

Window Presentation Foundation

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- <http://www.miwilhelm.de>
- Raum 2.202
- Tel. 03943 / 659 338

. Inhalt

- Einführung WPF
- Layouts
- C# Sprache
- Dialog-Elemente, Menüs
- 2D- / 3D-Grafik, Audio, Video, Animation
- **Office**
- Routen Events, Dependency Properties, Command
- Textdarstellung (Flow-FixedDocuments)
- Datenbanken

Office-Anbindung

■ Mögliche Varianten

- Über eine COM-Schnittstelle
- Direktes Schreiben einer Office-Datei

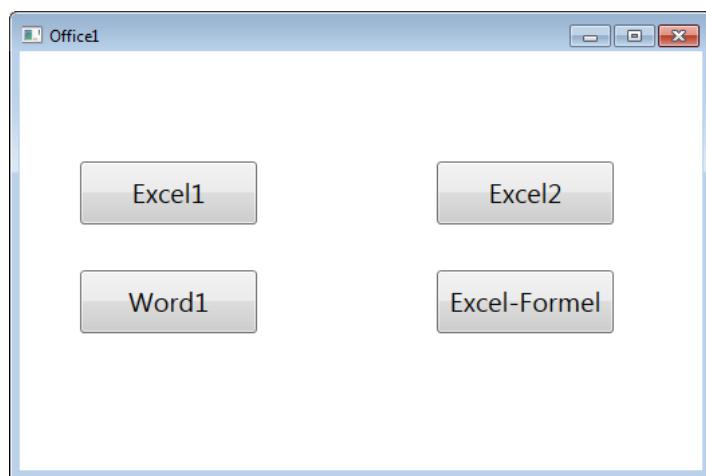
■ COM-Schnittstelle

- Über eine COM-Schnittstelle
- Ein COM-Objekt ist ein öffentliches Objekt im Betriebssystem vorhanden
- "Globale Variable"
- Hat Methoden zur Steuerung

