

4 Массивтермен жұмыс жасағанда нұсқағыштарды қолдану

4.1 Деректер массивы бойынша нұсқағышты жылжыту үшін қосу және алу операцияларын қолдану

Нұсқағыштарға кейбір арифметикалық операциялар қолданылуы мүмкін, бірақ оларды пайдалану деректерді компьютер жадысында орналастыру бойынша жақсы білімдерді қажет етеді. Өйткені нұсқағыштың бір бірлікке ұлғаюы нұсқағыш жататын типті келесі деректің мекенжайына ауысуды білдіреді.

Қарапайымдылық үшін деректер ретінде **char** типіндегі айнымалылардың массивын қолданамыз. Оларға массивтың инициализациялауы кезінде латын әліпбиінің алғашқы 10 әріпін жазамыз. Массив бойынша нұсқағышты жылжытып, жеке әріптерді баса отырып, біз қандай да бір сөзді баса аламыз, мысалы, «BENEFIC».

Бағдарламаның бастапқы коды:

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static unsafe void Main()
        {
            char[] masc = new char[]
                { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J' };
            fixed (char* Mptr = masc)
            {
                char* ptr = Mptr;
                ptr++;
                Console.WriteLine(*ptr);
                ptr = ptr + 3;
                Console.WriteLine(*ptr);
                ptr += 3;
                Console.WriteLine(*ptr);
                ptr -= 3;
                Console.WriteLine(*ptr);
                ptr++;
                Console.WriteLine(*ptr);
                ptr += 3;
                Console.WriteLine(*ptr);
                ptr -= 6;
                Console.WriteLine(*ptr);
            }
            Console.WriteLine();
            Console.ReadKey();
        }
    }
}
```

Работа программы:

BENEFIC

Бағдарламада `fixed` операторы пайдаланылған, ол келесі жазылу форматына ие:

```
fixed ([type][*] [имя указателя] = [&][имя объекта])  
{ тіркелген объектімен әрекеттер (действия с фиксированным объектом) }
```

«**fixed** операторы, қоқысты тазалаушы орнын ауыстырмауы (жылжытпауы) үшін мекенжайы нұсқағышқа енгізілетін объектіні бекітіп алады, осылай нұсқағыш дұрыс болып қалады. Бекіту дөңгелек жақшалардан соң жазылған блокты орындау уақытында өтеді.» [Павловская, б.352].

Мәселе қоқыс тазалаушы үнемі жұмыс істеуде болады және шоғыр жадыны оңтайландырумен шұғылданады - шоғыр жадысын оңтайландыру мақсатында жұмысын аяқтаған объектілерді жою және жұмыс істеп тұрған объектілерді жылжытумен шұғылданады.

Егерде біздің объект осындай оңтайландырудың нәтижесінде жадының басқа облысына ауысса, онда оның мекенжайлары өзгереді және ескі мекенжайлармен нұсқағыштарды пайдалану өз мәнін жояды. Іс жүзінде **fixed** операторы қоқыс тазалаушыға көрсетілген объектіні шоғырға жылжытуға болмайтындығын ескертеді.

Mptr нұсқағыштың көмегімен символдар массивының бастапқы мекенжайын бекітіп алып, біз массив элементтері тізімінің тақырыбын құрған боламыз. Массив элементтерінің тізімі бойынша жылжу үшін тағы бір нұсқағыш – инициализациялау кезінде тақырыпқа орнатылатын **ptr** ағымдағы нұсқағышы қажет. **Fixed** операторы блогының ішінде тақырыпты өзгертуге жасалған қарекеттер бағдарламаның компиляциясының қатесіне әкеп соғады, өйткені деректердің бекітілген облысының тақырыбын өзгерту деректердің жоғалуына әкеп соғу мүмкін еді.

Нұсқағыштарға арифметикалық операцияларды пайдалану компьютер жадысындағы деректердің мекенжайларына сәйкес келетін бүтін мәндермен ғана әрекеттер жасауға рұқсат береді. Мұнда нұсқағыштармен адресталатын деректер типінің көлемі автоматты түрде реттеледі. Элементтерінің мәндері компьютер жадысында реттімен орналастырылған массивтармен жұмыс кезінде арифметикалық операцияларды пайдалану ұсынылады.

4.2 Жадыда деректерді орналастыру үшін **stackalloc** операциясын пайдалану

C# тілінде **new** операциясының көмегімен нұсқағыштардың деректеріне жадыны бөлу қолданылмайды. Нұсқағыштар көмегімен адресталатын деректерге жадыны бөлудің негізгі тәсілі **stackalloc** операциясы, әрі мұнда жады стекте бөлінеді.

`stackalloc` операциясын жазу форматы:

тип* нұсқағыш атауы= **stackalloc** тип [саны];

Оқу бағдарламасы ретінде тұтас типті 10 айнымалы үшін жадының бөлуді және оларды таңдау әдісімен сұрыптауды қарастырамыз.

