

Листинг функции FileInfo

```
private void FileInfo()
{
    ApplicationClass spWordApp = new ApplicationClass();
    spWordApp.Visible = false;
    object missing = System.Reflection.Missing.Value;
    int nomer = 0;
    string str;
    try
    {
for (int i = 0; i < fileData.Count; i++)
    {
        nomer = i;
        fileData[i].Status = "анализ";
        FillList();
        object fileName = fileData[i].FullPath;
        object readOnly = false;
        object isVisible = false;
        Document spDoc = spWordApp.Documents.Open(ref fileName, ref missing, ref
readOnly,
ref missing, ref missing, ref missing,
ref missing, ref missing, ref missing,
ref missing, ref missing, ref isVisible,
ref missing, ref missing, ref missing, ref missing);
        spDoc.Activate();
        if (!doThread) { spDoc.Close(ref missing, ref missing, ref missing); break; }
        // количество страниц
        fileData[i].Str =
spWordApp.ActiveDocument.ComputeStatistics(WdStatistic.wdStatisticPages, ref
missing);
        if (!doThread) { spDoc.Close(ref missing, ref missing, ref missing); break; }
        //количество строк
        fileData[i].Strok =
spWordApp.ActiveDocument.ComputeStatistics(WdStatistic.wdStatisticLines, ref
missing);
        if (!doThread) { spDoc.Close(ref missing, ref missing, ref missing); break; }
        //количество слов
        fileData[i].Slov =
spWordApp.ActiveDocument.ComputeStatistics(WdStatistic.wdStatisticWords, ref
missing);
        if (!doThread) { spDoc.Close(ref missing, ref missing, ref missing); break; }
        //общее количество символов
        fileData[i].All = spWordApp.ActiveDocument.Characters.Count;
        if (!doThread) { spDoc.Close(ref missing, ref missing, ref missing); break; }
        //определение количества букв, абзацев, пробелов
```

```

object start = 0;
object end = fileData[i].All;
string allText = spWordApp.ActiveDocument.Range(ref start, ref end).Text;
int bufProc = 0;
fileData[i].Bukv = 0;
fileData[i].Abz = 0;
fileData[i].Probel = 0;
int allTextLength = allText.Length;
for (int j = 0; j < allTextLength; j++)
{
if (char.IsLetter(allText[j])) { fileData[i].Bukv++; } // ТОЛЬКО БУКВЫ
else if (allText[j] == '\r') { fileData[i].Abz++; }
else if (allText[j] == ' ') { fileData[i].Probel++; }
if ((100 * j) / (allTextLength + 1) > bufProc)
{
bufProc = (100 * j) / (allTextLength + 1);
SetData(str, bufProc);
}if (!doThread) {break;} }
if (!doThread) { spDoc.Close(ref missing, ref missing, ref missing); break; }
spDoc.Close(ref missing, ref missing, ref missing);
fileData[i].Status = "ВЫПОЛНЕН";
fileData[i].Data = true;
FillList();
if (!doThread) { break; }
} }
catch (Exception ex)
{
fileData[nomer].Status = "ошибка доступа";
}
finally
{
if (!doThread)
{
fileData[nomer].Status = "ГОТОВ";
}
spWordApp.Application.Quit(ref missing, ref missing, ref missing);
ShowProgressBarAndListViews(1);
stateStopAnalyze = false;
doThread = false;
btnAnalyze.Text = "Анализировать";
}}

