

МАССИВТЕР

Сілтемелік тип туралы мағлұмат

C# тілінде айнымалылардың екі типі бар:

- Мәнді. Мәнді типтегі айнымалылар сәйкес типте деректер мәнін сақтайды .
- Сілтемелік. Сілтемелік айнымалылар компьютер жадының физикалық адресін сақтайды, физикалық адрес бойынша компьютер жадында деректердің сәйкес мәндері орналасады.

Деректерді сақтау үшін мәнді типтегі айнымалыларға жады бағдарлама компиляциясы кезінде орындалады.

Кейде мәнді типтегі айнымалылар үшін компьютер жадын үлестіруді *жадының статикадық үлестіруі* деп атайды.

Мәнді типтегі барлық айнымалылар *бағдарламалық стекте* орналасады

МАССИВТЕР

Сілтемелік тип туралы мағлұмат

Сілтемелік типті айнымалыларға жады new операторының көмегімен *бағдарлама орындалу барысында* бөлінеді.

Сілтемелік типті айнымалыларға компьютер жадысын үлестіруді – жадының динамикалық үлестіруі деп аталады.

Сілтемелік тип әдетте құрылымы күрделі деректерді ұйымдастыру үшін қолданылады:

- массивтер,
- жолдар,
- кластар типіндегі деректер.

Барлық осы деректер *үйінді (куча)* деп аталатын компьютердің арнайы жадында сақталады. Сілтемелік типті айнымалылар - операциялық жүйе басқаратын үйіндіде орналасады.

.

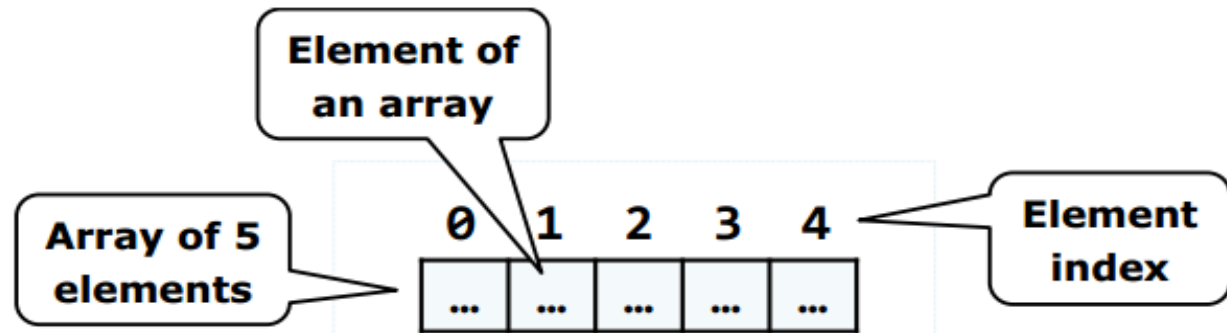
МАССИВТЕР

Массив ұғымы

Массив дегеніміз – бір айнымалы арқылы бір типтегі айнымалылар тобын сипаттау формасы.

Массив - деректер жинағын сақтау үшін пайдаланылатын ең негізгі тип.

Массив элементтерді қамтиды және оларға тік жақшалар [] арқылы индекс сілтеме жасайды.



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 **атауын** және **айнымалы индексін** көрсету керек. Сондықтан массив айнымалысын **индексті айнымалы** деп жиі атайды.

