

## Hiányzások

### T 54 481 06/2/4 feladat megoldása

Első lépésként a konzol applikáció megoldását mutatom meg. Az eddigiek közül ez a feladat tűnik a legegyszerűbbnek. Nincsenek osztályok, csak egy viszonylag egyszerű feladat. Mivel a feladat leírását korábban átküldtem, most is csak hivatkozom az egyes alfeladatokra.

1. Létrehozunk egy konzolalkalmazást **Hiányzasok** néven. Ebben alkalmazásban deklarálunk egy struktúrát, amelyben a hiányzások adatait tároljuk majd. Mivel több adatot is kell tárolnunk, ezért létrehozunk egy struktúratömböt.

A szeptember.csv állomány minden sora tartalmaz egy nevet, az osztály azonosítót, a hiányzás első és utolsó napját, valamint a hiányzott órák számát. Az első kettő szöveges, az utóbbi három egész típusú adat. Így a struktúra a következő lesz:

```
namespace Hiányzasok
{
    struct Mulaszt
    {
        public string nev;
        public string osztaly;
        public int enap;
        public int unap;
        public int ora;
    }
```

```
class Program
{
    A struktúra minden eleme publikus kell legyen, különben nem érhetjük el. Amikor a struktúránk elkészült a program osztályon belül deklarálunk egy 1000 elemű tömböt a struktúránkból.
```

```
class Program
{
    static Mulaszt[] m = new Mulaszt[1000];
```

A következő lépés a fájl adatainak beolvasása és tárolás a struktúratömbben. Ezt a részfeladatot a visszatérési érték nélküli **Beolvas** függvény végzi. Az első beolvasott sor a fejléct tartalmazza – arra nincs szükségünk, de be kell olvasnunk (tartalma a **fejlec** változóba kerül).

```
static void Beolvas(ref int db)
{
    string fnev = "E:\\forras\\szeptember.csv";
    StreamReader f = File.OpenText(fnev);
    string fejlec = f.ReadLine();
    while (!f.EndOfStream)
    {
```

```

        string sor = f.ReadLine();
        string[] szo = sor.Split(';');
        m[db].nev = szo[0];
        m[db].osztaly = szo[1];
        m[db].enap = Convert.ToInt32(szo[2]);
        m[db].unap = Convert.ToInt32(szo[3]);
        m[db].ora = Convert.ToInt32(szo[4]);
        db++;
    }
    f.Close();
}

```

Az adatok beolvasása után zárjuk a fájlt (**f.Close()** ;).

2. Az összes mulasztott óra kiíratásához nem kell mást tennünk, mint a struktúratömb ora elemében lévő számokat összeadjuk. A feladatot az int visszatérési értékű Orak függvény oldja meg.

```

static int Orak()
{
    int osszes = 0;
    for (int i = 0; i < m.Length; i++)
        osszes += m[i].ora;
    return osszes;
}

```

3. A **string** visszatérési értékű **Kerdez** függvény kérdezi meg a napot és a tanuló nevét. A függvénynek van egy **out** módosítóval rendelkező paramétere, mert nemcsak a tanuló nevét, de a nap számát is vissza kell kapnunk a későbbi feladatok megoldásához.

```

static string Kerdez(out int a)
{
    string b;
    Console.WriteLine("3. feladat");
    Console.Write("\t Kérem, adjon meg egy napot:");
    a = Int32.Parse(Console.ReadLine());
    Console.Write("\t Tanulo neve: ");
    b = Console.ReadLine();
    return b;
}

```

Amikor a főprogramból meghívjuk a **Kerdez** függvényt, az egész típusú paraméter előtt szintén szerepelni kell az **out** szócskának.

4. A feladat megoldásához készítettem a bool visszatérési értékű Volte függvényt. A függvény paraméterként kapja a Kerdez függvény által beolvasott tanuló nevét. A while ciklussal megkeressük a struktúratömbben a tanuló nevét. Ha volt hiányzása a Volte függvény true értékkel tér vissza, egyébként false a visszatérési értéke. A főprogramban if szerkezetben megvizsgálom a függvény visszatérési értékét. Értékétől függően íratom ki az üzenetet.

```

static bool Volte(string nev)
{
    int i = 0;
    while (nev != m[i].nev) i++;
    if (m[--i].ora != 0) return true;
    else return false;
}

