

Szöveges fájlok kezelése

Példa: Szövegfájlok beolvasása

Egy hálózati forgalomfigyelő program segítségével egy fájlba kigyűjtöttük azokat az IP címeket, amelyekről megpróbálták elérni a hálózaton lévő gépünket. Olvassuk be egy *string* tömbbe ezeket az IP címeket!

A feladat megoldásakor most nem használjuk a megnyitást segítő *OpenFileDialog* vezérlőt.

A megoldásban az osztályban létrehozunk egy *ipcimek* nevezetű maximum 100 elemű *string* tömböt.

```
string[] ipcimek = new string[100];
...
private void megnyitasBtn_Click(object sender, EventArgs e)
{
    FileStream fajl = new FileStream(@"M:\adat.txt", FileMode.Open);
    StreamReader olvaso = new StreamReader(fajl);
    int db = 0;
    while (!olvaso.EndOfStream)
    {
        ipcimek[db] = olvaso.ReadLine();
        db++;
    }
    olvaso.Close();
    fajl.Close();
}
```

A fájlhoz való hozzáféréshez létrehozuk a fájl objektumot a *FileStream* osztályból.

A létrehozáskor meg kell adnunk a fájl nevét és a fájllal végezni kívánt műveletet.

```
FileStream fajl = new FileStream(@"M:\adat.txt", FileMode.Open);
```

A fájl nevének megadásakor ügyeljünk arra, hogy az elérési útnál a C# a „\” karaktert különlegesen kezeli. Ezért vagy a @ jelet használjuk az elérési út megadása előtt, vagy pedig „\\” jelethasználunk.

Például:

```
FileStream fajl = new FileStream("M:\\\adat.txt", FileMode.Open);
```

A második paraméter a fájllal elvégzendő műveleteket határozza meg. A *FileMode* paraméter lehetséges értékeit az alábbi táblázat tartalmazza.

Paraméter	Leírás
<i>Append</i>	Ha a fájl nem létezik, akkor létrehozza. Ha létezik a fájl, akkor megnyitja, és a végére írja ki az új adatokat.
<i>Create</i>	Létrehoz egy üres fájlt. Ha már létezett az adott néven a fájl, akkor felülírja.
<i>CreateNew</i>	Létrehoz egy üres fájlt. Ha már létezett az adott néven a fájl, akkor egy kivételt generál.
<i>Open</i>	Megnyitja a fájlt. Ha a fájl nem létezik, akkor hibát generál.
<i>OpenCreate</i>	Megnyitja a fájlt. Ha nem létezik, akkor létrehozza.
<i>Truncate</i>	Megnyitja a fájlt, majd 0 hosszúságúra csonkítja.

A `FileStream` objektum létrehozásakor egy harmadik paramétert is használhatunk. Ennek segítségével a fájlhoz való hozzáférési jogosultságot adhatjuk meg. A `FileAccess.Read` csak olvasási jogot, a `Write` csak írási jogot, a `ReadWrite` olvasási és írási jogot kér a fájlra.

A szövegfájl kezelését végző `StreamReader` osztály példányosításakor paraméterként a fájl nevét kell megadnunk. Ha szükséges, akkor második paraméterként a szövegfájl kódolását is megadhatjuk a `CurrentEncoding` felsorolás értékeivel.

Miután megnyitottuk a fájlt olvasásra, ki kell olvasni belőle az adatokat. Ezt a `StreamReader` objektum metódusaival tudjuk megtenni.

A `Read()` metódus egy karaktert tud beolvasni, míg a `ReadLine()` metódus egy egész sort. A metódusok függvényként a visszatérési értéküként adják meg a beolvasott tartalmat.

Amikor egy szövegfájlból beolvasunk adatokat, nem mindig tudjuk, hogy hány adatsorunk van a fájlban. Ilyenkor használhatjuk az `EndOfStream` tulajdonságot, amely igaz értéket ad vissza, ha a fájl végén vagyunk.

A fájlból az adatokat ciklus segítségével dolgozhatjuk fel. A ciklus lehet dinamikus, számlálós vagy akár `foreach` ciklus is.

A fájlműveletek befejezésekor a `StreamReader` és a `FileStream` objektumokat a `Close()` eljárással bezárhatjuk.

A `StreamReader` osztály fontosabb metódusait és tulajdonságait az alábbi táblázat foglalja össze.

Metódus, tulajdonság	Leírás
<code>Read()</code>	Beolvas egy karaktert, és a következő karakterre lép.
<code>ReadLine()</code>	Beolvas egy sort, és a következő sorra lép. A beolvasott sort stringként adja vissza.
<code>Peek()</code>	Beolvas egy karaktert, de nem lép tovább. Ha nincs több karakter, akkor -1-et ad vissza.
<code>ReadToEnd()</code>	Az aktuális pozíciótól minden karaktert beolvas.
<code>Close()</code>	Lezárja a szövegfolyam olvasót.
<code>CurrentEncoding</code>	A szöveg kódolását adja meg.

Példa: Adatok kiírása szövegfájlba

Az előzőleg beolvasott IP címeket feldolgoztuk. Egy struktúra tömbbe kigyűjtöttük, hogy melyik IP címekről hányszor próbálkoztak a gép elérésével. Ezt a tömböt szeretnénk most kiírni egy fájlba.

A kigyűjtött címek egy struktúra típusú tömbbe kerülnek. A struktúra első mezője megadja az IP címet, a második pedig az adott IP cím előfordulását.

```
string[] ipcimek = new string[500];
struct Sipcimgyak
{
    public string cim;
    public int gyakszam;
}
```