Szöveges fájl kezelése (a Caesar-kód feltörése)

A kód feltörésének statisztikai háttere

Hosszabb szöveg esetén a Caesar-kód feltörése a statisztika módszereivel könnyen megoldható. Az egyes betűk előfordulásának gyakorisága ugyanis jellemző az adott nyelvre, így a titkos szövegben előforduló leggyakoribb karakterekről feltételezhetjük, hogy azok a leggyakoribb betűknek felelnek meg.

betű	A	B	C	D	E	F	G	H	I	J
magyar	135	20	5	22	145	9	32	17	46	17
angol	81	16	32	37	123	23	16	51	72	1

betű	N	O	P	Q	R	S	T	U	V	W
magyar	58	74	10	-	42	51	78	21	20	-
angol	72	79	23	2	61	66	96	31	9	20

Az egyes betűk gyakorisága a magyar és az angol nyelvben 1000 karakterre v
A táblázat elemzésével keressünk valami kiugró jellegzetességet a magya
(Forrás: Svékus Olivér: Titkosírások. Móra, Budapest, 1964)

Az oldalt látható statisztikát az a program készítette, amellyel a következő pontban fogunk megismerkedni. A statisztika egy magyar nyelvű szöveg (Molnár Ferenc: A Pál utcai fiúk első fejezete) Caesar-kóddal titkosított változatának betűgyakoriságát mutatja.

A 1457
B 349
C 318
D 7
E 2
F 404
G 671
H 2153
I 362
J 198
K 396
L 2093
M 149
N 553
O 244
P 697
Q 163
R 903
S 986
T 630
U 1003
V 1274
W 155
X 0
Y 651
Z 969

Leolvasható, hogy a titkos szövegben a két leggyakoribb betű a H és az L. E két betű távolsága az angol ábécében 4 karakter. A betűgyakoriságot tartalmazó táblázatunk szerint a magyar nyelvben a két leggyakoribb betű az A és az E, ezek távolsága szintén 4 karakter. Ez a „kettős csúcs” elárulja, hogy felehetően az A-nak a H, az E-nek az L felel meg, vagyis a Caesar-kódnál alkalmazott eltolás 7 karakter.

A fenitek alapján tehát sokat segít egy olyan program elkészítése, amely megszámolja az egyes betűk gyakoriságát a szöveges állományban A következő lépésben ezt a programot készítjük el.

Példa: Szöveges fájl betűgyakorisága

Példánkban megnyitjuk a *C:\Titkos.txt* fájlt, és megszámoljuk, melyik betűből hány darab van benne. Feltételezzük, hogy a szöveg csak az angol ábécé nagybetűit tartalmazza. Az egyszerűség kedvéért egy konzolalkalmazást készítünk, és a végeredmény is a konzolra írjuk ki.

Három változót kell deklarálnunk. A karakterek számát a 256 elemű s tömb fogja tartalmazni, értelemszerűen a 80-as kódú karakterek számát például az s(80) elem. Az előző pont szerint a fájl objektummal fogjuk elérni az adatfájlt a meghajtón, az olvasó objektummal pedig kiolvassuk az adatokat belőle:

```
Imports System.IO
Module Module1

  Sub Main()
      Dim s(256) As Integer
      Dim fájl As IO.FileStream
      Dim olvasó As IO.StreamReader

      fájl = New IO.FileStream("D:\Előadás
      anyagok\PROGRAMOZÁS\Gyakorlat\12.
      évfolyamnak\Fájlkezelés\Titkos.txt", FileMode.Open)
      olvasó = New IO.StreamReader(fájl)
```

A szövegfolyamot az olvasó objektum *Read* metódusaival fogjuk karakterenként beolvasni. A szöveg végét a *Peek* metódussal tudjuk meghatározni. Ez leellenőrzi a következő karaktert, s ha már nincs több karakter, akkor értéke -1 lesz.