```

5. Az adott napon hiányzók nevének és osztályának kiírását a visszatérési érték nélküli **Hianyok** függvény végzi. Paraméterként megkapja a 3. feladatban beolvasott napot.

```

static void Hianyok(int nap)
{
    Console.WriteLine("5. feladat: Hiányzó tanulók 2017.
                        szeptember {0}-én", nap);
    for (int i = 0; i < m.Length; i++)
    {
        if (m[i].enap == nap)
        {
            Console.WriteLine("\t {0} ({1})", m[i].nev,
                              m[i].osztaly);
        }
    }
}

```

6. Adatok kiírását a **Kiir** függvény végzi. Itt sincs visszatérési érték.

```

static void Kiir()
{
    string fnev = "osszesites.csv";
    StreamWriter f = File.CreateText(fnev);
    string[] oszt = { "1a", "1b", "2a", "2b", "3a", "3b",
                      "4a", "4b", "5a", "5b", "6a", "6b", "7a",
                      "7b", "8a", "8b" };
    foreach (string j in oszt)
    {
        int szam = 0;
        for (int i = 0; i < m.Length; i++)
            if (Equals(j, m[i].osztaly)) szam += m[i].ora;
        string sor = j + ";" + szam.ToString();
        f.WriteLine(sor);
    }
    f.Close();
}

```

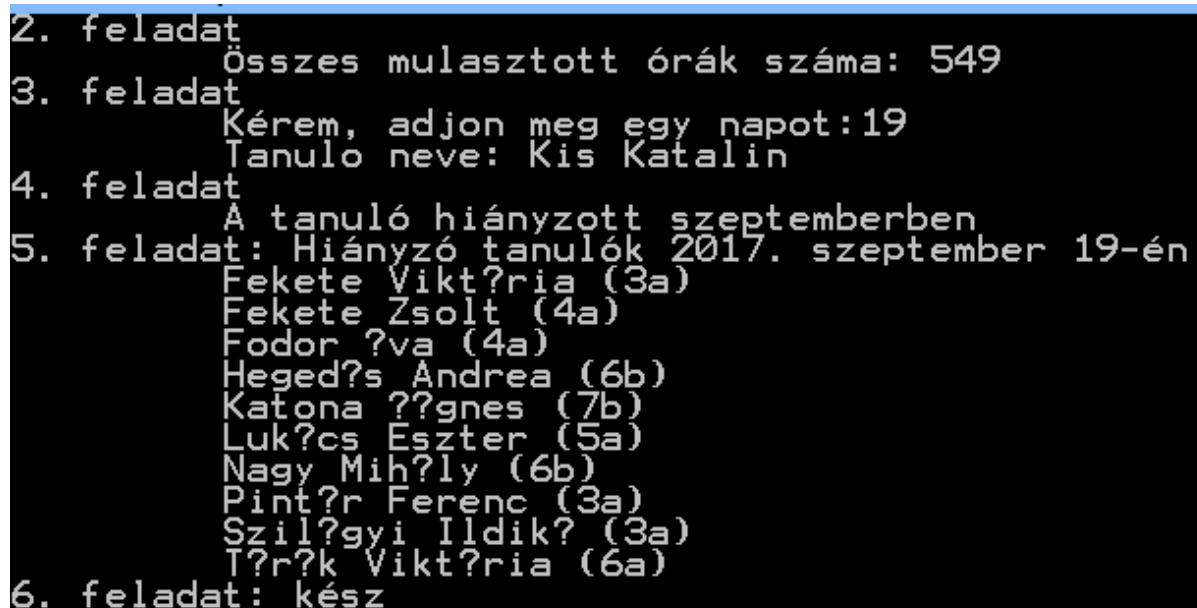
A függvény elején deklarálom az **oszt** nevű tömböt, amelyben felsorolom az osztályok neveit. Ezután **foreach** ciklussal végigmegyek a tömb elemein. A cikluson belül a **for** ciklus pedig a struktúrátömb elemeit olvassa végig. Ha az **oszt** aktuális eleme egyezik a

struktúratömbben lévő osztálynévvel, akkor a számláló (**szam** változó) értékét növelem a struktúratömb **ora** nevű elemében lévő értékkel. A **string**ek összehasonlítását az **Equals** metódus végzi.