МАССИВТЕР

Объявление массива

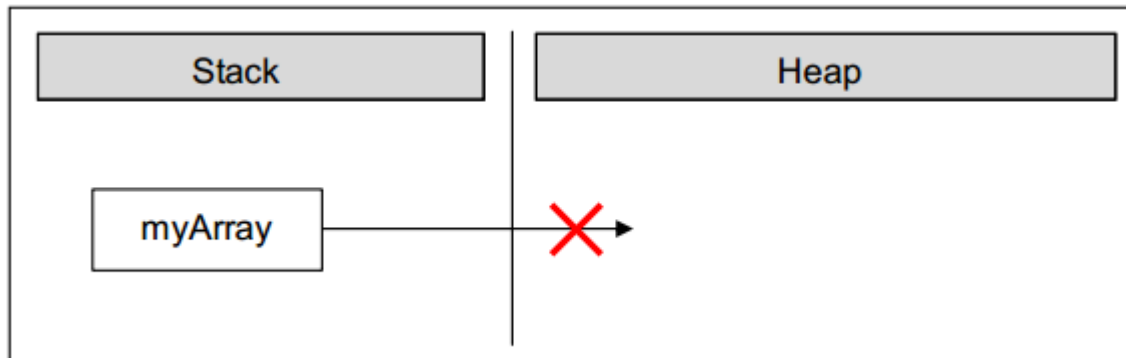
Мы объявляем массив в C # следующим образом:

```
int[] myArray;
```

Когда мы объявляем переменную типа массива, она является ссылкой, которая не имеет значения (она указывает на null).

Это связано с тем, что память для элементов еще не выделена.

На рисунке ниже показано, как выглядит объявленная переменная массива, когда память для элементов массива еще не выделена:



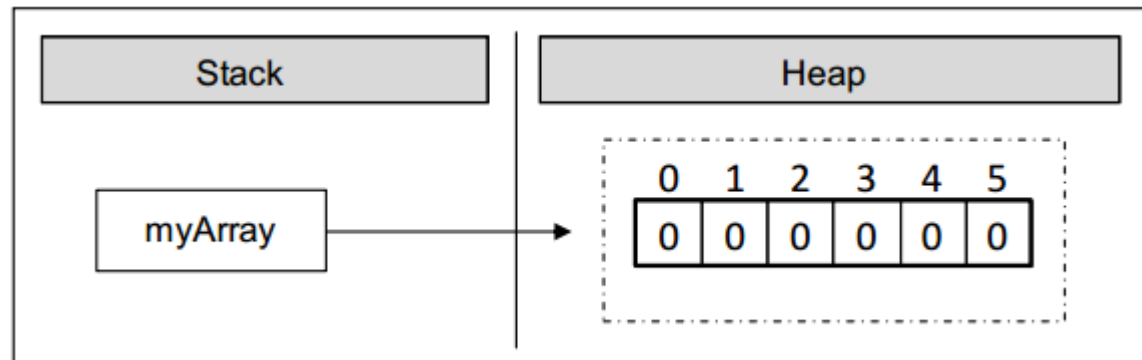
В стеке выполнения программы переменная с именем myArray и его значение установлено равным нулю (это означает, что он не имеет значения).

Создание массива - оператор «new»

В C# мы создаем массив с помощью ключевого слова **new**, которое используется для выделения памяти:

```
int[] myArray = new int[6];
```

В этом примере мы выделяем массив длиной 6 элементов типа **int**. Это означает, что в **динамической памяти (куче)** выделяется область из 6 целых чисел, и они



На рисунке показано, что после выделения памяти для массива переменная `myArray` указывает на адрес в динамической памяти, где находятся значения. В C# элементы массива всегда хранятся в динамической памяти (называемой также кучей).

Во время выделения памяти для массива мы устанавливаем общее число элементов в скобках (неотрицательное целое число), определяя его длину.

Тип элементов записывается после зарезервированного слова `new`, поэтому мы указываем, какие типы элементов будут выделены в памяти.

Келесі мысал бүтін сандардың бір өлшемді массивін жасайды:

```
int[] mySet = new int[5];  
mySet[0] = 1;  
mySet[1] = 2;  
mySet[2] = 3;  
mySet[3] = 4;  
mySet[4] = 5;
```

Мысалда, массивтің бірінші элемент индексі **0**-ге, ал екінші элемент индексі **1**-ге тең. Соңғы элемент индексі **4**-ке тең.

