

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Мысал, 4 жолдан, 2 бағанадан тұратын кі өлшемді массивті құру.

```
int[,] array = new int[4, 2];
```

Следующее объявление создает трехмерный массив с количеством элементов 4, 2 и 3.

```
int[ , , ] array1 = new int[4, 2, 3];
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Мысал. [Rank қасиеті](#) массив өлшемін көрсетеді.

```
class TestArraysClass
{ static void Main()
  {
    // Declare and initialize an array:
    int[,] theArray = new int[5, 10];
    System.Console.WriteLine("The array has {0} dimensions.",
theArray.Rank);
  }
}
// Output: The array has 2 dimensions.
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Массивті инициализациялау.

1) Массив можно инициализировать при объявлении, как показано в следующем примере.

```
// Two-dimensional array.
```

```
int[,] array2D = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, {  
7, 8 } };
```

```
// The same array with dimensions specified.
```

```
int[,] array2Da = new int[4, 2] { { 1, 2 }, { 3, 4 }, { 5, 6 },  
{ 7, 8 } };
```

```
// A similar array with string elements.
```

```
string[,] array2Db = new string[3, 2] { { "one", "two" }, {  
"three", "four" }, { "five", "six" } };
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

```
// Accessing array elements.  
System.Console.WriteLine(array2D[0, 0]);  
System.Console.WriteLine(array2D[0, 1]);  
System.Console.WriteLine(array2D[1, 0]);  
System.Console.WriteLine(array2D[1, 1]);  
System.Console.WriteLine(array2D[3, 0]);  
System.Console.WriteLine(array2Db[1, 0]);
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

2) Можно также инициализировать массив, не указывая его размерность.

```
int[,] array4 = { { 1, 2 }, { 3, 4 }, { 5, 6 },  
{ 7, 8 } };
```

3) Если нужно создать переменную массива без инициализации, то необходимо использовать оператор **new**, чтобы присвоить массив переменной. В следующем примере показано использование **new**.

```
int[,] array5; array5 = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6  
, { 7, 8 } }; // ОК - дұрыс
```

```
//array5 = {{1,2}, {3,4}, {5,6}, {7,8}}; // Error - қате
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Следующий пример присваивает значение конкретному элементу массива.

```
array5[2, 1] = 25;
```

4) В следующем примере кода элементы массивов инициализируются значениями по умолчанию (за исключением массивов массивов).

```
int[,] array6 = new int[10, 10];
```



## Көп өлшемді массивтер

Анықтама бойынша массивтер бір типтегі айнымалыларды және бір өлшемді массивтерді біріктіреді.

Біріктірілген массивтер **көп өлшемді** массивтер деп аталады.

Егер біріктірілген массивтердің өлшемдері бірдей болса, онда ондай көп өлшемді массивтер **«тік төртбұрышты»** массивтер деп аталады.

Егер біріктірілген массивтердің өлшемдері бірдей болмаса, онда ондай көп өлшемді массивтер **«сынық»** (jagged) массивтер деп аталады.



## Көп өлшемді массивтер

**Сынық массивті** жариялаған кезде екінші индекс өлшемі көрсетілмейді, индекстер квадрат жақшаға алынады (үтір қолданылмайды), мысалы:

```
int[][] mtovar = new int[5][];
```

«Бағаналар» саны әр сынық массив «жолдарды» үшін динамикалық түрде анықталады:

```
j = rnd.Next() % 21 + 10;
```

```
mtovar[i] = new int[j];
```

Сынық массив элементтеріне әрбір индексті квадрат жақшаға орналастыру керек, мысалы:

```
mtovar[i][j] = rnd.Next() % 61 + 10;
```

## Көп өлшемді массивтер

**Есеп.** Кәсіпорын иесінің 5 сауда орыны бар. Әрбір сауда орынында түрлі тауарлар саны 10 мен 30 арасындағы кездейсоқ сан болады. Әрбір тауардың құны 10 мен 70 ш.б. аралығындағы кездейсоқ сан болады. 5 сауда орынында өткізілетін тауар санын және олардың жалпы құнын табу керек.

