағдарлама жұмысының үдерісінде анықталады немесе пайдаланушы диалогында берілуі тиіс, мысалы, каталогтағы файлдар немесе қоймадағы тауарлар саны.

Таза синтаксис жағынан статикалық және динамикалық массивтерді жариялауда маңызды айырмашылық жоқ. Шынында массивтерді жариялауда сандық сипаттамалар болмайды, мысалы, `double[] mas;`, ал массивті инициализациялау бағдарлама жұмысы уақытында new операциясы арқылы орындалады. Егер диалогта массив элементтер саны берілсе, динамикалық массивті жариялау мен инициализациялауды бірге орындауға болады. Мысалы:

```
Console.WriteLine("masi massivinin elementterin engiz ");
int size = int.Parse(Console.ReadLine());
int[] masi = new int[size];
```

C# тілінде динамикалық массивтер болып бір өлшемді массивтер ғана болады. Динамикалық массивтер элементтерінің санына сай келетін айнымалылардың мәні оларды инициализациялауға дейін анықталуы керек.

5.4-есеп. Топ студенттері «C# тілінде бағдарламалау» пәні бойынша емтихан тапсырады. Топтағы студенттер саны диалог режимінде беріледі (20 көп емес). Бағалар (балл түрінде анықталады) 20-дан 100 дейінгі аралықта кездейсоқ түрде құрылады. Топтың емтихан бағасын динамикалық массивте ұйымдастыру керек. Емтихан нәтижесін шығарыңыз.

Бағдарламаның коды:

```
using System;
using System.Collections.Generic;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("20-dan          aspaitin          noptagi
studentterdin sanin engiziniz");
            int size = int.Parse(Console.ReadLine());
            int[] masi = new int[size];
            int i;
            Random rnd = new Random();
            for (i = 0; i < size; i++)
                Console.Write(" {0,3}", i + 1);
```

```

Console.WriteLine();
for (i = 0; i < size; i++)
{
masi[i] = rnd.Next() % 81 + 20;
Console.Write(" {0,3}", masi[i]);
}
Console.WriteLine();
Console.WriteLine("Enter pernesin basiniz");
Console.ReadLine();
}
}
}

```

Бағдарлама жұмысы:

20-dan aspaitin noptagi studentterdin sanin engiziniz

15

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

76 72 28 63 72 90 95 58 39 38 60 56 87 92 91

Enter pernesin basiniz

Массивтің әрбір мәнін шығарғанда оның позициясының саны көрсетіледі – {0, 3}, ол бағдарлама жұмысының нәтижесін ыңғайлы түрде көрсетуге мүмкіндік береді.

5.5 Өзін-өзі тексеру сұрақтары

1 Мәнді типтегі айнымалы туралы түсінік.

2 Сілтемелік типтегі айнымалы туралы түсінік.

3 Массивте қандай айнымалыларды орналастыруға болады ?

4 Бүтін типті айнымалылар үшін бір өлшемді массивті жариялаудың дұрыс нұсқасын қалай жазуға болады?

5 Бүтін типті 10 айнымалы үшін бір өлшемді массивті инициализациялаудың дұрыс нұсқасын қалай жазуға болады?

6 Бағдарламаның келесі үзіндісі нені орындайды:

```
for (I = 0; I <= 10; I++)
```

```
Console.Write(" {0} \t", a[I]); ?
```

7 a[11] массиві үшін бағдарламаның келесі үзіндісі нені орындайды :

```
for (I = 1; I <= 10; I++)
```

```
{k = a[I]; a[I] = a[11 - I]; a[11 - I] = k;} ?
```

8 Бағдарламаның келесі үзіндісі нені орындайды :

```
k = a[0];
```

```
for (I = 0; I <= 10; I++)
```

```
if (a[I] > k) k = a[I];
```

```
Console.WriteLine(" {0} ", k); ?
```

9 Бағдарламаның келесі үзіндісі нені орындайды:

```
for (I = 0; I <= 9; I++)
```

{ k = a[I]; a[I] = a[I+1]; a[I+1] = k; } ?

10 Бағдарламаның келесі үзіндісі нені орындайды:

I = 0;

while (I <= 10)

{

a[I] = a[I] * (-1);

I++;

}?

