

テキスト・ファイル入出力

Visual C++2005/CLR の機能を用いてテキストファイルの入出力を行う例を作成した。入出力するファイル名は `OpenFileDialog` および `SaveFileDialog` を用いて行うと通常の Windows におけるファイル操作と同様にファイル名の取得ができる。これらはツールボックスからフォーム上に貼り付けて利用するが、フォームに貼り付けると他のクラス型のオブジェクトとは異なり、フォームの下に表示される (図 1)。

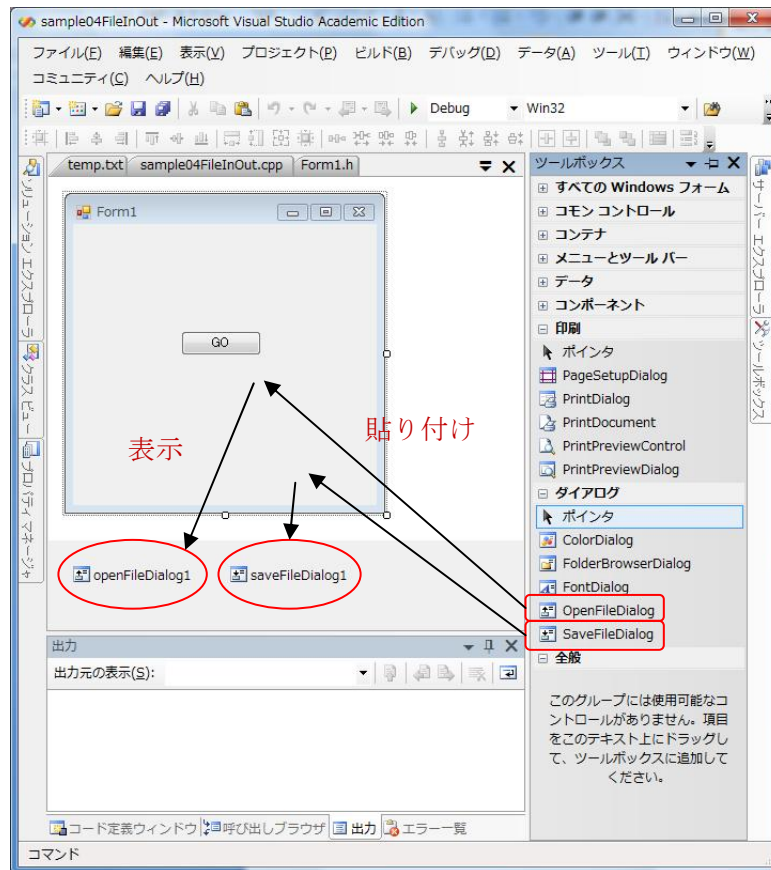


図 1 OpenFileDialog と SaveFileDialog の貼り付けとそれらの表示状態。

これらのダイアログは `ShowDialog()` メソッドによって提示される。設定したファイル名は `FileName` プロパティによって取得できる。

指定したファイル名からのテキストファイルの読み込みは `StreamReader` クラスを用いて行う。定義

```
StreamReader ^ 変数 = gcnew StreamReader(ファイル名、エンコーディング);
```

により、変数のメソッド `ReadLine()` によってテキストファイルから 1 行単位で読み込むこ

とができる。エンコーディングは日本語テキストファイルを扱うときは

```
GetEncoding("shift-jis")
```

を指定するが、名前空間も含めて指定するときは

```
System::Text::Encoding::GetEncoding("shift-jis")
```

となる。

読み込まれた 1 行内に複数のデータ (文字列) があるとき、それらの文字列を取り出すための関数

```
String ^ SeekStr( String ^ str, int pos );
```

 (1)

