

TÖMBÖK

C# NYELVEN

Mire jók a tömbök?

A tömb olyan adatszerkezet, amely több azonos típusú elem tárolására alkalmas.

Statikus tömbök létrehozásakor megadjuk:

- Az elemek típusát
- A tömb dimenzióját
- A tömb nevét
- Az elemek számát
- Nem kötelezően: az elemek kezdőértékét

Tömbök deklarációja, definiálása, inicializálása

Egy dimenziós tömbök = vektorok

méret; ha megadjuk az elemeket, elhagyható

típus

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

`string[] nevek=new string[] {"Ali", "Bea", "Ida"};`

`double[] tortek={0.5,1.3, 2.8, 3.14};`

név

indexek

Egy 10 elemű vektor:

0.	1.	2.	3.	4.	5.	6.	7.	8.	9.
15	27	11	42	32	21	30	4	20	13

A tömbelemek elérése

- A tömb elemeire az indexükkel hivatkozunk
- Pl: `t[0]`, `nevek[2]`
- A kezdőindex mindig 0
- Az utolsó elem `indexe=elemszám-1`
- A `t` tömb elemszáma: `t.Length`

A tömb bejárása for ciklussal

```
for(int i=0; i<t.Length; i++)  
{  
    Console.Write(t[i]+" ");  
}
```

```
/*végigmegyünk a tömb elemein, és kiírjuk egymás  
mellé szóközzel elválasztva*/
```

A tömb bejárása foreach ciklussal

```
foreach(int elem in t)
```

```
{
```

```
    Console.Write(elem+" ");
```

```
}
```

```
/*végigmegyünk a tömb elemein, és kiírjuk egymás  
mellé szóközzel elválasztva*/
```

A tömb rendezése

`Array.Sort(t); /*növekvő sorrendbe rendezi a t tömböt*/`

`Array.Sort(t1,t2); /*a t1 tömb, mint kulcs szerint rendezi a t1 és t2 tömböt*/`

`Array.Sort(t,index,darab); /*a t tömb indexedik elemétől kezdve a megadott darab elemet rendezi*/`

Tömb metódusok

`t.Contains(elem);` */*true vagy false értéket ad vissza
aszerint, hogy az elem megtalálható-e
a t tömbben*/*

`t.Sum();` */*a t tömb elemeinek összege*/*

`t.Average();` */*a t tömb elemeinek átlaga*/*

`t.Max();` */*a t tömb legnagyobb eleme*/*

`t.Min();` */*a t tömb legkisebb eleme*/*

`Array.IndexOf(t,elem);` */*a megadott elem hányadik a t
tömbben; ha nincs benne,akkor -
1-et ad vissza*/*

Több dimenziós tömbök

2 dimenziós tömbök = mátrixok

```
int[, ] t=new int[4,3];
```

t[0,0] 25	t[0,1] 17	t[0,2] 14
t[1,0] 5	t[1,1] 10	t[1,2] 8
t[2,0] 33	t[2,1] 4	t[2,2] 51
t[3,0] 2	t[3,1] 42	t[3,2] 11