■ Natives Schreiben mittels BIFF

- Binary Interchange FileFormat

Beispiel: COM-Schnittstelle mit WPF



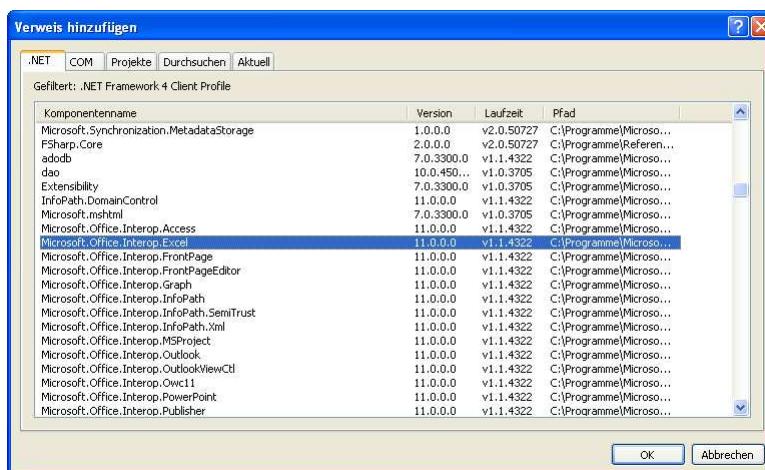
COM-Schnittstelle: Beispiel Excel

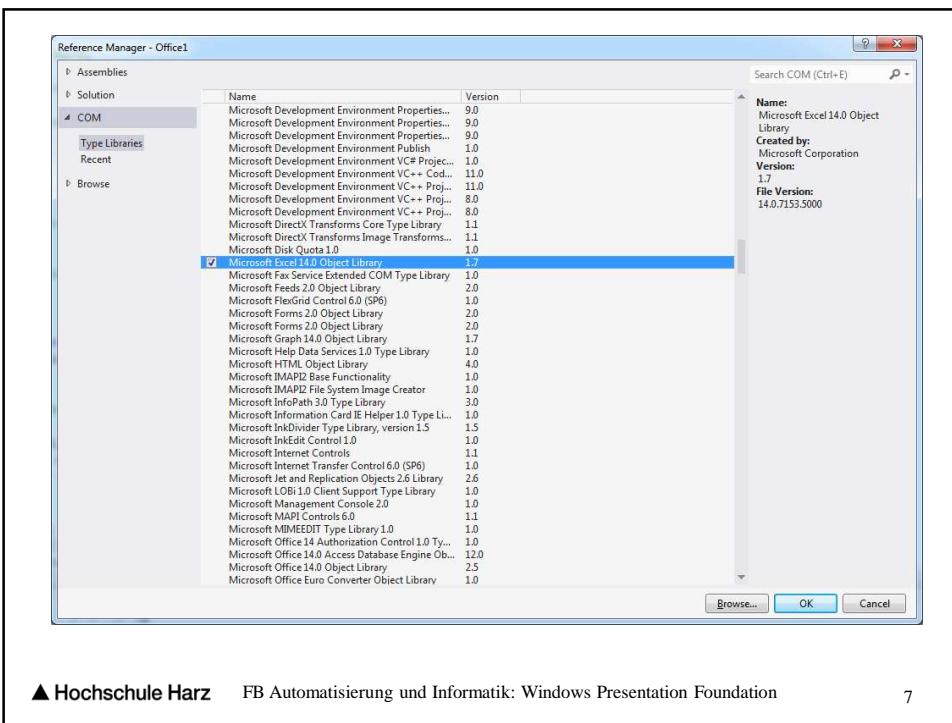
■ Ablauf

- o Projekt erstellen, WPF
- o Verweis zum COM-Objekt importieren
 - Projektfenster öffnen
 - Rechte Maustaste, Eintrag: Verweise, References
 - Register ".NET"
 - Eintrag "Microsoft.Office.Interop.Excel"
 - Eintrag "Microsoft.Office.Interop.Word"
- o Testschalter einfügen
- o Objekt erzeugen (Excel und Winword)
- o Methoden zur Steuerung und zum Einfügen

Verwendung der COM-Schnittstelle ist langsam

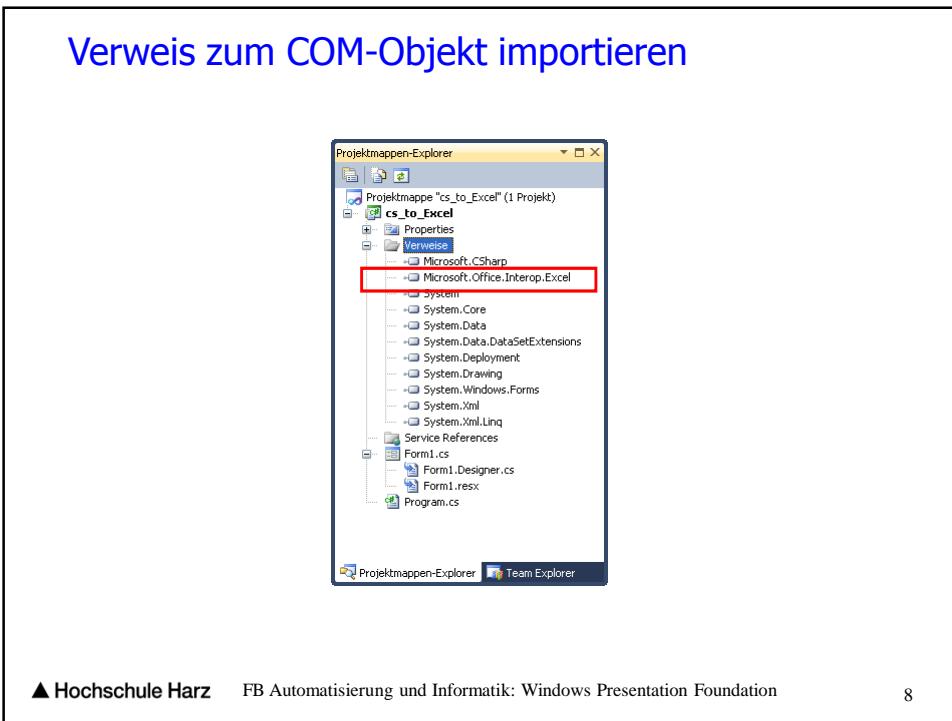
Verweis zum COM-Objekt importieren





▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

7



▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

8

1. Beispiel

```
var xl = new Excel.Application();
xl.Workbooks.Add();
xl.Visible=true;

xl.Cells[1,1] = "Zahl1";//Row,Column
xl.Cells[1,2] = "Zahl2";
xl.Cells[2,1] = "123";
xl.Cells[2,2] = "345";
xl.Cells[3,1] = "456";
xl.Cells[3,2] = "678";
xl.Cells[4,1].Select();

•using Excel = Microsoft.Office.Interop.Excel;
```

▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

9

Adressierung mittels [i,j]: Verweis „System.Drawing“

```
xl.Cells[1, j].Select();
xl.ActiveCell.Interior.Color = System.Drawing.Color.Yellow; //
xl.ActiveCell.Font.Color = System.Drawing.Color.Red;

xl.ActiveCell.Font.Name = "Times New Roman";
xl.ActiveCell.Font.Size = 14;
xl.ActiveCell.Font.Bold = true;

// Hintergrund: Benannte Parameter, Reihenfolge egal
xl.ActiveCell.BorderAround(
    LineStyle: Excel.XLLineStyle.xlDash,
    Weight: Excel.XIBorderWeight.xlMedium
);
```

▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

10

Adressierung mittels Range, Verweis „System.Drawing“

```
xl.Range["A1:J1"].Font.Name = "Verdana";
xl.Range["A1:J1"].Font.Color = System.Drawing.Color.Red;
xl.Range["A1:J1"].Font.Size = 14;
xl.Range["A1:J1"].Font.Bold = true;

xl.Range["A1:J1"].VerticalAlignment =
    Excel.XlVAlign.xlVAlignCenter;
xl.Range["A1:J1"].HorizontalAlignment =
    Excel.XlHAlign.xlHAlignCenter;

xl.Range["A1:J1"].BorderAround(
    LineStyle: Excel.XlLineStyle.xlDash,
    Weight: Excel.XlBorderWeight.xlMedium);
```

1. Aufgabe

- Erstellen eines neuen Projektes
 - Einbau zweier Schalter
-
- 1. Schalter:
 - Test der einfachen Ausgabe, siehe 1. Beispiel
 - 2. Schalter:
 - Aufbau einer Tabelle mit Header und Inhalt
 - Header mit "Zahl1 bis Zahl10"
 - Inhalt: 200 Zeilen
 - Alle Zellen, deren Summe bzgl. Col und Row mod 50 Null ist, sollen mit roter Hintergrundfarbe gezeichnet werden
 - **xl.ActiveCell.Interior.Color = 255;**

	A	B	C	D	E	F	G	H	I	J	K
1	Zahl1	Zahl2	Zahl3	Zahl4	Zahl5	Zahl6	Zahl7	Zahl8	Zahl9	Zahl10	
2	7	7	7	7	7	7	7	7	7	7	7
3	8	8	8	8	8	8	8	8	8	8	8
4	9		9	9	9	9	9	9	9	9	9
5	10	10	10	10	10	10	10	10	10	10	10
6	11	11	11	11	11	11	11	11	11	11	11
7	12	12	12	12	12	12	12	12	12	12	12
8	13	13	13	13	13	13	13	13	13	13	13
9	14	14	14	14	14	14	14	14	14	14	14
10	15	15	15	15	15	15	15	15	15	15	15
11	16	16	16	16	16	16	16	16	16	16	16
12	17	17	17	17	17	17	17	17	17	17	17
13	18	18	18	18	18	18	18	18	18	18	18
14	19	19	19	19	19	19	19	19	19	19	19
15	20	20	20	20	20	20	20	20	20	20	20
16	21	21	21	21	21	21	21	21	21	21	21
17	22	22	22	22	22	22	22	22	22	22	22
18	23	23	23	23	23	23	23	23	23	23	23
19	24	24	24	24	24	24	24	24	24	24	24
20	25	25	25	25	25	25	25	25	25	25	25
21	26	26	26	26	26	26	26	26	26	26	26
22	27	27	27	27	27	27	27	27	27	27	27
23	28	28	28	28	28	28	28	28	28	28	28
24	29	29	29	29	29	29	29	29	29	29	29
25	30	30	30	30	30	30	30	30	30	30	30
26	31	31	31	31	31	31	31	31	31	31	31
27	32	32	32	32	32	32	32	32	32	32	32
28	33	33	33	33	33	33	33	33	33	33	33

