

Struktúra

Befejezésül a struktúra definiálására, használatára, struktúra vektorra nézzünk egy teljesebb példát:

Példa:

```
using System;
struct struktúra_pelda
{
    public int kor;
    public string név;
}
class struktúra_használ
{
    public static void Main()
    {
        struktúra_pelda sp=new struktúra_pelda();
        sp.kor=5;
        sp.név="Éva";
        // struktúra_pelda vektor
        struktúra_pelda [] spv=new struktúra_pelda[5];
        int i=0;
        // beolvasás
        while(i<5)
        {
```

IV.2. Struktúra

Gyakran szükségünk van az eddigiektől eltérő adatszerkezetekre, amikor a leírni kívánt adatunkat nem tudjuk egy egyszerű változóval jellemezni. Elég, ha a talán legkézenfekvőbb feladatot tekintjük: hogyan tudjuk a sík egy pontját megadni? Egy lehetséges megadási mód, ha a pontot mint a sík egy derékszögű koordináta-rendszerének pontját tekintem, és megadom az X és Y koordinátákat.

A struktúradefiniálás az ilyen feladatok megoldása esetén lehetővé teszi, hogy az összetartozó adatokat egy egységként tudjuk kezelni.

A struktúradefiníció formája a következő:

Összeállította: Szeli Márk

```
struct név {  
    hozzáférés típus mezőnevek;  
    ...  
    hozzáférés függvénydefiníció ;  
};
```

Hivatkozás egy mezőre: *változónév.mezőnév*

A struktúrákon azonos típus esetén az értékadás elvégezhető, így ha például a és b azonos típusú struktúra, akkor az $a=b$ értékadás szabályos, és az a minden mezője felveszi b megfelelő mezőértékeit.

Példa:

```
struct pont  
{  
    int x;  
    int y;  
};  
...  
pont a;  
...  
struct egy  
{  
    string name;  
    int kor;  
};  
egy[] c=new egy[10]; // 10 elemű struktúra vektor
```

A fenti definíció teljesen jó, csak a hozzáférési szint megadásának hiányában minden mező privát, azaz a struktúra mindkét adata csak belülről látható. A struktúra alapértelmezett mezőhozzáférése privát (*private*). Minden taghoz külön meg kell adni a hozzáférési szintet.

Struktúra esetén a megengedett hozzáférési szintek:

private	csak struktúrán belülről érhető el
public	bárki elérheti ezt a mezőt
internal	programon belülről (assembly) elérhető

Példa:

```
struct személy  
{  
    public string név;
```

Összeállította: Szeli Márk

```
    public int kor;  
};
```

Az egyes mezőkre hivatkozás formája:

Példa:

```
    személy st=new személy();  
    System.WriteLine( st.kor); // kor kiírása
```

Struktúra definíció esetén a nyelv nem csak adatmező, hanem függvénymező definiálást is megenged. Azt a függvénymezőt, aminek ugyanaz a neve, mint a struktúrának, konstruktornak nevezzük. Paraméter nélküli konstruktor nem definiálható, azt mindig a környezet biztosítja. A struktúra értéktípusú adat, így a vermen és nem a dinamikus memóriában jön létre. Általában elmondható, hogy a struktúra majdnem úgy viselkedik, mint egy osztály, de funkcionalitását tekintve megmarad a különböző típusú adatok tárolásánál.

A struktúrákkal kapcsolatosan érdemes megjegyezni három alapvető tulajdonságot:

- Egy struktúra adatmezőt a definiálás pillanatában nem inicializálhatunk.

Példa:

```
    struct alma  
    {  
        string nev="jonatán"; // fordítási hiba  
        ...  
    }
```

- Struktúra adatmezőket a *default* konstruktor nem inicializál. A kezdőérték beállításáról, ha szükséges, saját konstruktorral vagy egyéb beállítási móddal kell gondoskodni.

Példa:

```
    struct pont  
    {  
        public int x,y;  
    }  
    pont p=new pont();  
    p.x=2; p.y=4;
```

- Bár a struktúra érték típusú, azért a *new* operátorral kell biztosítani a saját konstruktorral történő inicializációt. Ez ebben az esetben a vermen fog elhelyezkedni.

Összeállította: Szeli Márk

Befejezésül a struktúra definiálására, használatára, struktúra vektorra nézzünk egy teljesebb példát:

Példa:

```
using System;
struct struktúra_példa
{
    public int kor;
    public string név;
}
class struktúra_használ
{
    public static void Main()
    {
        struktúra_példa sp=new struktúra_példa();
        sp.kor=5;
        sp.név="Éva";
        // struktúra_példa vektor
        struktúra_példa [] spv=new struktúra_példa[5];
        int i=0;
        // beolvasás
        while(i<5)
        {
```