МАССИВТЕР

Массив ұғымы

Бағдарламада массивті сипаттау екі кезеңнен тұрады:

- **массивті жариялау**
- **массив инициализациясы.**

Массивті жариялауда айнымалы типі мен аты анықталады, мысалы:

int[] mas1; - атауы **mas1** болатын бүтін айнымалылар массивін жариялау,

double[] mas2; - атауы **mas2** болатын нақты айнымалылар массивін жариялау.

Бұл орайда бағдарлама компиляциясы барысында бағдарлама стегінде мәндері “нөл” **mas1** және **mas2** массивтердің аттаулары жазылады.

Массив айнымалыларына үйіндіде (куча) компьютер жадысы бөлінеді, мысалы, ол массив инициализациясынан бастап басталаы.

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

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

МАССИВТЕР

Массив ұғымы

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 (пустые строки) болады;
- символдық айнымалыларда - символ болмайды (отсутствие символа).

Массивті инициализациялаудан кейін оның айнымалыларын бағдарламада қолдануға болады.

МАССИВТЕР

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.

Синтаксис жағынан статикалық және динамикалық массивтерді жариялауда маңызды айырмашылық жоқ.

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:

МАССИВТЕР

Динамикалық массивтер

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.

Жол элементтері сақталатын массивін жариялау

```
string[] stringArray = new string[6];
```

Массив инициализациясы

```
string[] weekDays = { "Sun", "Mon", "Tue", "Wed",  
"Thu", "Fri", "Sat" };
```

```
int[] array1 = new int[] { 1, 3, 5, 7, 9 };
```

МАССИВТЕР

1) Для небольшого количества переменных можно совмещать объявление и инициализацию массивов, особенно если значения элементов массива определяются не в программе – элементы массива представлены константами массива. Например:

```
float[] 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 };
```

2) Второй вариант **объявления** и **инициализации** массива требует **сначала объявление переменной** типа массив, а затем его **инициализацию**. Например:

```
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 };
```

```
int[] array3;  
array3 = new int[] { 1, 3, 5, 7, 9 };
```

Можно объявить переменную массива без инициализации, но при присвоении массива этой переменной нужно использовать оператор **new**.

МАССИВТЕР

3) В третьем варианте объявление и инициализация массива выполняются в одном действии. Например:

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

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

Деректерді өңдеудің көптеген алгоритмдерінің ішінен келесі алгоритмдері ерекшелеуге болады:

- **сұрыптау /сортировки** (деректерді кейбір сипаты бойынша реттеу)
- **іздеу/поиска** (деректерде берілген сипат бойынша элементті анықтау).

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

Массив элементтерін сұрыптау

Массив элементтерін сұрыптау – массив элементтерін белгілі бір сипаты бойынша реттеу. Мысалы, массив элементтері бүтін немесе нақты типтері үшін өсу немесе кему бойынша. Массив элементтері жолдық (строковые) типі үшін массив элементтерін алфавит бойынша орналастыру.

Біз массив элементтерін сұрыптаудың тек екі әдісінің алгоритмдерін және бағдарламалақ жүзеге асырылуын қарастырамыз.

- 1) Таңдау әдісі («метод выбора»)
- 2) «Көпіршікті» әдіс («пузырьковый» метод сортировки)

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

Таңдау әдісімен сұрыптау алгоритмі.

Массив элементтерін таңдау әдісімен, кему тәртібінде сұрыптау алгоритмі келесі амалдардың орындалуын болжайды.

Массивтің **бірінші элементін** қарастырамыз, оның мәнін массивтің қалған басқа элементтерімен кезекпен салыстырамыз (2-ден соңғы элементке дейін).

Егер мәні үлкен элемент кезіксе, онда ол бірінші элементпен орын ауыстырады.

Массивті бірінші рет тексерудің нәтижесінде **бірінші элемент ең жоғары мәніне ие болады.**

Содан кейін екінші элементті қарастырамыз, оның мәнін массивтің қалған басқа элементтерімен кезекпен салыстырамыз (3-ден соңғы элементке дейін).