を用意した。この関数は myetc.h に宣言されているが、リスト 1 にその定義を示す。

リスト 1. 関数 SeekStr。

```
String ^ SeekStr( String ^ str, int pos ){
    String ^ vstr = "";
    array<wchar_t> ^ separator = { ' ', '\t', ',' };
    array<String ^> ^ ary_str;
    ary_str = str->Split( separator );
    int L = ary_str->Length;
    int ckp = -1;
    int tmpPos = 0;
    do {
        if (ary_str[tmpPos]->Length > 0) ckp++;
        tmpPos++;
    } while ((ckp != pos) && (tmpPos < L) );
    if (ckp == pos) vstr = ary_str[tmpPos-1];

    return vstr;
}
```

String クラスの Split メソッドを用いてセパレータとして設定された文字によって文字列 str を部分文字列に分割している。セパレータは、関数 Split(separator) の引数 separator に設定されているもので、リスト 1 では半角空白文字「」、タブ「\t」、半角コンマ「,」が指定されている。(1)の実行により、文字列 str に含まれる pos 番目の部分文字列が関数値として返される。位置は 0 番目から数える。

テキストファイルへの書き出しは StreamWriter クラスの Write メソッド、あるいは WriteLine メソッドで行う。

```
StreamWriter ^ 変数 = gcnew StreamWriter( ファイル名、論理値、エンコーディング );
```

と宣言して、

```
変数->WriteLine( );
```

というように書き出す。論理値が true の場合は、既に同じファイル名のファイルが存在する場合にはその末尾に追記となる。false であれば、上書きである。エンコーディングは StreamReader クラスのときと同じで、日本語ファイルとして書き出すときは

```
GetEncoding("shift-jis")
```

と指定する。

上記の機能を用いたプログラム例をリスト2に示す。

リスト2 テキストファイル入出力の例。

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {  
  
    button1->Enabled = false;  
  
    //  
    // 入力用テキストファイル名と入カストリームの設定  
    //  
    openFileDialog1->Title = "Input File Name ?";  
    if (openFileDialog1->ShowDialog() ==  
        System::Windows::Forms::DialogResult::Cancel) {  
        button1->Enabled = true;  
        return;  
    }  
    String ^nmfin = openFileDialog1->FileName;  
  
    using namespace System::IO; // For StreamReader and StreamWriter.  
  
    StreamReader ^ fin = gcnw StreamReader( nmfin,  
        System::Text::Encoding::GetEncoding("shift-jis") );  
  
    //  
    // 出力用テキストファイル名と出カストリームの設定  
    //  
    saveFileDialog1->Title = "Output File Name ?";  
    if (saveFileDialog1->ShowDialog() ==  
        System::Windows::Forms::DialogResult::Cancel) {  
        fin->Close();  
        button1->Enabled = true;  
        return;  
    }  
    String ^ nmfout = saveFileDialog1->FileName;  
  
    StreamWriter ^ fout = gcnw StreamWriter( nmfout, false,  
        System::Text::Encoding::GetEncoding( "shift-jis" ) );  
    fout->WriteLine("入力ファイル..." + nmfin);  
    fout->WriteLine();  
  
    //  
    // 入力ファイルの先頭部分 ("/"で始まる行まで) の読み飛ばし  
    //  
    do {  
        String ^ str = fin->ReadLine();  
        if (str->Length > 0)  
            if (str->Substring(0,1) == "/" ) break;  
    } while( true );  
  
    //  
    // データ数をnにカウント。  
}
```

```

//
int n = 0;
do {
    String ^ s = fin->ReadLine();
    if (s->Substring(0,1) == "/" break;
    n++;
} while (true);
fin->Close();

double * x, * y;
x = new double[n];
y = new double[n];
//
//
//          データの読み込み。
//          入力ファイルを先頭から読み込み直す。
//
fin = gcnew StreamReader( nmfin,
                        System::Text::Encoding::GetEncoding("shift-jis") );
do{
    String ^ str = fin->ReadLine();
    fout->WriteLine(str);
    if (str->Length > 0)
        if (str->Substring(0,1) == "/" break;
}while( true );
for (int i = 0; i < n; i++){
    String ^ s = fin->ReadLine();
    x[i] = double::Parse(SeekStr(s, 0));
    y[i] = double::Parse(SeekStr(s, 1));
}

fout->WriteLine();
fout->WriteLine("入力データ...");
for (int i = 0; i < n; i++)
    fout->WriteLine("x[" + (i+1).ToString() + "] = " +
                    x[i].ToString() +
                    " y[" + (i+1).ToString() + "] = " +
                    y[i].ToString() );

