Alapok

.NET Framework C# projekt létrehozása

References jobb klikk, Add reference, Microsoft Excel object … library pipa, aztán OK

Összes használt namespace:

* using System;
* using System.Collections.Generic;
* using System.Linq;
* using System.Text;
* using System.Threading.Tasks;
* using Microsoft.Office.Interop.Excel;
* using System.IO;
* using System.Text.RegularExpressions;

Excel file-ok megnyitására példányosítani kell:

* \_Application excel = new Application();

Létezik workbook, illetve worksheet

* Workbook wb;
* Worksheet ws;

Egyszerű értékadás:

* wb = excel.Workbooks.Open(@"C:\mappa\...\excelneve.xlsx");
* ws = wb.Worksheets[1]; a wb projekt 1. worksheete lesz, 1-től kezdi a számozást, nem 0-tól

Fájl bezárása:

* wb.Close(); rákérdez, hogy rámentsen-e, ha egymás után több fájlt kezelünk akkor idegesítő lehet a popup
* wb.Close(0); nem ment rá
* wb.Close(1); rámenti

Fájl mentése másként:

* wb.SaveCopyAs(@"C:\mappa\...\ujexcelneve.xlsx");

Példaprogram

A következőkben végigmegyünk egy mappa tartalmán, megnyitjuk egymás után az excel fileokat, átrakjuk az összes értéket az első worksheetre, majd kiolvassuk az adatokat.

Ha simán akarnál végigmenni egy mappa tartalmán és az összes excel filet egyenként megnyitni, csinálni vele valamit, majd bezárni, akkor errort adna ki, mivel az excel a megnyitás előtt csinál mindegyik excel fájlhoz egy üres temporary verziót, ~$fájlnév.xlxs névvel, ezért csinálunk egy regexet, ami kiszűri ezeket.

public void readInFiles(){

var allFiles = Directory.EnumerateFiles(@"C:\mappa\...\célmappa");

Regex fileFilter = new Regex(@"^(.)\*?\\[^~$]+.xlsx$");

foreach (string file in allFiles)

{

if (fileFilter.IsMatch(file))

{

wb = excel.Workbooks.Open(file);

//valamit csinálunk a file excellel pl

mergeSheets();

ws = wb.Worksheets[1];

writeOutData();

wb.Close(0);

}

}

}

MergeSheets()

Hogy könnyebb legyen a workbookból kiolvasni, ne kelljen ha egy worksheet végére értünk átállni a következő worksheetre, ezért az összes worksheet tartalmát átmásoljuk az elsőre, hogy csak a worksheet[1]-et kelljen kezelnünk (később ws változó).

Végigmegyünk a worksheeteken a másodiktól kezdve:

* for (int i = 2; i < wb.Worksheets.Count+1; i++)

{

//csinálunk valamit

}

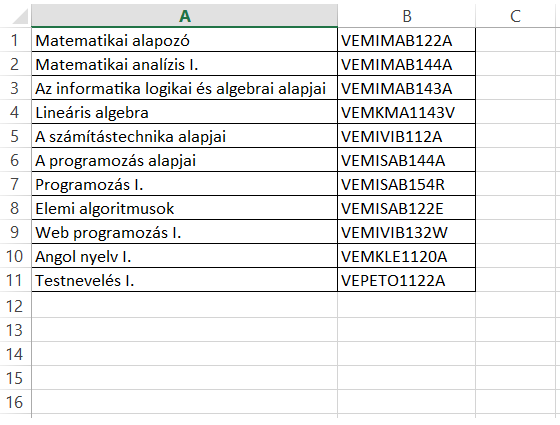
Minden ciklusban meg kell állapítani a következőket:

Az 1. sheet jelenlegi adatait:

* int goalSheetRow = getRowNum(wb.Worksheets[1]);

ez első sheet utolsó használt sora, ehhez majd hozzá kell adni egyet, és hozzá adni az A betűt, ide fogjuk másolni az i.sheetet

* string goalSheetStart = "A" + (goalSheetRow + 1);



goalSheetRow

goalSheetStart

Az i. sheet adatait:

* int sourceSheetRow = getRowNum(wb.Worksheets[i]);
* int sourceSheetColumn = getColNum(wb.Worksheets[i]);

Az i.sheetnél kombináljuk a fenti 2 adatot, hogy megkapjuk az utolsó használt cella nevét, itt pl B+12=B12

* string sourceSheetEnd = GetExcelColumnName(sourceSheetColumn) + sourceSheetRow;



sourceSheetRowwww

sourceSheetColumn

sourceSheetEnd

Végül kombináljuk a következő függvény segítségével:

* wb.Worksheets[i].Range["A1", sourceSheetEnd].Copy(wb.Worksheets[1].Range[goalSheetStart]);

Ennek a jelentése:

melyikmunkalapből.Range[mettől, meddig].Copy(melyikmunkalapba.Range[honnantólkezdve]);

getRowNum függvény kifejtése, ami csak egy másik beépitett függvényt meghiv és visszaad egy számot, de kitettük külön függvénybe, hogy átláthatóbb legyen.