Массивті екінші рет тексерудің нәтижесінде біз сұрыпталған массивтің екінші элементінің мәнін анықтаймыз. Тексеру массивтегі соңғының алдында тұрған элементіне дейін осылай қайталанады, соңғы тексеруде массивтің алдыңғы элементі массивтің соңғы элементімен ғана салыстырылады.

Массив элементтерінің саны N-ге тең болса, алгоритмде орындалатын салыстыру операцияларының саны (О.С.) келесі түрде есептеледі:

$$K.O. = (N-1)+(N-2)+(N-3)+ \dots + 2 + 1 = N \cdot (N-1) / 2 \quad (7)$$

МАССИВТЕРДІ СҰРЫПТАУ АЛГОРИТМДЕРІ

```
static void Main()
{
    int i,j,b;
    int[] a = new int[11];
    Random rnd = new Random();

    // формирование и печать массива
    Console.WriteLine("Массив до сортировки: ");
    for (i = 0; i <= 10; i++)
    {
        a[i] = rnd.Next()%101 - 50;
        Console.WriteLine(" {0}", a[i]);
    }
    Console.WriteLine();
    // сортировка элементов массива методом выбора
    for (i = 0; i <= 9; i++)
        for (j = i + 1; j <= 10; j++)
            if (a[i] < a[j])
                { b = a[i]; a[i] = a[j]; a[j] = b; }
    // печать массива после сортировки
    Console.WriteLine("Массив после сортировки: ");
    for (i=0; i<=10; i++)
        Console.WriteLine(" {0}", a[i]);
    Console.WriteLine();
    Console.ReadLine();
}
```

массив элементтерін кему тәртібінде
сұрыптау алгоритмі

Работа программы:

Массив до сортировки: -15 -24 -28 46 -38 22 -27 -49 -43 22 -27

Массив после сортировки: 46 22 22 -15 -24 -27 -27 -28 -38 -43 -49

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

«Көпіршікті» әдіспен сұрыптау алгоритмі.

Массив элементтерін *өсу тәртібінде* «көпіршікті» әдіспен сұрыптау алгоритмі келесі әрекеттердің орындалуын қарастырады.

1) Массивтің бірінші элементін екіншісімен салыстырамыз және егер біріншісі үлкен болса, онда олар орындарын алмастырады. Одан кейін массивтің екінші элементі үшіншісімен салыстырылады, егер екінші элемент үлкен болса, онда олар орындарын алмастырады және т.с.

Массивті бірінші рет «қарап шығудың» нәтижесінде массив элементінің ең үлкен мәні ең соңына жазылады.

2) Массивтің бірінші элементін қайтадан аламыз және оны екінші элементпен салыстырамыз – барлық процесті массивтің соңғы тұрған элементтің алдындағы элементке дейін (до предпоследнего элемента) қайталаймыз - оның мәнін құрамыз.

3) Соңғы салыстыру операциясына дейін осылай қайталанады (массивтің 1 және 2 элементтері арасында).

Сонымен массив элементтерінің мәні өсу ретімен сұрыпталады.

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

«Көпіршікті» әдіспен сұрыптау алгоритмі.

«Көпіршікті» және таңдау әдістері алгоритмдерінің есептеу тиімділіктері бірдей.

Массив элементтерін кему тәртібінде «Көпіршікті» әдіспен сұрыптау алгоритмінің бағдарламалық жүзеге асырылуын қарастырайық.

