

Struktúrák kezelése

A struktúra fogalma és létrehozása

Struktúrák segítségével könnyen kezelhetünk összetartozó adatokat, amelyek akár eltérő típusúak is lehetnek. Például egy egységben kezelhetjük egy iskolai osztály jelét, létszámát és tanulmányi átlagát.

A struktúra használatához először meg kell adnunk annak szerkezetét, vagyis azt, hogy milyen mezőkből épül fel:

```
struct EgyOsztaly
{
    public string jel;
    public int letszam;
    public double atlag;
}
```

Ha például a *tizedé* egy *Egyosztaly* típusú változó, akkor az osztály adatait a következőképpen adhatjuk meg:

```
tizedé.letszam = 31;
```

Egy iskola természetesen nem egy osztályból áll; az iskola osztályainak adatait tárolhatjuk például egy struktúratömbben:

```
static int n = 16;
static EgyOsztaly[] oszt = new EgyOsztaly[n];
```

Ekkor, ha a 10.d osztály adatait a tömb 8. eleme tárolja:

```
oszt[7].jel = "10.d";
oszt[7].letszam = 31;
```

Adatok beolvasása fájlból egy struktúra tömbbe

Nagy mennyiségű adatot általában nem közvetlenül a billentyűzetről kérünk be, hanem egy adatfájlból. Az érettségi vizsgán például az adatokat többnyire egy szöveges állományban kapjuk. Esetünkben a fájl minden sora egy-egy osztály adatait tartalmazza szóközzel elválasztva:

```
9.a 32 4,55
9.b 31 4,88
9.c 28 4,6
9.d 29 4,77
```

A C#-ban a fájl megnyitása után az adatokat soronként olvassuk be. A beolvasott sorokat a korábban már megismert *Split()* függvénnyel daraboljuk fel a szóközők mentén az *egysor* nevű, három elemű sztringtömbbe. Végül az osztályok adatait konvertálás után egyenként áthelyezzük a struktúra tömb megfelelő elemének mezőibe:

```

StreamReader sr = new StreamReader("osztaly.txt");
string[] egysor = new string[3];
for (int i = 0; i < n; i++)
{
    egysor = sr.ReadLine().Split(' ');
    oszt[i].jel = egysor[0];
    oszt[i].letszam = Convert.ToInt32(egysor[1]);
    ...
}

```

A teljes példa

A következő példa a teljes programrészt tartalmazza. A *Beolvas()* függvény az *osztaly.txt* fájl adatait beolvassa az *oszt[]* struktúratömbbe, a *Legjobb()* függvény pedig megkeresi és kiírja a legnagyobb átlagú osztályt. (A programban feltételezzük, hogy ez egyértelmű.)

```

...
struct EgyOsztaly
{
    public string jel;
    public int letszam;
    public double atlag;
}
static int n = 16;
static EgyOsztaly[] oszt = new EgyOsztaly[n];

static void Main(string[] args)
{
    Beolvas();
    Legjobb();
    ...
    Console.ReadKey();
}

static void Beolvas()
{
    StreamReader sr = new StreamReader("osztaly.txt");
    string[] egysor = new string[3];
    for (int i = 0; i < n; i++)
    {
        egysor = sr.ReadLine().Split(' ');
        oszt[i].jel = egysor[0];
        oszt[i].letszam = Convert.ToInt32(egysor[1]);
        oszt[i].atlag = Convert.ToDouble(egysor[2]);
    }
    sr.Close();
}

static void Legjobb()
{
    int maxh = 0;
    for (int i = 0; i < n; i++)
        if (oszt[i].atlag > oszt[maxh].atlag)
            maxh = i;
    Console.WriteLine("A legjobb osztály: {0}", oszt[maxh].jel);
}

```