StringList を使ったランキング機能

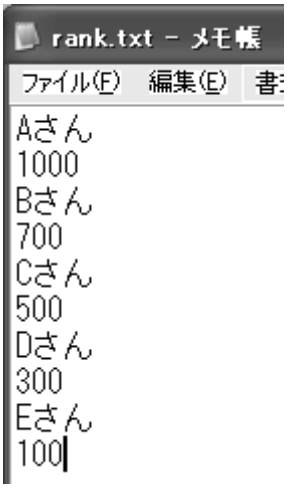
ゲームをプレイするに当って、以前よりもうまくなったとわかるとさらにゲームが楽しくなるものです。そういった自分の上達具合を知る物差しとして、ハイスコアや順位を表示すランキング機能が挙げられます。このランキング機能を、文字を扱うことのできる「StringList」を使って作ってみましょう。

今回は、上位5位のランキングを表示したり記録したりすることができるプログラムを作ってみます。

① 下準備　（初期の用意）

下準備として「名前」「得点」の順番で初期データを作ります。メモ帳などを使いましょう。

初期データを作る際は、読み込む範囲よりも行数が少ないと、読み込み出来ずにエラーが起こるので注意してください



② StringList の宣言

次に、ヘッダーのユーザー宣言のところに StringList を使うことの宣言を書きます。下記のようにしてください。

TStringList *変数名;

変数名は自分で決めます。今回は「Rank」という変数を作りました。
このとき変数名の前に必ず「*」をつけてください。

③　ランキングの読み込み

StringList に使用の宣言をしたら、ソースファイル（cpp）の fastcall に、次のように宣言します。

```
<ヘッダー>
private:        //ユーザー宣言
        TStringList *Rank;
//---------------------------------------------------------------------
<cpp>
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
        //メモリの確保
        Rank = new TStringList;
        //データ読み込み
        Rank->LoadFromFile( "読み込むファイルの名前" );
}
//---------------------------------------------------------------------
```

これにより、変数 Rank に初期データ「rank.txt」が読み込まれました。

④　メモリの消去

コンポーネントなどを使っている際には気になりませんが、今回のようにメモリを自分で確保した場合、自分で消す処理をしないとプログラムを終了してもコンピュータに確保した領域が残ったままになります。

この確保した領域の消去を忘れると、コンピュータが重くなり、挙句の果てにフリーズしてしまってプログラムが駄目になる、なんてことが大いにありえます。ですから必ずメモリを消去するようにしましょう。なお、このメモリを消去する処理を「デストラクタ」といいます。

デストラクタは関数の宣言のような手順で作ります。
初めに、ヘッダーファイルのユーザー宣言と cpp に下記のように書き加えてください。
デストラクタの領域は fastcall の次くらいに作っておけば良いでしょう。
メモリを削除する命令は delete を使います。

<ヘッダー>

```
public:          //ユーザー宣言
        __fastcall TForm1::~TForm1();
//---------------------------------------------------------------
```

<cpp>

```
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
                          ～中略～
}
__fastcall TForm1::~TForm1()
{
    delete Rank;         //メモリの消去
}
//---------------------------------------------------------------
```

Delete で作った変数「Rank」を消去します。これでプログラムを終了するたびに確保したメモリを消去することができます。

⑤　ランキングの情報を取得

上位５名文のランキングを表示するには、名前記録用の文字型変数と得点記録用の数値型変数がそれぞれ５つ必要になります。加えて、上位の更新もできるようにするなめ、作業用の変数も必要になります。よってそれぞれ６つの文字、数値型変数が要ります。

変数を宣言したら、読み込んだテキスト内容を変数に代入してみましょう。

このときデータは言った変数の方はすべて AnsiString 型のものとして扱われます。「名前」などはそのままで良いのですが「得点」などの数値も文字として扱われているため計算できずに困るので、文字から数値へ型変換しておく必要があります。

これには

StrToIntDef(引数、数値)

を使います。() の引数に入るのは、文字変数型の整数値です。
数値は、引数に入っていたものが数字でない場合、その数値を返す、ということを意味しています。この部分はそれほど気に病まずに、「0」と書いておいてください。

以上を踏まえると次のようになります。

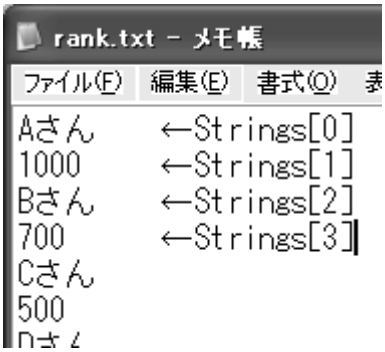
```
//ヘッダーでの宣言内容
        int Score[6];
        AnsiString Name[6];

//cpp に書く内容

    for(int i=0;i<5;i++){
        Name[i] = Rank->Strings[i*2];
        Score[i] = StrToIntDef(Rank->Strings[i*2+1],0);
    }
```

Name[]には「A さん」「B さん」・・・、
Score[]には「1000」「700」・・・、が代入されます。

注意点として、String[]は右の図のように、元のファイルの１行目が String[0] 、2 行目が String[1]、・・・となります。行の場所と String[ ]の配列を取り違えないように気をつけてください。



⑥ ランキングの更新

ゲームにおいて高得点をとった場合、当然その度記録を更新しなければなりません。このランキングの更新を再現してみましょう。

ランキングを更新する方法は色々ありますが、今回は StringList の書き換えを行ってみましょう。

初めに得点も大小に関係なく、作業用の変数、今回だと Score[5]に得点、Name[5]にプレイした人の名前を代入します。名前は Edit などに書き込んだものを代入するようにしましょう。この作業用の変数とランキングのスコアを低い順に比較し、順位を入れ替えていく・・・という方法です。

⑦ 更新内容の保存

そして、ランキングの更新をした後はその更新した結果を読み込み元（今回の場合は「rank.txt」）を更新しなくてはなりません。（今のままでは「rank.txt」は影響を受けていない）そこで、更新されたデータをメモ帳に上書きして保存しましょう。これでランキングを保存することができます。

変数名->SaveToFile(“上書きしたいファイル名”);

と書くことで上書きして保存できます。

```
//ヘッダーでの宣言内容
int Your_Score,temp_score;
AnsiString temp_name;

//cppに書く内容

Score[5] = Your_Score;          //最新のスコアを代入
Name[5] = Edit1->Text;          //Editから文字を代入
//スコアを比べて順位を入れ替え------------------------------
for(int i=5;i>=1;i--){
     if(Score[i] > Score[i-1]){
         temp_score = Score[i];
         Score[i] = Score[i-1];
         Score[i-1] = temp_score;

          temp_name = Name[i];
          Name[i] = Name[i-1];
          Name[i-1] = temp_name;
     }
}
//更新内容--------------------------------------------------
for(int i=0;i<5;i++){
        Rank->Strings[i*2] = Name[i];
        Rank->Strings[i*2+1] = Score[i];
    }

    Rank->SaveToFile("rank.txt");
//----------------------------------------------------------
```

以上でランキングの解説は終わります。ランキング機能は StringList 以外のやり方でもできますので、是非考えてみてください。