```
static void Main()
{
    int[][] mtovar = new int[5][];
    int k = 0;    int j,kol,cto;    string buf;
    Random rnd = new Random();
    //Сынық массивті жариялау
    for (int i = 0; i < 5; i++)
    {
        j = rnd.Next() % 21 + 10; mtovar[i] = new int[j];
    }
    Console.WriteLine("Matrisa kurildi!!");
}
```

## Көп өлшемді массивтер

```
kol = 0; cto = 0;
// Сынық массивті құру және экранға шығару
for (int i = 0; i < 5; i++)
{
    Console.WriteLine("{0}: ", mtovar[i].Length);
    kol = kol + mtovar[i].Length;
    for ( j = 0; j < mtovar[i].Length; j++)
    {
        mtovar[i][j] = rnd.Next() % 61 + 10;
        cto = cto + mtovar[i][j];
        Console.WriteLine(mtovar[i][j]+" ");
    }
    Console.WriteLine();
}
Console.WriteLine();
Console.WriteLine("Tayar sani = {0}", kol);
Console.WriteLine("Olardin bagasi = {0}", cto);
Console.WriteLine("Enter pernesin basiniz");
Console.ReadLine();
}
```

## Көп өлшемді массивтер

```
Matrisa kurildi!!
```

```
16: 62 69 27 29 64 17 35 20 19 14 50 69 31 31 60 26
```

```
22: 38 64 56 68 25 55 37 25 43 69 24 23 65 22 65 38 17 67 65 43 15 70
```

```
17: 30 63 53 52 49 11 45 19 63 66 33 28 69 19 42 14 65
```

```
23: 14 42 70 57 66 64 22 26 65 17 19 59 35 34 27 21 39 41 64 25 44 16 33
```

```
10: 48 31 16 26 58 44 21 56 67 13
```

```
Tayar sani = 88
```

```
Olardin bagasi = 3618
```

```
Enter pernesin basiniz
```

# C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

## Array массивтер класы

Айнымалыларды массивке біріктіру түрлі алгоритмдерде және деректерді өңдеу жүйелерінде қолданылады, мысалы, реляциялық деректер базасында немесе тізімдермен берілген массивтерде сұрыптау мен іздеу алгоритмдерінде. Сондықтан көптеген визуалды бағдарламалау орталарында массивтермен жұмыс істеуге арналған арнайы кластар бар

C# тілінде массивтермен жұмыс жасау үшін `System.Array` класы қолданылады. Осы класта массивтермен жұмыс істеуге арналған статикалық қасиеттер мен әдістер бар. Ең жиі қолданылатын қасиеттердің бірі - массивте элементтер санын есептейтін **`Length`** типіндегі қасиет. Мысалы, `masi` массивінің элементтер саны **`masi.Length`** арқылы анықталады.

Бүтін типтегі **`Rank`** қасиет – массив өлшемін анықтауға мүмкіндік береді. `System.Array` класының ең жиі қолданылатын әдістері келесі кестеде көрсетілген.

# C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

## System.Array класының ең жиі қолданылатын әдістері

Әдіс	Сипаттама
Static int <b>BinarySearch</b> (Array, object, IComparer);	Екілік іздеу әдісі. Сұрыпталған бір өлшемді Array массивінде IComparer интерфейсі арқылы object элементін іздейді және элемент индексін қайтарады, егер элемент табылмаса теріс санды қайтарады.
public static void <b>CopyTo</b> (Array, Index);	Ағымдағы бір өлшемді массивтен барлық элементтерді Array массивіне Index индексінен бастап көшіреміз.
public static Array <b>CreateInstance</b> (Type ElementsType, int[] Lengths, int[] LowerBounds);	Әрбір өлшемі бойынша элементтер саны Lengths және Индекстердің төменгі шегі LowerBounds болатын, ElementsType типіндегі элементтерден тұратын көп өлшемді массивті құрайды. Қайта жүктелетін әдістер индекстері 0-ден басталады бір өлшемді және екі өлшемді массивтерді құруға мүмкіндік береді.
public int <b>GetLowerBound</b> (Dimension);	Dimension өлшемі бойынша индекстің ең кіші мәнін қайтарады.
public int <b>GetUpperBound</b> (Dimension);	Dimension өлшемінде индекстің ең үлкен мәнін қайтарады.

# C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

## System.Array класының ең жиі қолданылатын әдістері

public static void <b>Reverse</b> (Array);	Бір өлшемді Array массивінің элементтер тәртібін кері орналастырады.
public static void <b>Sort</b> (Array);	Бір өлшемді Array массивін сұрыптайды
public static void <b>Clear</b> (Array, Index, Length);	Массивті тазарту. Бір өлшемді Array массивінде Length элементтерін орналастырамыз. Элементтер типіне қарай Index элементінен бастап мәндері 0, false немесе null болады.
public int <b>GetLenght</b> (Dimension);	Dimension өлшемі бойынша массив элементтерінің санын қайтарады.

## C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

`Array.GetLowerBound (Int32), GetUpperBound(0)` әдістері

```
// // Бір өлшемді бүтін санды массивті құру
int[] integers = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };

// Массивтің жоғарғы және төменгі жиектерін алыңыз.
int upper = integers.GetUpperBound(0);
int lower = integers.GetLowerBound(0);
Console.WriteLine("Elements from index {0} to {1}:", lower,
upper);
    Console.ReadLine();
```

**Нәтиже**