```
using System;
```

```

namespace ConsoleApplication1
{
    class Program
    {
        static unsafe void Main()
        {
            int i, j, b;
            int* masptr = stackalloc int[10];
            // Проверка инициализации массива
            for (i = 0; i < 10; i++)
            {
                Console.WriteLine(" {0}", masptr[i]);
            }
            Console.WriteLine();
            Random rnd = new Random();
            // формирование и печать массива
            Console.WriteLine("Массив до сортировки: ");
            for (i = 0; i < 10; i++)
            {
                masptr[i] = rnd.Next() % 101 - 50;
                Console.WriteLine(" {0}", masptr[i]);
            }
            Console.WriteLine();
            // сортировка элементов массива методом выбора
            for (i = 0; i < 9; i++)
                for (j = i + 1; j < 10; j++)
                    if (masptr[i] < masptr[j])
                        { b = masptr[i]; masptr[i] = masptr[j]; masptr[j] = b; }
            // печать массива после сортировки
            Console.WriteLine("Массив после сортировки: ");
            for (i = 0; i < 10; i++)
                Console.WriteLine(" {0}", masptr[i]);
            Console.WriteLine();
            Console.ReadKey();
        }
    }
}

```

Работа программы:

0 0 0 0 0 0 0 0 0 0

Массив до сортировки: 23 -27 -35 33 -31 -31 23 4 42 40

Массив после сортировки: 42 40 33 23 23 4 -27 -31 -31 -35

Егерде жадыны бөлу үшін нұсқағыш пайдаланығанын білмесек, онда бағдарламада `masptr` деген қызық атаумен қарапайым массив пайдаланған деп болжауға болады.

Массивқа нұсқағыш арқылы жүгінген кезде орта массив шекарасының бұзылуын бақыламайды және де индексті тым өзгерте отырып, массив үшін бөлінген жадының шекарасынан «шығу» және кейбір деректерді немесе бағдарламаны бұзу қауіпі бар. Массивқа нұсқағыштың көмегімен жүгіну кезіндегі екінші ыңғайсыздық нұсқағыштармен жұмыс істемейтін `Length` қасиетінің көмегімен массив ұзындығын анықтай алмау болып табылады.

Стекте жадыны бөлу кезінде массивтың барлық элементтеріне нөлдік мәндер беріледі – деректердің алғашқы инициализациясы орындалады.

Stackalloc операциясының көмегімен жадыны жеке айнымалыларға бөліп беруге болады – бір элементті массив сияқты. Бағдарламаның келесі фрагментінде екі нұсқағыштың пайдаланылуы көрсетілген, олар үшін жады стекте **stackalloc** операциясының көмегімен бөлінген болатын.

```
int* Aptr = stackalloc int[1];
int* Bptr = stackalloc int[1];
. . .
Aptr[0] = 5;
Bptr[0] = Aptr[0];
Console.WriteLine("Bptr[0] =" + Bptr[0]);
```

При работе фрагмента будет напечатано Bptr[0] = 5

Стекте жадыны босату **stackalloc** операциясы орналасқан блоктың жұмысы аяқталғанда автоматты түрде жүргізіледі – біздің жағдайымызда бұл бағдарлама.

4.3 Нұсқағыштар массивын пайдалану

Нұсқағыштар, қарпайым айнымалылар сияқты, массивта сақталынуы мүмкін. Нұсқағыштар массивын сипаттау келесі форматқа ие:

тип* [] нұсқағыштар _массивының_ атауы = new тип*[элементтер_саны];

Мысалы, int* [] masiPtr = new int*[10];

Нұсқағыштар массивын пайдаланудың оқу мысалы ретінде минус 50-ден 50-ге дейін диапазонында 10 кездейсоқ санды қалыптастырылуын, олардың жадыға, ал олардың мекенжайларының нұсқағыштар массивына жазылуын қарастырамыз.

Бағдарламаның бастапқы коды:

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static unsafe void Main()
        {
            int i, j, b;
            int*[] masiPtr = new int*[10];
            Random rnd = new Random();
            // формирование массива
            for (i = 0; i < 10; i++)
            {
                int* aPtr = stackalloc int[1];
                aPtr[0] = rnd.Next() % 101 - 50;
                masiPtr[i] = &aPtr[0];
            }
            // печать массива
            Console.Write("Массив :   ");
            for (i = 0; i < 10; i++)
                Console.Write(" {0}", *masiPtr[i]);
        }
    }
}
```

```

    Console.WriteLine();
    Console.ReadKey();
}
}
}

```

Работа программы:

Массив : -8 -43 31 19 23 31 -36 -46 -2 -21

Массив элементінің әр мәні үшін жадының бөлінуі `int* aPtr = stackalloc int[1]`; операциясының көмегімен жүргізіледі, ал массив элементтерінің мәндері жадысының мекенжайлары нұсқағыштардың массивына жазылады.

4.4 Құрылымдармен жұмыс кезіндегі нұсқағыштарды пайдалану

Нұсқағыштардың көмегімен құрылымдармен жұмыс кезінде оқу мысалы ретінде жазбалар стегін құрамыз, ал қандай да бір бүтін типті айнымалы және стектің келесі элементіне нұсқағыш олардың өрістері болады (ұқсас жазбаға нұсқағыш).

```

using System;

namespace ConsoleApplication1
{
    class Program
    {
        unsafe struct zveno
        {
            int nom;
            zveno* next;
            static unsafe void Main()
            {
                zveno zap = new zveno();
                zveno* tek1, tek2;
                zveno* zagol;
                zagol = &zap;
                tek2 = zagol;
                zagol->nom = 10;
                zagol->next = null;
                for (int i = 1; i < 5; i++)
                {
                    zveno* zap1 = stackalloc zveno[1];
                    tek1 = &zap1[0];
                    tek2->next = tek1;
                    tek1->next = null;
                    tek1->nom = i;
                    tek2 = tek1;
                }
                tek1 = zagol;
                while (tek1 != null)
                {
                    Console.Write(" " + tek1->nom);
                }
            }
        }
    }
}

```

```
        tek1 = tek1->next;
    }
    Console.WriteLine();
    Console.ReadKey();
}
}
}
```

Работа программы:

10 1 2 3 4