

File kezelés

Olvasás text file-ból

Amennyiben egy text fájlból szeretnénk olvasni, első lépésben meg kell nyitni a fájlt. Ehhez példányosítani kell a StreamReader osztályból:

```
StreamReader r = new StreamReader("C:\\adatok.txt", Encoding.Default);
```

A text fájlokban az adatok sorokra vannak bontva. Egy lépésben egy sort szoktunk általában beolvasni. Erre a ReadLine() metódus használható, amely a sort sztring típusként adja meg:

```
string s = r.ReadLine();
```

Megnyitás után az első olvasás a fájl legelső sorára hajtódik végre. Minden olvasási művelet során automatikusan lépünk a fájlban a következő sorra. A ReadLine() többszöri alkalmazása révén előbb-utóbb minden sort be tudunk olvasni.

File végéreérés

Amennyiben egy text fájlból szeretnénk olvasni, első lépésben meg kell nyitni a fájlt. Ehhez példányosítani kell a StreamReader osztályból:

Amennyiben folyamatosan olvasunk a fájlból, úgy haladunk előre a fájlban lévő adatok feldolgozásában. Előbb-utóbb el fogjuk érni a fájl végét. A fájl végének elérésekor már nem tudunk tovább olvasni, hiszen ott már nincs adat.

A fájl végének elérését le tudjuk kérdezni a Peek() segítségével. A Peek() hivatalosan a következő byte értékét adja meg a nélkül, hogy a pozíciót léptetné. De ha nincs következő byte, akkor a Peek() -1 értéket ad vissza.

Pl.:

```
StreamReader r = new StreamReader("C:\\adatok.txt", Encoding.Default);
```

```
int db = 0;
```

```
while (r.Peek() != -1)
```

```
{
```

```
string s = r.ReadLine();
```

```
db++;
```

```
}
```

A fenti kis ciklus megszámolja, hány sor van a fájlban.

Pozicionálás file-ban

A StreamReader-el megnyitott text fájlban nem lehet pozicionálni sem előre, sem hátra.

Szöveges file megnyitása írásra

A text fájlok írásra történő megnyitására a StreamWriter osztályból kell példányosítani. Paraméterként meg kell adni a fájl nevét és a kódlapot. Ez utóbbi leggyakrabban a Default, amelyet a Windows aktuálisan is használ.

```
StreamWriter w = new StreamWriter("C:\\rainbow.txt", false, Encoding.Default);
```

A fájl nevének megadásakor érdemes a teljes elérési útvonalat leírni. Ügyeljünk, hogy a sztring literálok belsejében a `` karakternek speciális jelentése van, így egyetlen vissza-perjel leírásához a dupla `` jelet kell alkalmazni.

Írás a file-ba

A text fájlok írásra történő megnyitására a StreamWriter osztályból kell példányosítani.

Pl.:

```
StreamWriter w = new StreamWriter("C:\\rainbow.txt", false, Encoding.Default);
```

A fájlba tudunk írni a Write() és WriteLine() metódusok segítségével. Ezek használata megegyezik a Console.Write() és Console.WriteLine() használatával. Ugyanazok a formázási beállítások használhatóak, stb.

A Write() és WriteLine() között az a különbség, hogy a WriteLine() a kiírás végén egy sorvége jelet is ír a text fájlba, amíg ezt a Write() nem teszi.

Így több Write() használata során az adatok mindegyike kiíródik a fájlba, de ugyanabba a sorba kerülnek. A WriteLine() használata során minden kiírás adatai új sorba kerülnek.

File bezárása

Egy sikeresen megnyitott text fájlba való írás (vagy olvasás) végén a fájlt illik bezárni.

Erre használható a Close() metódus, mely lezárja a fájlt.

A fájl lezárásakor a módosítások garantáltan kiíródnak a háttértárolóba. E nélkül bizonyos mennyiségű adatmódosítás elveszhet.

Pl.:

```
StreamReader r = new StreamReader(@"C:\adatok.txt", Encoding.Default);
```

```
//...
```

```
r.Close();
```

vagy

```
StreamWriter w = new StreamWriter("C:\\rainbow.txt", false, Encoding.Default);
```

```
//...
```

```
w.Close();
```

Készítette: Kasalyák Máté 12/D