```
Elements from index 0 to 9:
```



# C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

## **Array.CreateInstance (Type, Int32)**

- Type типіндегі және берілген ұзындықта бірөлшемді Array массивін құрады , индексация 0-ден басталады.

Мысалы,

```
Array my1DArray = Array.CreateInstance(typeof(Int32), 5);
```

## Array классының әдістерін программада қолдану

**1 есеп.** Минус 40 пен 40 аралығындағы кездейсоқ бүтін сандардан тұратын A массивін құру. Массивті шығару. Массивтің барлық элементтерін сұрыптау. Құрылған массив элементтерін өлшемі жағынан бірдей басқа массивке көшіру. Онда диалог режимінде берілген кілт бойынша элементтің бинарлық іздестірілуін орындау. Бағдарламаны жазғанда System.Array класының әдістерін қолдану.

Бағдарлама коды:

**Массивті жариялау:**

```
public static Array masi = Array.CreateInstance(typeof(int), cnt, niz);
```

**Массив элементтерін қолдану:**

```
masi.SetValue((rnd.Next(81) - 40), i); - циклда массив элементтерін жазуды орындау  
Console.Write(masi.GetValue(i) + " "); - циклда массив элементтерін оқуды орындау
```

**Сорттау және басып шығару:**

```
Array.Sort(masi);  
foreach (int i in masi)  
    Console.Write(i + " ");
```

## Array классының әдістерін программада қолдану

System.Array класының әдістері арқылы құрылған массивті а массивіне көшіреміз:

```
Array.Copy(masi, a, masi.Length);
```

а массивін реверстеу:

```
Array.Reverse(a);
```

Кілт бойынша бинарлы іздестіру:

```
int idx = Array.BinarySearch(masi, k);
```

## C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

Есеп. Минус 40 пен 40 аралығындағы кездейсоқ бүтін сандардан тұратын A массивін құру. Массивті шығару. Массивтің барлық элементтерін сұрыптау. Құрылған массив элементтерін өлшемі жағынан бірдей басқа массивке көшіру. Онда диалог режимінде берілген кілт бойынша элементтің бинарлық іздестірілуін орындау. Бағдарламаны жазғанда System.Array класының әдістерін қолдану.

**Array.CreateInstance(typeof( int),ctl,niz);** әдісі арқылы 15 элементтен тұратын бір өлшемді masi массивін құраймыз (Lengths қасиеті ctl айнымалысына тең)

For циклінде masi.GetLowerBound(0) қолданылады. Айнымалының бастапқы мәні 1-ге, ал соңғы мәні – masi.Length 15-ке тең.

```
for (int i = masi.GetLowerBound(0); i <= masi.Length; i++)  
masi.SetValue((rnd.Next(81) - 40), i);
```

## C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

Массив элементтерін қолдану `System.Array` класының әдістері арқылы ғана мүмкін:

- жазуда - **`masi.SetValue`** және
- оқуды орындағанда - **`masi.GetValue`**.

Массивті сұрыптауды **`Array.Sort(masi)`**; статикалық әдісімен орындаймыз.

Массивті `foreach` циклі арқылы монитор экранына шығарамыз:

```
foreach (int i in masi)  
Console.Write(i + " ");  
Console.WriteLine();
```

`System.Array` класының әдістері арқылы құрылған массивті басқа массивке көшіру:

```
Array.Copy(masi, a, masi.Length);
```

# C# ТІЛІНІҢ СТАНДАРТТЫ МАССИВТЕРІ

Есеп. Минус 40 пен 40 аралығындағы кездейсоқ бүтін сандардан тұратын A массивін құру. Массивті шығару. Массивтің барлық элементтерін сұрыптау. Құрылған массив элементтерін өлшемі жағынан бірдей басқа массивке көшіру. Онда диалог режимінде берілген кілт бойынша элементтің бинарлық іздестірілуін орындау. Бағдарламаны жазғанда System.Array класының әдістерін қолдану.

```
Massiv kurildi!?:
18 25 23 32 -33 -33 34 -40 24 39 -33 32 9 40 19
Massivti sorttaimiz jane shigaramiz:
-40 -33 -33 -33 9 18 19 23 24 25 32 32 34 39 40
Massivti bytin sandardan tyratin massivke koshiremiz:
-40 -33 -33 -33 9 18 19 23 24 25 32 32 34 39 40
Shana massivti reverstey jane shigary:
40 39 34 32 32 25 24 23 19 18 9 -33 -33 -33 -40
Kilt boinsha massivte sandi binarli izdey
Izdey kiltin engiziniz
24
Sannin indeksi = 9
jok kiltti engiziniz
555
Sannin indeksi = -17
```