A ciklus magjába összesen egy utasítás kerül. Az *olvasó* objektum Read metódusa beolvassa a következő karaktert, és visszaadja annak kódját. Az ennek megfelelő karakterek számát tehát eggyel növelni kell:

```
      Do While olvasó.Peek > -1
            s(olvasó.Read) += 1
      Loop
```

Végül kiíratjuk a karakterek számát a konzolra. Feltételezésünk szerint ezek mindegyike angol nagybetű:

```
      For i As Integer = AscW("A") To AscW("Z")
            Console.WriteLine(ChrW(i) & " --- " & s(i))
      Next
      olvasó.Close()
      fájl.Close()
      Console.ReadLine()
  End Sub
End Module
```

A StreamReader osztály fontosabb metódusai

Metódus	Leírás
Read	Beolvassa a következő karaktert, és eggyel továbblép.
Peek	Beolvassa a következő karaktert, de nem lép tovább. Ha nincs több karakter, -1-et ad vissza.
ReadLine	Beolvas egy sort, és sztringént adja vissza.
ReadToEnd	Az aktuális pozíciótól kezdve minden karaktert beolvas.
Close	Lezárja a szövegfolyam-olvasót

A StreamReader és StreamWriter osztály fontosabb tulajdonságai

Tulajdonság	Leírás
BaseStream	A szövegfolyamot reprezentálja.
Encoding	A szöveg kódolását adja meg (StreamWriter esetén).
Current Encoding	A szöveg kódolását adja meg (StreamReader esetén).

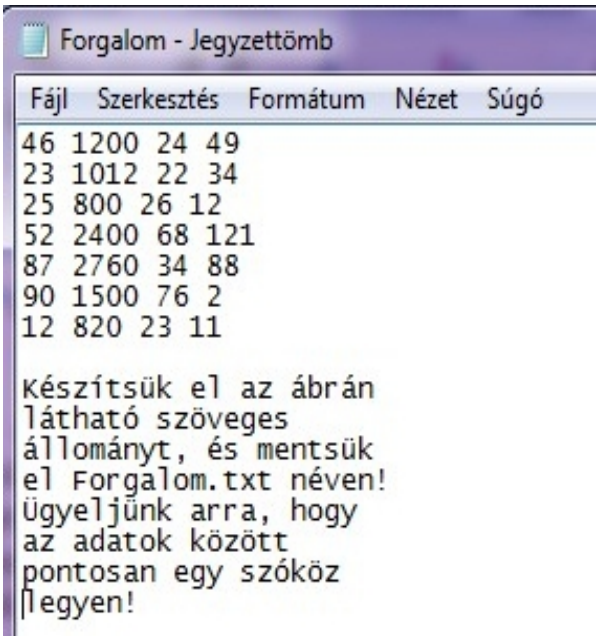
A StreamWriter osztály fontosabb metódusai

Metódus	Leírás
Write	Szöveget ír a fájlba mező szövegfolyamba.
Close	Lezárja a szövegfolyamírót.

Numerikus adatok szöveges állományban: a Split függvény

Gyakori megoldás, hogy numerikus adatokat szöveges állományban tárolnak; ennek vitathatatlan előnye, hogy a szövegfájl akár egy egyszerű jegyzettömbbel is könnyen olvasható.

A következő adathalmaz például azt tartalmazza, hogy egy adott közlekedési csomóponton a múlt héten hány gépjármű haladt át. Minden nap adatai külön sorba kerültek. Az adatsor elemei rendre az áthaladt motorkerékpárok, a személygépkocsik, a 7,5 tonnásnál kisebb, illetve a 7,5 tonnát meghaladó teherszállító járművek száma:



Példaként olvassuk be a fájl adatait, és számítsuk ki, hogy az egyes gépjárműtípusokból hány darab haladt át a héten, majd írjuk a fájl végére egy új sorba!