	A	B	C	D	E	F	G	H	I	J	K
29	34	34	34	34	34	34	34	34	34	34	34
30	35	35	35	35	35	35	35	35	35	35	35
31	36	36	36	36	36	36	36	36	36	36	36
32	37	37	37	37	37	37	37	37	37	37	37
33	38	38	38	38	38	38	38	38	38	38	38
34	39	39	39	39	39	39	39	39	39	39	39
35	40	40	40	40	40	40	40	40	40	40	40
36	41	41	41	41	41	41	41	41	41	41	41
37	42	42	42	42	42	42	42	42	42	42	42
38	43	43	43	43	43	43	43	43	43	43	43
39	44	44	44	44	44	44	44	44	44	44	44
40	45	45	45	45	45	45	45	45	45	45	45
41	46	46	46	46	46	46	46	46	46	46	46
42	47	47	47	47	47	47	47	47	47	47	47
43	48	48	48	48	48	48	48	48	48	48	48
44	49	49	49	49	49	49	49	49	49	49	49
45	50	50	50	50	50	50	50	50	50	50	50
46	51	51	51	51	51	51	51	51	51	51	51
47	52	52	52	52	52	52	52	52	52	52	52
48	53		53	53	53	53	53	53	53	53	53
49	54		54	54	54	54	54	54	54	54	54
50	55		55	55	55	55	55	55	55	55	55
51	56		56	56	56	56	56	56	56	56	56
52	57		57	57	57	57	57	57	57	57	57
53	58		58	58	58	58	58	58	58	58	58
54	59		59	59	59	59	59	59	59	59	59
55	60		60	60	60	60	60	60	60	60	60
56	61		61	61	61	61	61	61	61	61	61
57	62		62	62	62	62	62	62	62	62	62

COM-Objekt benutzen

```
private void bnExcel1_Click(object sender, RoutedEventArgs e) {
{
    var xl = new Excel.Application();
    xl.Workbooks.Add();
    xl.Visible = true; // Absturz bei größeren Zeilen
    xl.Cells[1, 1] = "Zahl1"; // Row, Column
    xl.Cells[1, 2] = "Zahl2";

    xl.Cells[2, 1] = "123";
    xl.Cells[2, 2] = "345";

    xl.Cells[3, 1] = "456";
    xl.Cells[3, 2] = "678";
    xl.Cells[4, 1].Select();
    xl.Visible = true; // hier kein Absturz bei größeren Zeilen
}
```

▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

15

COM-Objekt benutzen

```
private void button2_Click(object sender, EventArgs e)
{
    var xl = new Excel.Application();
    xl.Workbooks.Add();
    for (int j = 1; j <= 10; j++)
    {
        xl.Cells[1, j] = "Zahl1"; // Row, Column
    }
    for (int i = 2; i < 2000; i++)
    {
        for (int j=1;j<=10;j++)
            xl.Cells[i, j] = i+5;
    }
    xl.Cells[4, 1].Select();
    xl.Visible = true;
}
```

▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

16

Weitere Methoden: Verweis „System.Drawing“

```
xl.Cells[1, j] = "Zahl"+j; // Row, Column  
xl.Cells[1, j].Select();  
xl.ActiveCell.Interior.Color = System.Drawing.Color.Yellow;  
// xl.ActiveCell.Interior.PatternColor = System.Drawing.Color.Red;  
xl.ActiveCell.Font.Color = System.Drawing.Color.Red;  
  
xl.ActiveCell.Font.Name = "Times New Roman";  
xl.ActiveCell.Font.Size = 14;  
xl.ActiveCell.Font.Bold = true; // Underline, etc.  
  
// Benannte Parameter, Reihenfolge egal  
xl.ActiveCell.BorderAround  
    LineStyle: Excel.XLLineStyle.xlDash,  
    Weight: Excel.XlBorderWeight.xlMedium);  
}  
xl.Columns[1].AutoFit();
```

LineStyle

- xlContinuous default
- xlDash
- xlDashDot
- xlDashDotDot
- xlDot
- xlDouble
- xlLineStyleNone
- xlSlantDashDot
- xlLineStlyeNone