```

Листинг функции SetFile

```
private void SetFile()
{
    stegoData.Status = "запись";
    FillEmbedList(true);
    string str;
    // 1 этап - получение искомой записываемой байтовой последовательности
    byte[] buf;
    FileStream fstream = new FileStream(stegoData.FullPathStego, FileMode.Open,
    FileAccess.ReadWrite);
    buf = new byte[fstream.Length];
    fstream.Read(buf, 0, (int)fstream.Length);
    fstream.Close();
    SetData(str, 100);
    if (!doThread) { ShowProgressBarAndListView(1); }
    else
    {
        // 2 этап - получение полной битовой последовательности (согласно шапке)
        //реальный размер массива
        long realSize = (stegoData.SizeStego % stegoData.BitsPerSymbol() != 0) ?
        (((stegoData.SizeStego / stegoData.BitsPerSymbol()) + 1) * stegoData.BitsPerSymbol())
        : stegoData.SizeStego;
        byte[] full_mass = new byte[realSize];
        byte[] FirstSimbol = new byte[11];
        //запись 11 бит параметров
        FromIntToBinary(ref FirstSimbol, 3, stegoData.Kerning);
        FromIntToBinary(ref FirstSimbol, 5, stegoData.Scaling);
        FromIntToBinary(ref FirstSimbol, 7, stegoData.Spacing);
        FromIntToBinary(ref FirstSimbol, 10, stegoData.Color);
        //длина имени файла
        FromIntToBinary(ref full_mass, 8, stegoData.NameStego.Length);
        SetData(str, 100);
        int bufProc = 0;
        //запись имени
        int i = 8 + 16;
        int schet = 0;
        int LengthNameStego = stegoData.NameStego.Length;
        foreach (char z in stegoData.NameStego)
        {
            FromIntToBinary(ref full_mass, i, (int)z);
            i += 16;
            schet++;
            if ((100 * schet) / (LengthNameStego) > bufProc)
            {
```

```

bufProc = (100 * schet) / (LengthNameStego);
SetData(str, bufProc);
    }
    if (!doThread) { break; }
}
i += 3;
if (!doThread) { ShowProgressBarAndListViews(1); }
else
{
    bufProc = 0;

    //длина стегосообщения (Байт)
    FromIntToBinary(ref full_mass, i, (int)stegoData.SizeStegoFile);
    // стегосообщение в битовом виде
    int bufLength = buf.Length;
    for (int j = 0; j < bufLength; j++)
    {
i += 8;
FromIntToBinary(ref full_mass, i, buf[j]);

if ((100 * j) / (bufLength) > bufProc)
{
    bufProc = (100 * j) / (bufLength);
    SetData(str, bufProc);
}
if (!doThread) { break; }
}

    if (!doThread) { ShowProgressBarAndListViews(1); }
    else
    {
// 3 этап - скрывание данных в Word
ApplicationClass spWordApp = new ApplicationClass();
spWordApp.Visible = false;
object missing = System.Reflection.Missing.Value;
try
{
    object fileName = stegoData.FullPathContainer;
    object readOnly = false;
    object isVisible = false;
    Document spDoc = spWordApp.Documents.Open(ref fileName, ref missing, ref
readOnly,
ref missing, ref missing, ref missing,
ref missing, ref missing, ref missing,
ref missing, ref missing, ref isVisible,
ref missing, ref missing, ref missing, ref missing);

```

```

spDoc.Activate();
//общее количество символов
stegoData.Stego = spWordApp.ActiveDocument.Characters.Count;
if (doThread)
{
// проведение записи данных
int simbols = (int)(realSize / stegoData.BitsPerSymbol()); // количество символов
// первый символ
spWordApp.ActiveDocument.Characters[1].Font.Kerning =
FromBinaryToInt(FirstSimbol, 0, 10);
spWordApp.ActiveDocument.Characters[1].Font.Spacing =
FromBinaryToSpacing(FirstSimbol, 10, 1);

int perechod = 0;
bufProc = 0;

for (int j = 2; j <= simbols + 1; j++)
{
if (!doThread) { break; }
spWordApp.ActiveDocument.Characters[j].Font.Color =
FromBinaryToColor(full_mass, perechod, stegoData.Color,
spWordApp.ActiveDocument.Characters[j].Font.Color);
perechod += stegoData.Color * 3;
if (!doThread) { break; }
spWordApp.ActiveDocument.Characters[j].Font.Kerning =
FromBinaryToInt(full_mass, perechod, stegoData.Kerning);
perechod += stegoData.Kerning;
if (!doThread) { break; }
spWordApp.ActiveDocument.Characters[j].Font.Scoring =
FromBinaryToScoring(full_mass, perechod, stegoData.Scoring);
perechod += stegoData.Scoring;
if (!doThread) { break; }
spWordApp.ActiveDocument.Characters[j].Font.Spacing =
FromBinaryToSpacing(full_mass, perechod, stegoData.Spacing);
perechod += stegoData.Spacing;
if (!doThread) { break; }
if ((100 * j) / (simbols + 1) > bufProc)
{
bufProc = (100 * j) / (simbols + 1);
SetData(str, bufProc);
}}
if (doThread)
{
//имя сохранения
string saveAs = Path.GetDirectoryName(stegoData.FullPathContainer);