Fontos a végén a .Row, ezért adja vissza az utolsó nem üres sort.

public int getRowNum(Worksheet sheet){

int result=sheet.Cells.Find("\*", System.Reflection.Missing.Value,

System.Reflection.Missing.Value, System.Reflection.Missing.Value, XlSearchOrder.xlByRows, XlSearchDirection.xlPrevious, false, System.Reflection.Missing.Value, System.Reflection.Missing.Value).Row;

return result;

}

Ugyanigy a getColNum kifejtése, amit ugyanazt csinálja, de a végén egy .Column van

public int getColNum(Worksheet sheet){

int result=sheet.Cells.Find("\*", System.Reflection.Missing.Value,

System.Reflection.Missing.Value, System.Reflection.Missing.Value, XlSearchOrder.xlByColumns, XlSearchDirection.xlPrevious, false, System.Reflection.Missing.Value, System.Reflection.Missing.Value).Column;

return result;

}

Ez a két függvény megkeresi az első olyan sort/oszlopot, ami üres, szóval, ha valamiféleképpen a táblázatunkban lenne pl egy üres sor/oszlop, utána pedig megint adatok, akkor azt a részét levágná, erre figyelni kell

a GetExcelColumnName egy netről vett függvény, ami visszaadja az adott szám alapján a megfelelő betűt, pl 1=A, 2=B…

private string GetExcelColumnName(int columnNumber){

int dividend = columnNumber;

string columnName = String.Empty;

int modulo;

while (dividend > 0){

modulo = (dividend - 1) % 26;

columnName = Convert.ToChar(65 + modulo).ToString() + columnName;

dividend = (int)((dividend - modulo) / 26);

}

return columnName;

}

writeOutData();

Lekérjük, hogy az adott excel 1.sheetje hol ér véget, végigmegyünk rajta, és ha az adott sor első és második értéke nem nulla, akkor kiirjuk őket, ez azért fontos, mert ha valami nulla lenne, és ki akarnánk iratni, akkor errot kapnánk.

public void writeOutData(){

int lastRow = getRowNum(ws);

for (int currentRow = 1; currentRow < lastRow+1; currentRow++){

if (ws.Cells[currentRow, 1].Value2 != null && ws.Cells[currentRow, 2].Value2 != null){

Console.WriteLine(ws.Cells[currentRow, 1].Value2+" "+ ws.Cells[currentRow, 2].Value2);

}

}

}

Kérdések:

1. Miért fontos, hogy ha egy mappa tartalmából az összes excel fájlt szeretnénk kezelni, hogy egy regexszel kiszűrjük őket, akkor is, hogy ha csak excel fájlokat tartalmaz a mappa?
2. Az Excelben 0-tól kezdődik a cellák és a worksheetek számozása is.

Igaz

Hamis

1. Hogyan kell bezárnunk egy excel file-t, ha azt akarjuk, hogy a program automatikusan bezárja azt, és ne mentsen rá?

wb.Close();

wb.Close(1);

wb.Close(0);

1. Ha A worksheetről szeretnénk valamit átmásolni B worksheetre, akkor melyiket kell választanunk?

A.Range[…].Copy(B.Range[…]);

B.Range[…].Copy(A.Range[…]);

1. Mire kell vigyáznunk, hogy ha worksheet.Cells.Find("\*", System.Reflection.Missing.Value,System.Reflection.Missing.Value, System.Reflection.Missing.Value, XlSearchOrder.xlByRows, XlSearchDirection.xlPrevious, false, System.Reflection.Missing.Value, System.Reflection.Missing.Value).Row; függvénnyel akarjuk meghatározni egy worksheet legutolsó nem üres sorát?
2. A Range függvény használatánál hogyan kell megadni a határokat?

A sorokat és oszlopokat is számként

Az oszlopokat betűként és sorokat számként

Az oszlopokat számként és sorokat betűként

1. A Microsoft Excel Object Library-d verziója attól függ, hogy az Office/Excel milyen verziója van a gépeden.

Igaz

Hamis

1. Miért fontos, hogy mielőtt kiiratnánk egy cella tartalmát, ellenőrizzük, hogy az nem üres-e?
2. Az új projekt készítésénél melyik verziót érdemes választani:

.NET Core

.NET Framework

.NET Standard

1. A fejlesztői környezet felismeri Microsoft.Office.Interop.Excel; namespace-t ha van Office telepítve a gépre.

Igaz

Hamis