//
//
//          計 算
//
double sumx = 0.0, ssumx = 0.0, sumy = 0.0, ssumy = 0.0, sumxy = 0.0;
for (int i = 0; i < n; i++){
    sumx += x[i];
    ssumx += sqr(x[i]);
    sumy += y[i];
    ssumy += sqr(y[i]);
    sumxy += x[i] * y[i];
}
double meanx = sumx / n;
double meany = sumy / n;
double varx = (ssumx - sqr(sumx) / n) / n;
double sdx = Math::Sqrt( varx );
double uvarx = varx * n / (n - 1.0);

```

```

double sdux = Math::Sqrt( uvarx );
double vary = (ssumy - sqr(sumy) / n) / n;
double sdy = Math::Sqrt( vary );
double uvary = vary * n / (n - 1.0);
double sduy = Math::Sqrt( uvary );
double cov = (sumxy - sumx * sumy / n) / n;
double r = cov / (sdx * sdy );
fout->WriteLine("平均値(X) = " + meanx.ToString("F5"));
fout->WriteLine("分散(X) = " + varx.ToString("F5"));
fout->WriteLine("標準偏差(X) = " + sdx.ToString("F5"));
fout->WriteLine("不偏分散(X) = " + uvarx.ToString("F5"));
fout->WriteLine("不偏分散の平方根(X) = " + sdux.ToString("F5"));
fout->WriteLine("平均値(Y) = " + meany.ToString("F5"));
fout->WriteLine("分散(Y) = " + vary.ToString("F5"));
fout->WriteLine("標準偏差(Y) = " + sdy.ToString("F5"));
fout->WriteLine("不偏分散(Y) = " + uvary.ToString("F5"));
fout->WriteLine("不偏分散の平方根(Y) = " + sduy.ToString("F5"));
fout->WriteLine("共分散 = " + cov.ToString("F5"));
fout->WriteLine("相関係数 = " + r.ToString("F5"));

fout->Close();
MessageBox::Show(nmfout + " に出力されました。");
Close();
}

```

上のリストを含むプロジェクト全体のファイルは圧縮ファイル sample04FileInOut.ZIP とし
てまとめ、ホームページに用意した。この圧縮ファイルにはデータファイルのサンプル
SampleData.txt も含まれている。このファイルはリスト3のような内容である。

リスト3. サンプルデータファイル Sampledata.txt。

```

サンプル・データ
/
1 1
1 2
2 2
2 3
3 3
3 4
4 5
5 5
5 6
/

```

2変量のデータが1行に1対ずつ書かれているが、これらのデータはスラッシュ/で始まる行で
挟まれている。このデータを入力データとしてリスト2のプログラムを実行するとリスト4の結
果をえる。

リスト4. リスト3のデータに対する実行結果。

入力ファイル... H:\C++\Lecture2009\sample04\fileInOut\sample04\fileInOut\SampleData.txt

サンプル・データ

/

入力データ...

x[1] = 1 y[1] = 1

x[2] = 1 y[2] = 2

x[3] = 2 y[3] = 2

x[4] = 2 y[4] = 3

x[5] = 3 y[5] = 3

x[6] = 3 y[6] = 4

x[7] = 4 y[7] = 5

x[8] = 5 y[8] = 5

x[9] = 5 y[9] = 6

平均値(X) = 2.88889

分散(X) = 2.09877

標準偏差(X) = 1.44871

不偏分散(X) = 2.36111

不偏分散の平方根(X) = 1.53659

平均値(Y) = 3.44444

分散(Y) = 2.46914

標準偏差(Y) = 1.57135

不偏分散(Y) = 2.77778

不偏分散の平方根(Y) = 1.66667

共分散 = 2.16049

相関係数 = 0.94907