Az adatokat célszerűen a *forgalom* (7,4) többen fogjuk tárolni. A feladatot két részre bontjuk, a *Beolvasás* modulban feltöltjük a tömböt, míg a *Kiírás* modulban összegezzük az adatokat, és kiírjuk a fájlba:

```
Imports System.IO
Imports System.Console
Module Module1
    Private forgalom(7, 4) As Integer

    Sub Main()
        Beolvasás()
        Kiírás()
        ReadLine()
    End Sub
```

Az adatok beolvasása

Az adatok beolvasásához létrehozzuk a *fájlbe* és az olvasó objektumokat:

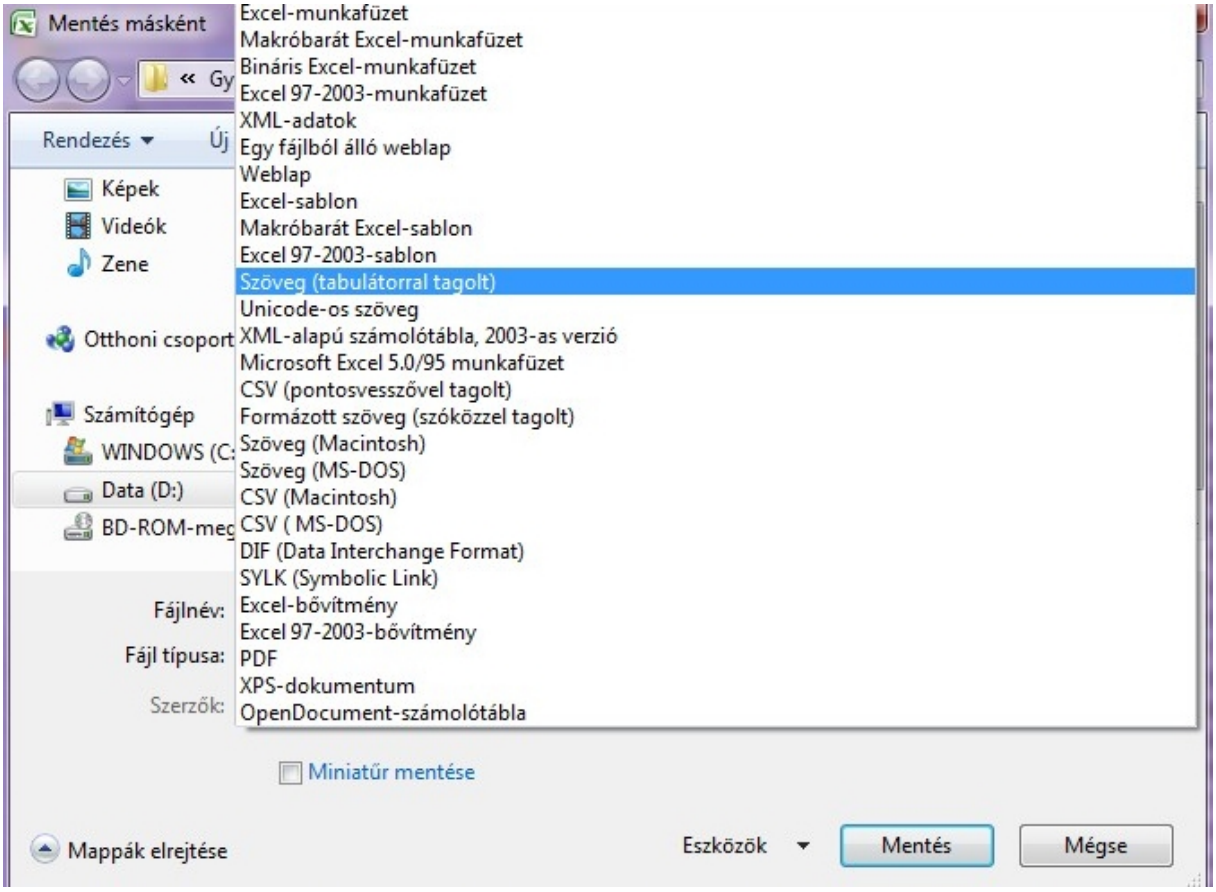
```
    Sub Beolvasás()
        Dim fájlbe As FileStream = New FileStream("D:\Előadás
        anyagok\PROGRAMOZÁS\Gyakorlat\12.
        évfolyamnak\Fájlkezelés\Forgalom.txt", FileMode.Open)
        Dim olvasó As New StreamReader(fájlbe)
```

