

# Vektorok

## Tömbök kezelése

Azonos típusú adatok sorozatát *tömbnek* nevezzük. Az egydimenziós tömb vagy *vektor* elemeire azok sorszámaival hivatkozhatunk. (A kétdimenziós tömb vagy mátrix elemeit két adattal, a sor- és az oszlopazonosítóval érhetjük el.)

Példánkban beolvassuk a  $b$  tömbbe az  $n=10$  egész számot, kiszámoljuk azok összegét, majd kiírjuk a számokat valamint az összeget.

```
static void Main(string[] args)
{
    int n = 10;
    int[] b = new int[n];

    for (int i=0; i<n; i++)
    {
        Console.Write("{0}. elem=", i);
        b[i] = Convert.ToInt32(Console.ReadLine());
    }

    int s = 0;
    for (int i = 0; i < n; i++)
        s = s + b[i];

    for (int i = 0; i < n; i++)
        Console.Write("{0} ", b[i]);
    Console.Write(" elemek összege={0}", s);

    Console.ReadKey();
}
```

## Példa függvények alkalmazására

A következő példában a program a *fakt* függvényt hívja meg. A *fakt* függvény megkapja az  $n$  egész típusú változó értékét, mely a  $gy$  változóba kerül és a *return* utasítással visszaadja az  $s$  egész változó értékét.

```
static void Main(string[] args)
{
    Console.Write("n=");
    int n = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("{0}!={1}", n, fakt(n));
    Console.ReadKey();
}

static int fakt(int gy)
{
    int s = 1;
    for (int i = 1; i <= gy; i++)
        s = s*i;
    return s;
}
```

## Példa tömbök kezelésére függvények használatával

Következő példánkban véletlenszerűen előállítjuk és kiíratjuk Dudley Durstly karácsonyi ajándékainak árát (Feladat1), meghatározzuk azok összértékét (Feladat2), majd a 200\$-nál értékesebb ajándékok számát (Feladat3), végül a legértékesebb ajándék árát (Feladat4).

A részfeladatokat a Main függvényből egyenként hívjuk meg.

A megoldás leírásában a magyarázatok a // jelek után, zöld színnel találhatók.

```
//A Main() függvényt megelőzik a valamennyi függvényből elérhető globális változók
static int n = 10;
static int[] ajándék = new int[n];
//A Main() függvény csupán meghívja a részfeladatokat végző függvényeket
static void Main(string[] args)
{
    Feladat1();
    Feladat2();
    Feladat3();
    Feladat4();

    Console.ReadKey();
}

//Az adatok véletlenszerű előállítása és kiírása
//A void arra utal, hogy a függvény nem ad vissza értéket
static void Feladat1()
{
    //Az rnd egy véletlenszerű sorozat lesz
    Random rnd = new Random();
    Console.WriteLine("Dudley Durstly ajándékai ($) : ");
    for (int i = 0; i < n; i++)
    {
        //A .Next(100,301) 100 és 300 közötti véletlenszerű egészet állít elő
        ajándék[i] = rnd.Next(100, 301);
        Console.WriteLine(ajándék[i] + " ");
    }
    Console.WriteLine();
}

//Az összérték meghatározása az Összegzés tételével
static void Feladat2()
{
    int összeg = 0;
    for (int i = 0; i < n; i++)
        összeg = összeg + ajándék[i];

    Console.WriteLine("Az ajándékok összértéke= " + összeg + " $");
}

//A 200$-nál drágább ajándékok száma a Megszámlálás tételével
static void Feladat3()
{
    //A db lokális változó lesz, csak ebben a függvényben érhető el
    int db = 0;
    for (int i = 0; i < n; i++)
        if (ajándék[i] > 200)
            db++;
    Console.WriteLine("200 $-nál {0} értékesebb ajándék van.", db);
}
```

```
//A legértékesebb ajándék értéke a maximum-kiválasztás tételével
static void Feladat4()
{
    //Feltételezzük, hogy az első ajándék a legértékesebb
    //maxh: a maximum helye a vektorban
    int maxh = 0;
    //Majd végignézzük az összes elemet, és ha nagyobbat találunk, az lesz a legnagyobb
    for (int i=1; i<n; i++)
        if (ajándék[i]>ajándék[maxh])
            maxh=i;
    Console.WriteLine("Dudley legértékesebb ajándéka {0} $-be t.",ajándék[maxh]);
}
```