

Windows alkalmazások készítése IV.

Ebben a fejezetben két olyan témát nézünk át (a fájlrendszer kezelése, a registry használata), amelyre egy nagyobb alkalmazás esetén szükségünk lehet. Érdemes a vázolt megoldási lehetőségeket áttanulmányozni, ám azok kipróbálásakor óvatosan járjunk el, mindkettővel könnyen tudunk kárt okozni.

A fájlrendszer kezelése

A fájlrendszert kezelő utasítások

A Windows fájlrendszerét a parancssori programokhoz hasonló utasításokkal kezelhetjük.

Egy adott fájl a `Kill` utasítással törölhetünk. Ehhez természetesen a fájl elérési útját és nevét ismernünk kell (kiválaszthatjuk pl. a `CommonDialog` vezérlővel). Például a `Kill "C:\Haha\Hehe.txt"` értelemszerűen az adott fájl törli. Használhatjuk egy karakter helyettesítésére a `?`, egy tetszőleges töredék helyettesítésére pedig a `*` dzsóker karaktert.

Egy üres mappa törlésére az `Rmdir` utasítás használható. Pl. a következő sor:
`Rmdir "C:\Haha\Ubul"` törli az `Ubul` nevű, üres mappát.

Egy új mappát az `Mkdir` utasítással hozhatunk létre. Pl. a `C:` meghajtón egy `ta` nevű mappát az `Mkdir "C:\ta"` utasítás hoz létre.

Az átnevezés illetve áthelyezés a `Name` utasítás segítségével hajtható végre. Értelemszerűen azonos elérési út esetén átnevezést, eltérő elérési út esetében áthelyezést jelent. Szintaxis: `Name <mit> As <mire>`

Példa: Rename

Következő példánkban egy mappa fájljait fogjuk átnevezni az eredeti névről `sielés-0.jpg`, `sielés-1.jpg`, `sielés-2.jpg` stb. nevekre. Az eljárás a képkategorizáló programok egyik rutinja lehet.

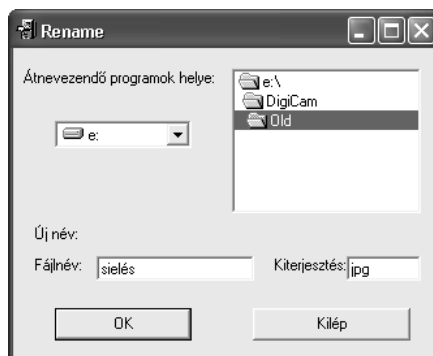
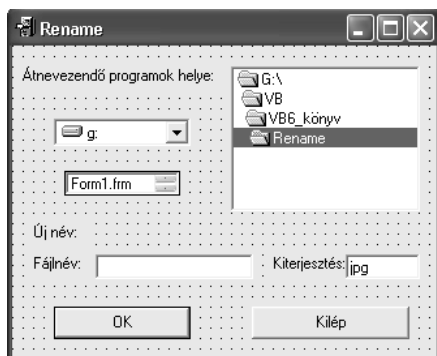
Helyezzünk el az űrlapon egy `DriveListBox`ot, pl. `Drive1` néven, egy `DirListBox`ot `Dir1` néven és egy `FileListBox`ot `File1` néven. Ez utóbbi `Visible` tulajdonságát állítsuk `False`-ra, ami azt jelenti, hogy a felhasználó nem fogja látni. A három lista segítségével választhatja ki a felhasználó, hogy melyik mappa fájljait kívánja átnevezni.

A következő ábra bal oldali részén az űrlapot tervezési nézetben, jobb oldali részén futás közben látjuk.

A három objektumot korábbi ismereteink alapján a következő kódrészlettel kapcsoljuk össze:

```
Private Sub Drive1_Change()  
    Dir1.Path = Drive1.Drive  
End Sub
```

```
Private Sub Dir1_Change()  
    File1.Path = Dir1.Path  
End Sub
```



A fenti kódrészlet garantálja, hogy az átnevezendő fájl neve a File1.List listában lesz (ez tartalmazza ugyanis a File1 vezérlő által mutatott fájlneveket), számuk pedig értelemszerűen File1.ListCount.

A felhasználó az új közös fájlnevet a txtName, a közös kiterjesztést pedig a txtKit nevű szövegmezőben adhatja meg (az ábrán a Fájlnév illetve a Kiterjesztés felirat utáni szövegmezők).