```

```

    saveAs += "\\\" + Path.GetFileNameWithoutExtension(stegoData.NameContainer) +
    \"_WithStego.docx\";
    object fileNameSave = saveAs;

    spDoc.SaveAs(ref fileNameSave, ref missing, ref missing, ref missing, ref missing, ref
    missing, ref missing, ref missing, ref missing, ref missing, ref missing, ref
    missing, ref missing, ref missing, ref missing);
    stegoData.Status = \"ВЫПОЛНЕН\";
    FillEmbedList(true);
}
}
object SaveChanges = false;
spDoc.Close(ref SaveChanges, ref missing, ref missing);
}
catch (Exception ex)
{
    stegoData.Status = \"ошибка доступа\";
}
finally
{
    spWordApp.Application.Quit(ref missing, ref missing, ref missing);
} } } }
if (!doThread)
{
    stegoData.Status = \"ГОТОВ\";
}
stateStopEmbed = false;
btnEmbed.Text = \"Встроить\";
doThread = false;
}
// mass - массив записи, start - точка старта записи - пишется справа - налево, z -
записываемое число
private void FromIntToBinary(ref byte[] mass, int start, int z)
{
    int kol = start;
    // перевести число в бинарный вид
    while (z / 2 >= 1)
    {
        mass[kol] = (byte)(z % 2); kol--;
        z = z / 2;
    }
    mass[kol] = (byte)z;
}
// mass - массив данных, start - точка старта счета, kol - количество разрядов
private int FromBinaryToInt(byte[] mass, int start, int kol)
{

```

```

    int rez = 0;
    for (int i = start; i < start + kol; i++)
    {
rez += mass[i] * (int)System.Math.Pow(2, ((start + kol - 1) - i));
    }
    return rez;
}
private int FromBinaryToScaling(byte[] mass, int start, int kol)
{
    int rez = 100 + FromBinaryToInt(mass, start, kol);
    return rez;
}
private float FromBinaryToSpacing(byte[] mass, int start, int kol)
{
    int tek = FromBinaryToInt(mass, start, kol);
    float rez = (tek % 2 == 0) ? ((float)-0.1 * ((float)(tek / 2))) : ((float)0.1 * ((float)((tek +
1) / 2)));
    return rez;
}
private WdColor FromBinaryToColor(byte[] mass, int start, int step, WdColor color)
//kol-содержит все 3
{
    // получить составляющие цвета
    int B = (((int)color) & 16711680) / 65536; // первая составляющая (FF0000)
    B &= 255 - (int)(System.Math.Pow(2, step) - 1); // очистка 11111111 - 111 =
11111000 - и - чистый шаблон
    B |= FromBinaryToInt(mass, start + 2 * step, step);
    int G = (((int)color) & 65280) / 256; // вторая составляющая (00FF00)
    G &= 255 - (int)(System.Math.Pow(2, step) - 1); // очистка 11111111 - 111 =
11111000 - и - чистый шаблон
    G |= FromBinaryToInt(mass, start + step, step);
    int R = ((int)color) & 255; // третья составляющая (0000FF)
    R &= 255 - (int)(System.Math.Pow(2, step) - 1); // очистка 11111111 - 111 =
11111000 - и - чистый шаблон
    R |= FromBinaryToInt(mass, start, step);
    return (WdColor)Microsoft.VisualBasic.Information.RGB(R, G, B);
}