Már csak a főprogram listája van hátra. Itt a 6. feladathoz is beraktam egy üzenetet, amely közli velem, hogy a fájl írása kész. A főprogram listája:

```
static void Main(string[] args)
{
    int db = 0, nap;
    string nev;
    Beolvas(ref db);
    Console.WriteLine("2. feladat");
    Console.WriteLine("\t Összes mulasztott órák száma:
        {0}", Orak());
    nev = Kerdez(out nap);
    Console.WriteLine("4. feladat");
    if (Volte(nev))
        Console.WriteLine("\t A tanuló hiányzott szeptember-
            ben");
    else
        Console.WriteLine("\t A Tanuló nem hiányzott szeptemberben");
    Hianyok(nap);
    Kiir();
    Console.WriteLine("6. feladat: kész");
    Console.ReadKey();
}
```

A program futási képe:



```
2. feladat
    Összes mulasztott órák száma: 549
3. feladat
    Kérem, adjon meg egy napot:19
    Tanulo neve: Kis Katalin
4. feladat
    A tanuló hiányzott szeptemberben
5. feladat: Hiányzó tanulók 2017. szeptember 19-én
    Fekete Vikt?ria (3a)
    Fekete Zsolt (4a)
    Fodor ?va (4a)
    Heged?s Andrea (6b)
    Katona ??gnes (7b)
    Luk?cs Eszter (5a)
    Nagy Mih?ly (6b)
    Pint?r Ferenc (3a)
    Szil?gyi Ildik? (3a)
    T?r?k Vikt?ria (6a)
6. feladat: kész
```

Az eredmények nem teljesen egyeznek meg a megadottakkal, mert csak szeptember 20-ig van meg a fájl (hiányzik a vége – viszont mindent jól csinál a program).

A feladat megoldható osztálydeklarációval is. Ekkor az osztályra bízunk az adatok feldolgozását. Az adattárolást pedig az osztályból létrehozott tömbbel valósítjuk meg. A lényeges eltérés az előző és a mostani megoldás között, hogy nem struktúratömbben, hanem osztálytömbben tároljuk az adatokat.

Az osztálydeklaráció:

```
class Mulasztas
{
    public string nev;
    public string osztaly;
    public int enap;
    public int unap;
    public int ora;
    public Mulasztas(string sor)
    {
        string[] szo = sor.Split(';');
        nev = szo[0];
        osztaly = szo[1];
        enap = Convert.ToInt32(szo[2]);
        unap = Convert.ToInt32(szo[3]);
        ora = Convert.ToInt32(szo[4]);
    }
}
```

A főprogram utasításai::

```
class Program
{
    static Mulasztas[] m = new Mulasztas[1000];
    static void Beolvas(ref int db)
    {
        string fnev = "E:\\forras\\szeptember.csv";
        StreamReader f = File.OpenText(fnev);
        string fejlec = f.ReadLine();
        while (!f.EndOfStream)
        {
            string sor = f.ReadLine();
            Mulasztas uj = new Mulasztas(sor);
            m[db] = uj;
            db++;
        }
        f.Close();
    }

    static int Orak(int veg)
    {

```

```

        int osszes = 0;
        for (int i = 0; i < veg; i++)
            osszes += m[i].ora;
        return osszes;
    }

```

```

static string Kerdez(out int a)
{
    string b;
    Console.WriteLine("3. feladat");
    Console.Write("\t Kérem, adjon meg egy napot:");
    a = Int32.Parse(Console.ReadLine());
    Console.Write("\t Tanulo neve: ");
    b = Console.ReadLine();
    return b;
}

```

```

static bool Volte(string nev)
{
    int i = 0;
    while (nev != m[i].nev) i++;
    if (m[--i].ora != 0) return true;
    else return false;
}

```

```

static void Hianyok(int nap, int veg)
{
    Console.WriteLine("5. feladat: Hiányzó tanulók 2017.
        szeptember {0}-én", nap);
    for (int i = 0; i < veg; i++)
    {
        if (m[i].enap == nap)
        {
            Console.WriteLine("\t {0} ({1})", m[i].nev,
                m[i].osztaly);
        }
    }
}

```

```

static void Kiir(int veg)
{
    string fnev = "osszesites.csv";
    StreamWriter f = File.CreateText(fnev);
    string[] oszt = { "1a", "1b", "2a", "2b", "3a", "3b",
        "4a", "4b", "5a", "5b", "6a", "6b", "7a", "7b",
        "8a", "8b" };
    foreach (string j in oszt)