ColorIndex

- xlColorIndexAutomatic default
- xlColorIndexNone

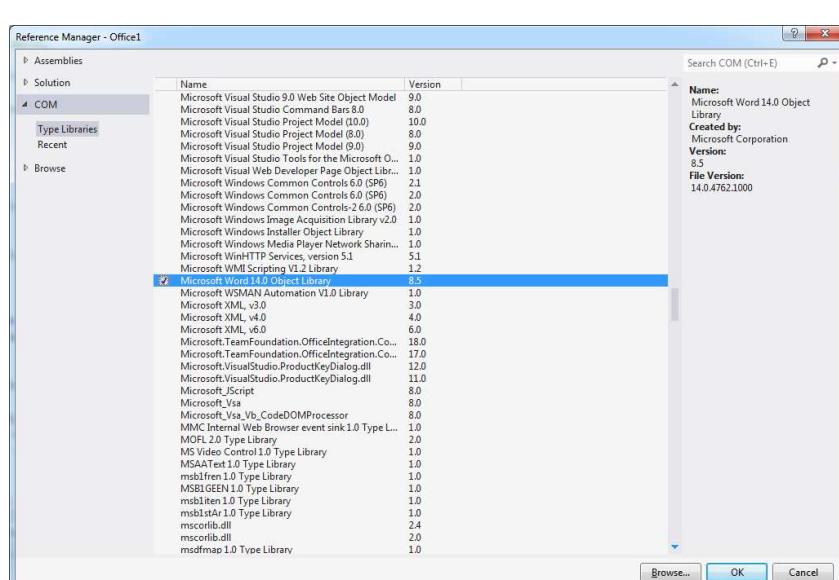
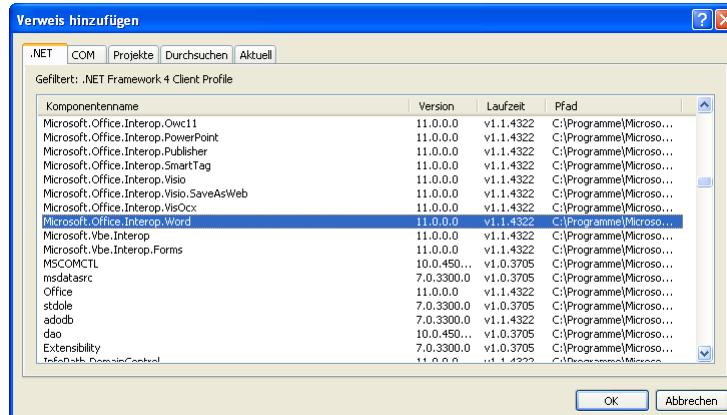
Color

- Konstante
- RGB-Wert

Weight

- xlHairline
- xlMedium
- xlThick
- xlThin default

Winword



Winword

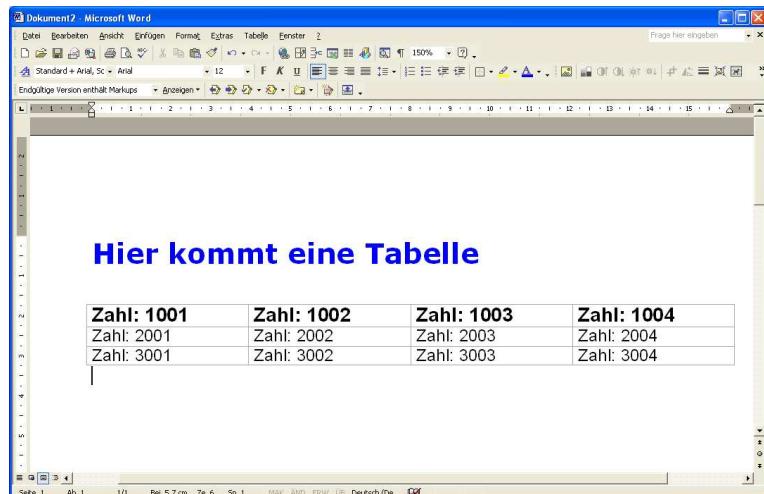
```
using Word = Microsoft.Office.Interop.Word;
```

```
var word = new Word.Application();
word.Visible = true;
word.Documents.Add();
word.Selection.TypeText("Hello World"); // Eingabetext
word.Selection.TypeParagraph();
word.Selection.Font.Color = Word.WdColor.wdColorRed;
word.Selection.Font.Name = "Times New Roman";
word.Selection.Font.Size = 14;
word.Selection.Font.Bold = 1;
word.Selection.TypeText("C Sharp ist toll"); // Eingabetext
```