Бағдарламаның коды:

МАССИВТЕРДІ СҰРЫПТАУ АЛГОРИТМДЕРІ

«Көпіршікті» әдіспен сұрыптау алгоритмі.

```
static void Main()
{
    int i,j,b;
    int[] a = new int[11];
    Random rnd = new Random();

    // формирование и печать массива
    Console.WriteLine("Массив до сортировки: ");
    for (i = 0; i <= 10; i++)
    {
        a[i] = rnd.Next()%101 - 50;
        Console.WriteLine(" {0}", a[i]);
    }
    Console.WriteLine();
    // сортировка элементов массива «пузырьковым» методом
    for (i=0; i<=9; i++)
        for (j=0; j<=9-i; j++)
            if (a[j]<a[j+1])
                { b=a[j];a[j]=a[j+1];a[j+1]=b;}
    // печать массива после сортировки
    Console.WriteLine("Массив после сортировки: ");
    for (i=0; i<=10; i++)
        Console.WriteLine(" {0}", a[i]);
    Console.WriteLine();
    Console.ReadLine();
}
```

```
Массив до сортировки:  -40 45 -29 30 11 -8 -26 -37 9 23 -41
Массив после сортировки:  45 30 23 11 9 -8 -26 -29 -37 -40 -41
```

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

Түрлі іздеу әдістерінің ішінен олардың тек үшеуінің алгоритмдері мен бағдарламалық жүзеге асырылуын қарастырамыз – массивте элемент мәні бойынша іздеу әдістері:

- **екілік**
- **блоктық**
- **тізбектеп**

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

Тізбектеп іздеу алгоритмі.

Элементтің тізбектеп іздеу алгоритмі ізделетін кілтті массивтің барлық элементімен, яғни бірінші элементінен бастап соңғы элементіне дейін тізбекті түрде салыстыруға негізделген. Егер массивтің соңғы элементі қаралып қойса, онда іздеу аяқталады.

Осы алгоритмді көрсету үшін оны кітаптың керекті бетін іздеу алгоритмімен салыстырады, мысалы, үш жүзінші бет. Әрбір бет бірінші беттен үш жүзінші бетке дейін тізбекті түрде парақталады.

Осы алгоритмнің кемшілігі – массив элементтерінің саны көп болған жағдайда іздеуге айтарлықтай көп уақыт жұмсалады. Іздеу уақыты N санына пропорционал, мұндағы N - массив элементтерінің саны.

Алгоритм артықшылығы - массивте элементтерді кез келген тәртіпте және тәртіпсіз түрде орналастыруға рұқсат беру.

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

Есеп. Іздестіру массиві 0-ден 99-дейінгі аралықтағы 20 кездейсоқ бүтін сандарынан тұрады. Диалог режимінде кез келген бүтін сан – **ізделінетін кілт беріледі. Осы сан (индекстері қандай) іздестіру массивінде неше рет кездесетінін табу керек.** Тізбектеп іздеу алгоритмін қолдану.

МАССИВТЕРДІ СҰРЫШТАУ АЛГОРИТМДЕРІ

```
static void Main()
{
    int i,k,n; int[] a = new int[20];
    int[] p = new int[20]; Random rnd = new Random();
    string buf;
    // формирование и печать массива
    Console.WriteLine("Массив поиска: ");
    for (i=0;i<10;i++)      Console.Write(" {0}", i);
    for (i=10;i<20;i++)    Console.Write(" {0}", i);
    Console.WriteLine();
    for (i = 0; i < 20; i++)
    {
        a[i] = rnd.Next()%100;
        if (a[i]>9) Console.Write(" {0}", a[i]);
        else Console.Write(" {0}", a[i]);
    }
    Console.WriteLine();
    Console.WriteLine("Введите ключ поиска");
    buf = Console.ReadLine();
    k = Convert.ToInt32(buf);    n=0;
    // Поиск элементов массива, соответствующих ключу поиска.
    for (i=0;i<20;i++)
    if (k == a[i]) {p[n]=i; n++;}
    if (n==0)
        Console.WriteLine("Элемента, соответствующего ключу поиска нет!!!");
    else
    {Console.WriteLine("Количество элементов, соответствующих ключу поиска = {0}",n);
    Console.WriteLine("Найденные индексы: ");
    for (i=0;i<n;i++)
    Console.Write(" {0}",p[i]);
    Console.WriteLine();
    }
    Console.ReadLine();
}
```