Maga az átnevezés az OK gombra kattintás (legyen a neve cmdOK) után hajtódik végre. A deklarációs részben definiált változók szerepe a következő. Az *strFO* egy vektor, amelybe beolvassuk az eljárás kezdetén az eredeti fájlneveket. Az *infF* változó fogja megadni a mappában lévő fájlok számát, ezen kívül szükségünk lesz egy *i* ciklusváltozóra is, ami végigfut majd a fájlokon:

```
Private Sub cmdOK_Click()  
    Dim strFO() As String  
    Dim intF As Integer  
    Dim i As Integer
```

A `Dim strFO() ...` deklaráció során ún. dinamikus tömböt hozunk létre. Ez azt jelenti, hogy a tömb tényleges méretét majd egy későbbi programrészben fogjuk megadni a `ReDim` utasítás segítségével.

Az eljárás első részében megkeressük a megadott mappa fájljait, és a fájlok nevét átmásoljuk az *strFO* tömbbe. Ekkor derül ki a fájlok tényleges száma (*intF* változó) is, és a dinamikus tömb méretét ennek megfelelően állítjuk be a `ReDim strFO(intF)` utasítással *intF* méretűre:

```
intF = File1.ListCount
ReDim strFO(intF)
For i = 0 To intF - 1
    strFO(i) = Dir1.Path & "\" & File1.List(i)
Next
```

A következő lépés az *strFO* tömbben felsorolt fájlok átnevezése. A fentiek alapján az alábbi ciklus már értelmezhető:

```
For i = 0 To intF - 1
    Name strFO(i) As Dir1.Path & "\" & txtName & _
        Str(i) & "." & txtKit
Next
```

A registry használata

A Windows rendszerben a regisztrációs adatbázis tartalmazza az operációs rendszer és az egyes programok beállításait. Ha szeretnénk alkalmazásunk adatait úgy elmenteni, hogy azokat a következő elindulásakor elérhessük, a mentendő adatokat programunknak is a registrybe kell írnia.

A registrybe írt adatokat a hozzájuk tartozó kulcs azonosítja, a kulcsok pedig szakaszokba szervezhetők. Ha például, ki szeretnénk menteni programunk főúrlapjának helyét, azaz bal felső sarkának koordinátáit (Top és Left), valamint szélességét (Width) és magasságát (Height), akkor ezeket egy szakaszba (Place) szervezhetjük.

A registrybe a `SaveSetting` utasítással írhatunk. Ennek négy paramétere a program neve, a szakasz neve, a kiírandó adatot azonosító kulcs és maga az adat. Például, ha az előzőekben említett program neve NL, akkor a főablak Top adatát a következő sor menti a registrybe:

```
SaveSetting ("NL", "Place", Left, frmNL.Top)
```

A fenti adat visszaolvasása a `GetSetting` függvénnyel történik. Ennek első három paramétere a program neve, a szakasz neve és a kulcs, ezek alapján adja vissza a megfelelő adatot. Negyedik paramétere egy ún. default érték, melyet akkor ad vissza, ha az adott kulcs hiányzik (pl. első indításkor):

```
frmNL.Top = GetSetting ("NL", "Place", Top, 0)
```

A registryből egyszerre egy egész szekciót törölhetünk a `DeleteSetting` utasítással. Az előző példa `Place` szekciójának törlése például:

```
DeleteSetting("NL", "Place")
```

A példák alapján látható, hogy nem írhatunk a registry tetszőleges ágába, a VB valamennyi adatot egy adott részbe ment.

Feladatok

1. Egérkezelés. Hogyan lehetne a rajzoló programunkat úgy elkészíteni, hogy ne folytonos vonalat húzzon, hanem sokszögvonalat?

2. Egészítsük ki a rajzoló programot úgy, hogy a vonal vastagságát is be lehessen állítani az eszköztárról!

3. A képnéző program továbbfejlesztése. A képet ne az űrlap hátterébe, hanem egy azon elhelyezett (és az űrlappal együtt átméreteződő) `ImageBox`-ba olvassuk be! Ha az `ImageBox` `Stretch` tulajdonságát `True`-ra állítjuk, a kép felveszi az `ImageBox` méretét.

4. Mentés másként a szövegszerkesztő programunkban. Készítsük el a *Mentés másként* menüpontot az `NL` programunkban! Ezzel megoldhatjuk, hogy új szöveget is el tudjunk menteni.

5. Az átnevező program továbbfejlesztése. Az átnevező programunk most ..., `sielés-9.jpg`, `sielés-10.jpg`, ..., névre nevezi át a fájlokat. Oldjuk meg, hogy 10-nél több fájl esetén megjelenjen a vezető nulla is, így: `sielés-09.jpg`! Általánosítsuk a problémát úgy, hogy 100 fájl esetén is háromjegyű legyen minden fájl sorszáma!

6. `Commander` készítése. Készítsünk két ablakos programot a fájlkezeléshez! Mindkét ablakban egy-egy mappa tartalma jelenjen meg, a fájlokat pedig kiválasztás után egy adott gombbal lehessen a két mappa között mozgatni. Ha pedig egy szöveges állomány nevére kattintunk, jelenjen meg annak tartalma!