Winword: 1. Beispiel

```
word.Selection.TypeParagraph();
word.Selection.TypeParagraph();
word.Selection.Font.Name = "Verdana";
word.Selection.Font.Color = Word.WdColor.wdColorBlue;
word.Selection.Font.Size = 20;
word.Selection.Font.Bold = 1;
word.Selection.Font.Underline =
    Microsoft.Office.Interop.Word.WdUnderline.wdUnderlineDash;
word.Selection.TypeText("Winword wird gesteuert\t1\t2\t3"); // Eingabetext
```

Winword: 2. Beispiel



▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

23

```
var word = new Word.Application();
word.Visible = true;
Microsoft.Office.Interop.Word.Document doc = word.Documents.Add();
word.Selection.Font.Name = "Verdana";
word.Selection.Font.Color = Word.WdColor.wdColorBlue;
word.Selection.Font.Size = 20;
word.Selection.Font.Bold = 1;
word.Selection.TypeText("Hier kommt eine Tabelle"); // Eingabetext
word.Selection.TypeParagraph();
word.Selection.TypeParagraph();
```

▲ Hochschule Harz FB Automatisierung und Informatik: Windows Presentation Foundation

24

```

word.Selection.Font.Name = "Arial";
word.Selection.Font.Color = Word.WdColor.wdColorBlack;
word.Selection.Font.Size = 12;
word.Selection.Font.Bold = 0;
Word.Range tableLocation = doc.Range(word.Selection.Start, word.Selection.Start);
Microsoft.Office.Interop.Word.Table table = doc.Tables.Add(Range: tableLocation,
NumRows: 3, NumColumns: 4);

// Überschrift-Format
for (int col = 1; col <= 4; col++)
{
    Word.Cell cell = table.Cell(1, col);
    cell.Range.Font.Bold = 1;
    cell.Range.Font.Size=14;
}
for (int row = 1; row <= 3; row++)
{
    for (int col = 1; col <= 4; col++)
    {
        Word.Cell cell = table.Cell(row, col);
        cell.Range.Text = "Zahl: "+row*100+col;
    }
}

```

wdUnderlineDash	Dashes.
wdUnderlineDashHeavy	Heavy dashes.
wdUnderlineDashLong	Long dashes.
wdUnderlineDashLongHeavy	Long heavy dashes.
wdUnderlineDotDash	Alternating dots and dashes.
wdUnderlineDotDashHeavy	Alternating heavy dots and heavy dashes.
wdUnderlineDotDotDash	An alternating dot-dot-dash pattern.
wdUnderlineDotDotDashHeavy	An alternating heavy dot-dot-dash pattern.
wdUnderlineDotted	Dots.
wdUnderlineDottedHeavy	Heavy dots.
wdUnderlineDouble	A double line.
wdUnderlineNone	No underline.
wdUnderlineSingle	A single line. <i>default</i> .
wdUnderlineThick	A single thick line.
wdUnderlineWavy	A single wavy line.
wdUnderlineWavyDouble	A double wavy line.
wdUnderlineWavyHeavy	A heavy wavy line.
wdUnderlineWords	Underline individual words only.

Excel durch das Binary Interchange FileFormat (BIFF)

- Das BIFF ist ein Format, welches Excel als Grunddateiformat verwendet
- Es gibt eine Schnittstelle, die elegant die "Schnelligkeit" der COM-Schnittstelle umgeht
- Hier wird direkt eine EXCEL-Datei geschrieben
- Dateiname: CExport2EXCEL.cs

Beispiel:

```
CExport2EXCEL exportexcel = new CExport2EXCEL("bsp1.xls");
exportexcel.openFile();
exportexcel.Write(1, 2, 1234); // col row int
exportexcel.closeFile();
```

```
Process Prog = new Process();
Prog.StartInfo.FileName = "excel.exe";
Prog.StartInfo.Arguments = "bsp1.xls";
Prog.Start();
```

• using System.Diagnostics; // run

3. Aufgabe: BIFF

- Erstellen eines neuen Projektes
 - Einbau zweier Schalter
-
- **1. Schalter:**
 - Schreiben einer Excel-Datei mit Ausgabe von int und double Zahlen
 - **2. Schalter:**
 - Schreiben einer Excel-Datei mit Ausgabe von int, double und Strings
 - 01.01.2010 (Datumsformat mit Copy-Paste)
 - 01/01/2010

Links

• <http://msdn.microsoft.com/de-de/library/ms262200%28v=Office.11%29.aspx>