```

Листинг функции GetFile

```
private void GetFile()
{
    stegoDataGet.Status = "чтение";
    string str;
    // 1 этап открыть документ Word
    ApplicationClass spWordApp = new ApplicationClass();
    spWordApp.Visible = false;
    object missing = System.Reflection.Missing.Value;
    try
    {
        object fileName = stegoDataGet.FullPathContainer;
        object readOnly = false;
        object isVisible = false;
        Document spDoc = spWordApp.Documents.Open(ref fileName, ref missing, ref
readOnly,
            ref missing, ref missing, ref missing,
            ref missing, ref missing, ref missing,
            ref missing, ref isVisible,
            ref missing, ref missing, ref missing, ref missing);
        spDoc.Activate();

        if (doThread)
        {
            int NumberOfAllSymbolsInFile = spWordApp.ActiveDocument.Characters.Count;
            SetData(" Получение сведений о файле", 100);
            if (doThread)
            {
                // получить параметры записи - 1 символ ++++++
                byte[] FirstSymbol = new byte[11];
                FromIntToBinary(ref FirstSymbol, 9,
                    (int)spWordApp.ActiveDocument.Characters[1].Font.Kerning);
                FromSpacingToBinary(ref FirstSymbol, 10,
                    spWordApp.ActiveDocument.Characters[1].Font.Spacing);
                stegoDataGet.Kerning = FromBinaryToInt(FirstSymbol, 0, 4);
                stegoDataGet.Scaling = FromBinaryToInt(FirstSymbol, 4, 2);
                stegoDataGet.Spacing = FromBinaryToInt(FirstSymbol, 6, 2);
                stegoDataGet.Color = FromBinaryToInt(FirstSymbol, 8, 3);
                //проверка на исключение
                if (stegoDataGet.BitsPerSymbol() > 31 || stegoDataGet.BitsPerSymbol() == 0) { throw
new Exception("Error"); }
                else
                {
                    // получить количество символов в файле ++++++
                    // количество бит в 1 символе
```

```

int kolBit = stegoDataGet.BitsPerSymbol();
// количество букв, которые содержат в себе 9 bit с размером имени файла
int CharactersWithFileSize = (9 % kolBit != 0) ? ((9 / kolBit) + 1) : (9 / kolBit);
//проверка на исключение
if ((CharactersWithFileSize + 1) > NumberOfAllSimbolsInFile) { throw new
Exception("Error"); }
else
{
// буфер, содержащий длину имени
int lenghtFileSize = CharactersWithFileSize * kolBit;
byte[] FileSize = new byte[lenghtFileSize];
int perechod = -1;
if (!ReadFromWord(2, CharactersWithFileSize + 1, ref FileSize, perechod, spWordApp,
str))
{
// длина имени файла
int lenghtFileName = FromBinaryToInt(FileSize, 0, 9);

// получить имя файла
+++++
// количество букв, которые содержат в себе имена файла:
// поскольку часть битов уже лежала в буфере FileSize необходимо ее учесть
int CharactersWithFileName = ((lenghtFileName * 16 + 19 - (CharactersWithFileSize
* kolBit - 9)) % kolBit != 0) ? (((lenghtFileName * 16 + 19 - (CharactersWithFileSize *
kolBit - 9)) / kolBit) + 1) : ((lenghtFileName * 16 + 19 - (CharactersWithFileSize *
kolBit - 9)) / kolBit);
//проверка на исключение
if ((CharactersWithFileSize + 1 + CharactersWithFileName) >
NumberOfAllSimbolsInFile) { throw new Exception("Error"); }
else
{
int lenghtNameFile = CharactersWithFileName * kolBit + CharactersWithFileSize *
kolBit - 9;
byte[] NameFile = new byte[lenghtNameFile];
perechod = 0;
// переписать биты из старого массива в новый
for (int i = 9; i < lenghtFileSize; i++)
{
NameFile[perechod] = FileSize[i]; perechod++;
}
perechod--;

if (!ReadFromWord(CharactersWithFileSize + 2, CharactersWithFileSize + 1 +
CharactersWithFileName, ref NameFile, perechod, spWordApp, str))
{
//преобразовать полученное имя из буфера в массив char

```

```

char[] name = new char[lenghtFileName];

int bufProc = 0;

for (int i = 0; i < lenghtFileName; i++)
{
name[i] = (char)FromBinaryToInt(NameFile, 16 * i, 16);
if (!doThread) { break; }
if ((100 * i) / (lenghtFileName) > bufProc)
{
bufProc = (100 * i) / lenghtFileName;
SetData(str, bufProc);
}
}
if (doThread)
{
stegoDataGet.NameStego = new string(name);
// получить длину стегосообщения в Байтах
+++++
// 19 bit лежат в NameFile
int SizeStegoByte = FromBinaryToInt(NameFile, lenghtFileName * 16, 19);
stegoDataGet.SizeStego = SizeStegoByte * 8 + 9 + lenghtFileName * 16 + 19;
stegoDataGet.SizeStegoFile = SizeStegoByte;
//остаток данных в буфере NameFile
int RestBitsNameFile = lenghtNameFile - lenghtFileName * 16 - 19;

// получить стегосообщение +++++
int CharactersWithStego = ((SizeStegoByte * 8 - RestBitsNameFile) % kolBit != 0) ?
(((SizeStegoByte * 8 - RestBitsNameFile) / kolBit) + 1) : ((SizeStegoByte * 8 -
RestBitsNameFile) / kolBit);
//проверка на исключение
if ((CharactersWithFileSize + 1 + CharactersWithFileName + CharactersWithStego) >
NumberOfAllSimbolsInFile) { throw new Exception("Error"); }
else
{
byte[] StegoFile = new byte[CharactersWithStego * kolBit + RestBitsNameFile];
perechod = 0;
// переписать биты из старого массива в новый
for (int i = lenghtNameFile - RestBitsNameFile; i < lenghtNameFile; i++)
{
StegoFile[perechod] = NameFile[i]; perechod++;
}
perechod--;
if (!ReadFromWord(CharactersWithFileSize + 2 + CharactersWithFileName,
CharactersWithFileSize + 1 + CharactersWithFileName + CharactersWithStego, ref
StegoFile, perechod, spWordApp, str))

