File kezelés

Text file megnyitása

A text fájlok megnyitására a StreamReader osztályból kell példányosítani. Paraméterként meg kell adni a fájl nevét és a kódlapot. Ez utóbbi leggyakrabban a Default, amelyet a Windows aktuálisan is használ.
StreamReader r = new StreamReader("C:\\adatok.txt", 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.

A text file jellemzői

A text file legfontosabb jellemzője, hogy az adatokat amiket tárol - szöveg formátumban vannak. A számítógép memóriájában bináris formában vannak az adatok (kettes komplemens, karakterisztika, mantissza, …), így text file-ba kiírás előtt azokat át kell alakítani, ami időveszteség. Hasonlóan – beolvasás után az adatokat általában valamilyen Parse() segítségével vissza kell alakítani bináris formájukra.

A szöveges formájú adatok ún. sorokra vannak tördelve. A sorok végét speciális vezérlő karakterek jelzik a fájlon belül. Ez Windows-os környezetben a \r\n, vagyis a 10-es és 13-as kódú vezérlő karakterek. Minden sor végén megtalálhatóak, így soronként plusz két bájtot jelentenek a fájlokban.

A text file-ok előnye azonban, hogy a bennük lévő adatok egyszerű szerkesztőprogramokkal (notepad, editorok) megtekinthetőek, módosíthatóak.

Megnyitás lehetséges problémái

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);
Elképzelhető, hogy a fájl azért nem megnyitható, mert nem is létezik, esetleg a fájlnév hibás, vagy az alkönyvtár sem létezik, vagy nincs rá jogunk, vagy egyéb problémánk támad.
A StreamReader konstruktora a sikertelenséget kivétel aktiválásával jelzi:

```
try
{
StreamReader r = new StreamReader(@"C:\adatok.txt", Encoding.Default);
//...
}
catch
{
//... nem sikerült a file megnyitása ...
}
```

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 végére érés

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 léptetetné. 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.

Pozícionálás file-ban

A StreamReader-el megnyitott text fájlban nem lehet pozícionálni 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: *Bakos Norbert*