```

```

    {
        int szam = 0;
        for (int i = 0; i < veg; i++)
            if (Equals(j, m[i].osztaly)) szam += m[i].ora;
        string sor = j + ";" + szam.ToString();
        f.WriteLine(sor);
    }
    f.Close();
}

static void Main(string[] args)
{
    int db = 0, nap;
    string nev;
    Beolvas(ref db);
    Console.WriteLine("2. feladat");
    Console.WriteLine("\t Összes mulasztott órák száma:
        {0}", Orak(db));
    nev = Kerdez(out nap);
    Console.WriteLine("4. feladat");
    if (Volte(nev))
        Console.WriteLine("\t A tanuló hiányzott szeptem-
            berben");
    else
        Console.WriteLine("\t A Tanuló nem hiányzott szept-
            emberben");
    Hianyozok(nap, db);
    Kiir(db);

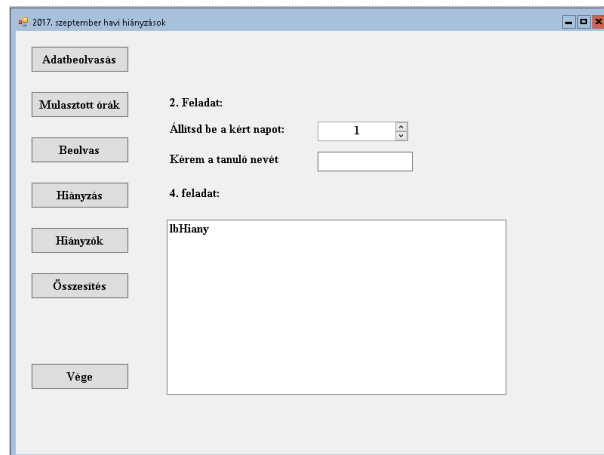
    Console.WriteLine("6. feladat: kész");
    Console.ReadKey();
}
}

```

Természetesen létezik más megoldás is, nem a bemutatott az egyetlen lehetséges megoldás. Tartozom még a grafikus megoldással.

## A feladat grafikus megoldása

Készítsd el az alábbi formot:



Most elneveztem a nyomógombokat: az **Adatbeolvasás** neve **olvas**, a **Mulasztott órák** neve **mulora**, a **Beolvas** neve **adat**, a **Hiányzás** neve **hiany**, a **Hiányzók** neve: **fel5**, az **Összesítés** neve: **kiir** és a **Vége** neve **vege**. A címkék közül a **2. Feladat** neve **fel2**, a **4. feladat** neve **volte**, illetve van egy címke az **Összesítés** és **Vége** gomb között, amelynek neve **fel6**. Ebben csak akkor jelenik meg a szöveg, ha az adatok fájlba írása megtörtént. A 3. feladatban bekért napot **NumericUpDown** elemmel oldottam meg (ha grafika, hát legyen grafika). Az elem neve: **ertek**. A **TextBox** neve **nev**. A **listBox** neve **lbHiany**.

A feladatban adattárolásra azt az osztályt használtam fel, amelyet a konzolos megoldásnál mutattam. Nem kell újra írni. A program nevén helyi menüt kérek, majd az **Add** opciónál feljövő menüből az **Existing item...** lehetőséget választom és megkeresem abban a programban, amelyben deklaráltam. Mindössze a névteret kell átírnom az aktuális program névterére.

Ezek után készítsük el a programot. Mire van szükségem? Először is deklarálnom kell a **Mulasztas** osztályból tömböt. Szükségem van a **db** változóra, amelyben a fájlban lévő sorok számát tárolom (a tömb eleminek ez lesz a száma). A 3. feladatban bekért adatok tárolását is úgy kell megoldanom, hogy elérhető legyen a program minden részében. Ezért a nap tárolására felveszem a **szam**, a tanuló nevének tárolására a **tanulo** változót. Ezeket a program elején privát változóként deklarálom. A **Mulasztas** tömb inicializálását a form konstruktorára bízom:

```
public partial class Form1 : Form
{
    private int db = 0;
    private int szam;
    private string tanulo;
    Mulasztas[] m;
    public Form1()
    {
        InitializeComponent();
    }
}
```



```

        m = new Mulasztas[100];
    }

```

Amíg nem olvastam be az adatokat, addig csak két nyomógombot érhetek el: ezek az Adatbeolvasás, illetve a Vége gomb. Az összes többi gombot letiltom:

```

private void Form1_Load(object sender, EventArgs e)
{
    mulora.Enabled = false;
    adat.Enabled = false;
    hiany.Enabled = false;
    fel5.Enabled = false;
    kiir.Enabled = false;
}

```

Az adatbeolvasás hasonlóan történik, mint a konzolos megoldásnál. Természetesen az osztálytömböt használom adattárolásra. A beolvasás végén engedélyezem azokat a gombokat, amelyek innentől használhatók, illetve letiltom az Adatbeolvasás gombot:

```

private void olvas_Click(object sender, EventArgs e)
{
    string fnev = "E:\\forras\\szeptember.csv";
    StreamReader f = File.OpenText(fnev);
    string fejlec = f.ReadLine();
    while (!f.EndOfStream)
    {
        string sor = f.ReadLine();
        Mulasztas uj = new Mulasztas(sor);
        m[db] = uj;
        db++;
    }
    f.Close();
    mulora.Enabled = true;
    adat.Enabled = true;
    kiir.Enabled = true;
    olvas.Enabled = false;
}