```

```

    {
// ВОССТАНОВИТЬ СТЕГОСООБЩЕНИЕ ++++++
bufProc = 0;
byte[] StegoFileByte = new byte[SizeStegoByte];
for (int i = 0; i < SizeStegoByte; i++)
{
    StegoFileByte[i] = (byte)FromBinaryToInt(StegoFile, 8 * i, 8);
    if (!doThread) { break; }
    if ((100 * i) / SizeStegoByte > bufProc)
    {
bufProc = (100 * i) / SizeStegoByte;
    }
}
if (doThread)
{
    string saveAs = Path.GetDirectoryName(stegoDataGet.FullPathContainer);
    saveAs += "\\ " + stegoDataGet.NameStego;

    FileStream fstream = new FileStream(saveAs, FileMode.Create,
FileAccess.ReadWrite);
    fstream.Write(StegoFileByte, 0, (int)StegoFileByte.Count());
    fstream.Close();

    stegoDataGet.Status = "ВЫПОЛНЕН";
    stegoDataGet.DataStego = true;
}}}}}}}}}}
}
object SaveChanges = false;
spDoc.Close(ref SaveChanges, ref missing, ref missing);
}
catch (Exception ex)
{
stegoDataGet.Status = "ошибка доступа";
}
finally
{
if (!doThread)
{
    stegoDataGet.Status = "ГОТОВ";
}
stateStopGetStego = false;
btnGetStegoFromWord.Text = "Получить";
doThread = false;
spWordApp.Application.Quit(ref missing, ref missing, ref missing);
ShowProgressBarAndListView(1);
}}

```

```

private void FromScalingToBinary(ref byte[] mass, int start, int kol)
{
    FromIntToBinary(ref mass, start, kol-100);
}
private void FromSpacingToBinary(ref byte[] mass, int start, float kol)
{
    int rez = (kol > 0) ? (((int)(kol * 20)) - 1) : ((int)(kol * (-20)));
    FromIntToBinary(ref mass, start, rez);
}
private void FromColorToBinary(ref byte[] mass, int start, int step, WdColor kol) //kol-
содержит все 3
{
    // получить составляющие цвета
    int B = (((int)kol) & 16711680) / 65536; // первая составляющая (FF0000)
    B &= (int)(System.Math.Pow(2, step) - 1);
    int G = (((int)kol) & 65280) / 256; // вторая составляющая (00FF00)
    G &= (int)(System.Math.Pow(2, step) - 1);
    int R = ((int)kol) & 255; // третья составляющая (0000FF)
    R &= (int)(System.Math.Pow(2, step) - 1); // очистка 11111111 & 111 = 111 - и -
ЧИСТЫЙ ШАБЛОН

    FromIntToBinary(ref mass, start, B);
    FromIntToBinary(ref mass, start - step, G);
    FromIntToBinary(ref mass, start - 2 * step, R);
}
private bool ReadFromWord(int start, int end, ref byte[] mass, int perechod,
ApplicationClass spWordApp, string str)
{
    int bufProc = 0;
    SetData(str, bufProc);
    for (int i = start; i <= end; i++)
    {
        if (!doThread) { return true; }
        if (stegoDataGet.Color != 0)
        {
            perechod += stegoDataGet.Color * 3;
            FromColorToBinary(ref mass, perechod, stegoDataGet.Color,
spWordApp.ActiveDocument.Characters[i].Font.Color);
        }
        if (!doThread) { return true; }
        if (stegoDataGet.Kerning != 0)
        {
            perechod += stegoDataGet.Kerning;
            FromIntToBinary(ref mass, perechod,
(int)spWordApp.ActiveDocument.Characters[i].Font.Kerning);
        }
    }
}

```

```

if (!doThread) { return true; }
if (stegoDataGet.Scaling != 0)
{
    perechod += stegoDataGet.Scaling;
    FromScalingToBinary(ref mass, perechod,
spWordApp.ActiveDocument.Characters[i].Font.Scaling);
}
if (!doThread) { return true; }
if (stegoDataGet.Spacing != 0)
{
    perechod += stegoDataGet.Spacing;
    FromSpacingToBinary(ref mass, perechod,
spWordApp.ActiveDocument.Characters[i].Font.Spacing);
}
if (!doThread) { return true; }

// ИСКЛЮЧИТЕЛЬНЫЙ СЛУЧАЙ: start==end
if (start == end) { SetData(str, 100); }
else
{
    if ((100 * (i - start)) / (end - start) > bufProc)
    {
bufProc = (100 * (i - start)) / (end - start);
    }
}
return false;
}

```