Az adatokat soronként olvassuk be a *ReadLine* metódus segítségével. Ez egy sztringet ad vissza, melyet a *Split* függvény segítségével egy előre megadott karakter, esetünkben a szóköz

mentén szétdarabolhatunk. A *Split* tehát egy sztringvektrot határoz meg, amely ebben az esetben 4 elemű. Az elemek számozása 0-val indul. Az így kapott sztringeket pl. a *CInt* függvény segítségével konvertálhatjuk egész számokká, s így a *tábla* tömb adott sorának elemeivé:

```
        Dim str(4) As String
        For i As Integer = 1 To 7
            str = olvasó.ReadLine.Split(" ")
            forgalom(i, 1) = CInt(str(0))
            forgalom(i, 2) = CInt(str(1))
            forgalom(i, 3) = CInt(str(2))
            forgalom(i, 4) = CInt(str(3))
        Next
        olvasó.Close()
        fájlbe.Close()
    End Sub
```

A *Split* metódus első paramétere adja meg, hogy milyen karakter mentén kell szétvágnunk a sztringet. Tipikus megoldás a táblázatkezelésben a pontosvessző, az így tagolt CSV (Comma-separated values) formátumot szinte minden táblázatkezelő program ismeri.

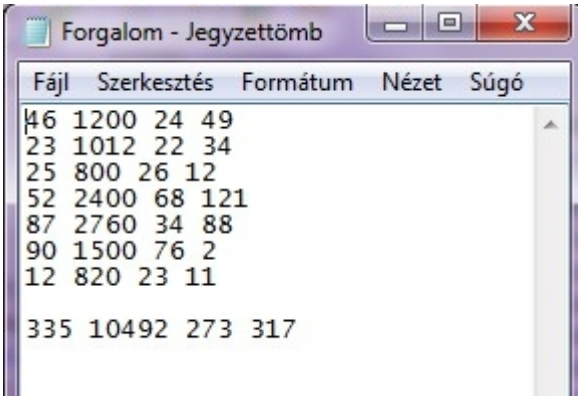


Az összeg kiírása

A forgalmi adatok összegezését természetesen az összegezés tételével végezhetjük el, ezúttal a négy adat egy vektor négy eleme lesz. Az adatokat hozzá kell írnunk a már meglévő fájlhoz, így azt *Append* módban nyitjuk meg. Hozzáíráskor a következő adat az utolsó bájt végére kerül, így ezúttal szükséges előtte egy üres *WriteLine* is, ami új sort kezd:

```
Sub Kiírás()
    Dim fájlki As FileStream = New FileStream("D:\Előadás
    anyagok\PROGRAMOZÁS\Gyakorlat\12.
    évfolyamnak\Fájlkezelés\Forgalom.txt", FileMode.Append)
    Dim író As New StreamWriter(fájlki)
    Dim összes(4) As Integer
    For i As Integer = 1 To 7
        For j As Integer = 1 To 4
            összes(j) += forgalom(i, j)
        Next
    Next
    író.WriteLine()
    író.WriteLine(összes(1) & " " & összes(2) & " " & összes(3) & "
    " & összes(4))
    író.Close()
    fájlki.Close()

End Sub
```



A program helyes lefutása esetén a fájl egy új sorral bővül.

Feladatok:

1) Átalakítás

Készítsünk programot, amely egy magyar nyelvű szövegből eltávolítja a szóközöket és írásjeleket, valamint a betűket az angol ábécé megfelelő nagybetűs változatára alakítja!

2) Lottó

Keressük meg az interneten a korábbi lottószámokat Excel formátumban! Mentsük el a kihúzott számokat CSV kiterjesztéssel, majd olvassuk be az adatokat, és határozzuk meg, melyik számot hányszor húzták ki! Melyik az 5 legritkábban kihúzott szám? Érdemes-e ezekre tippelnünk a jövő héten?

3) Osztálystatisztika

Rögzítsük egy osztály tanulóinak év végi osztályzatait tantárgyanként szóközökkel elválasztva egy szöveges állományban! Írjunk programot, amely a táblázat végére beszúrja a tantárgyi átlagokat!