```

Most már kiírathatjuk mennyi volt a mulasztott órák száma:

```

private void mulora_Click(object sender, EventArgs e)
{
    int osszes = 0;
    for (int i = 0; i < db; i++)
        osszes += m[i].ora;
    fel12.Text = "2. feladat: Összes mulasztott órák száma: "
        + osszes.ToString() + " óra!";
}

```

Beolvassuk a 3. feladatban kért napot és a tanuló nevét. A **NumericUpDown** úgy van beállítva, hogy csak 1 és 30 közötti értékeket lehessen bekérni (szeptember 30 napos). Nem csináltam semmilyen védelmet a dátumot illetően, hisz szombat vagy vasárnap esetén

nincs hiányzó (üres lesz a lista). A tanuló nevéhez a kiírásnál követtem el ellenőrzést, de az sem bonyolult. Engedélyezem a Hiányzás és Hiányzók gombokat (már megvan a működésükhöz szükséges adat):

```
private void adat_Click(object sender, EventArgs e)
{
    szam = (int) ertek.Value;
    tanulo = nev.Text;
    hiany.Enabled = true;
    fel5.Enabled = true;
    adat.Enabled = false;
}
```

A **Hiányzás** gombnál csak a **volte** címkében kell a megfelelő szöveget megjeleníteni:

```
private void hiany_Click(object sender, EventArgs e)
{
    int i = 0;
    if (tanulo != "")
    {
        while (tanulo != m[i].nev) i++;
        if (m[--i].ora != 0) volte.Text = "4. feladat: A tanuló hiányzott szeptemberben!";
        else volte.Text = "4. feladat: A tanulónak nem volt hiányzása szeptemberben!";
    }
    else volte.Text = "Nem adott meg nevet!";
}
```

A hiányzók nevei és osztályai az **lbHiany** **ListBox**-ban jelennek meg. Itt az **ir** **string** típusú változóban összeállítom a megjelenítendő adatot, majd hozzáadom a **ListBox**-hoz:

```
private void fel5_Click(object sender, EventArgs e)
{
    string ir = "5. feladat: Hiányzók 2017.09." +
        szam.ToString() + "-én";
    lbHiany.Items.Add(ir);
    for (int i = 0; i < db; i++)
    {
        if (m[i].enap == szam)
        {
            ir = "      " + m[i].nev + "; " + m[i].osztaly;
            lbHiany.Items.Add(ir);
        }
    }
}
```

Az adatkiírás megegyezik a konzolos megoldásával:

```
private void kiir_Click(object sender, EventArgs e)
```

```

{
    string fnev = "osszesites.csv";
    StreamWriter f = File.CreateText(fnev);
    string[] oszt = { "1a", "1b", "2a", "2b", "3a", "3b",
                      "4a", "4b", "5a", "5b", "6a", "6b", "7a", "7b",
                      "8a", "8b" };
    foreach (string j in oszt)
    {
        int szam = 0;
        for (int i = 0; i < db; i++)
            if (Equals(j, m[i].osztaly)) szam += m[i].ora;
        string sor = j + ";" + szam.ToString();
        f.WriteLine(sor);
    }
    f.Close();
    fel6.Text = "6. feladat kész!";
}

```

A **Vége** gomb utasítása:

```

private void vege_Click(object sender, EventArgs e)
{
    Application.Exit();
}

```

Végül a futási kép:

2017. szeptember havi hiányzások

Adatbeolvasás

Mulasztott órák

Beolvas

Hiányzás

Hiányzók

Összesítés

6. feladat kész!

Vége

2. feladat: Összes mulasztott órák száma: 549 óra!

Állítsd be a kért napot: 19

Kérem a tanuló nevét: Kis Katalin

4. feladat: A tanuló hiányzott szeptemberben!

5. feladat: Hiányzók 2017.09.19-en

- Fekete Viktória; 3a
- Fekete Zsolt; 4a
- Fodor Éva; 4a
- Hegedűs Andrea; 6b
- Katona Dávid; 7b
- Lukács Eszter; 5a
- Nagy Mihály; 6b
- Pintér Ferenc; 3a
- Szilágyi Ildikó; 3a
- Török Viktória; 6a