#include <vcl.h>

#include <stdio.h>

#include <stdlib.h> #include <string.h>

#pragma hdrstop

#include "Unit1.h"

//----------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

struct Day

{

float TempMorning;

float TempDay;

float TempEvening;

float TempNight;

float VlagMorning;

float VlagDay;

float VlagEvening;

float VlagNight;

float BarMorning;

float BarDay;

float BarEvening;

float BarNight;

char Time [50];

};

struct Day\* DataOfWeather = NULL;

int DataOfWeatherCount;

int ReadDataFromFile (AnsiString FileName)

{

FILE\* File; данных из него

char Str [500];

int i;

char\* StrPos;

char\* StrStart;

File = fopen (FileName.c\_str(), "r");

if (File == NULL) return 0;

DataOfWeatherCount = 0;

while (feof (File) == 0)

{

if (fgets (Str, sizeof (Str), File) != NULL)

{

if (strlen (Str) > 0 && Str [0] != ';')DataOfWeatherCount++;

}

}

DataOfWeather = (struct Day\*) malloc (DataOfWeatherCount \* sizeof

Day));

if (DataOfWeather == NULL)

{

fclose (File);

return 0;

}

memset (DataOfWeather, 0, DataOfWeatherCount \* sizeof (Day));

rewind (File);

i = 0;

while (feof (File) == 0)

{

if (fgets (Str, sizeof (Str), File) != NULL)

{

if (strlen (Str) > 0 && Str [0] != ';')

{

StrStart = Str;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].TempMorning = StrToFl (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].TempDay = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].TempEvening=StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].TempNight = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].TempNight = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].VlagDay = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].VlagDay = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].VlagNight = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].VlagDay = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].BarDay = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].VlagDay = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

StrPos = strchr (StrStart, ',');

if (StrPos != NULL)

{

\*StrPos = 0;

DataOfWeather [i].BarNight = StrToFloat (StrStart);

}

StrStart = StrPos + 1;

strncpy (DataOfWeather [i].Time, StrStart, sizeof (DataOfWeather [i].Time) - 1);

i++;

}

}

}

fclose (File);

return 1;

}

int AddDaysInListBox (TListBox\* ListBox)

{

ListBox->Clear();

if (DataOfWeather == NULL) return 0;

if (DataOfWeatherCount == 0) return 0;

for (int i = 0; i < DataOfWeatherCount; i++)

{

ListBox->AddItem (DataOfWeather [i].Time, NULL);

}

return 1;

}

void \_\_fastcall TForm1::ListBox1Click(TObject \*Sender)

{

TListBox\* ListBox = (TListBox\*) Sender;

int SelectedItem = ListBox->ItemIndex;

Label6->Caption = FloatToStr (DataOfWeather SelectedItem].TempMorning);

Label7->Caption = FloatToStr (DataOfWeather [SelectedItem].TempDay);

Label8->Caption = FloatToStr (DataOfWeather SelectedItem].TempEvening);

Label9->Caption = FloatToStr (DataOfWeather [SelectedItem].TempNight);

Label14->Caption = FloatToStr(DataOfWeather[SelectedItem].VlagMorning);

Label15->Caption = FloatToStr (DataOfWeather [SelectedItem].VlagDay);

Label16->Caption = FloatToStr (DataOfWeather[SelectedItem].VlagEvening);

Label17->Caption = FloatToStr (DataOfWeather [SelectedItem].VlagNight);

Label22->Caption = FloatToStr (DataOfWeather [SelectedItem].BarMorning);

Label23->Caption = FloatToStr (DataOfWeather [SelectedItem].BarDay);

Label24->Caption = FloatToStr (DataOfWeather [SelectedItem].BarEvening);

Label25->Caption = FloatToStr (DataOfWeather [SelectedItem].BarNight);

return;

}

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//----------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

AnsiString FileName;

OpenDialog1->Execute();

FileName = OpenDialog1->FileName;

if (ReadDataFromFile (FileName) == 0)

MessageBox (NULL, "Ошибка чтения из файла!", "Ошибка", MB\_OK | B\_ICONERROR);

return;

}

if (AddDaysInListBox (ListBox1) == 0)

{

MessageBox (NULL, "Ошибка добавления данных в istBox!","Ошибка",

MB\_OK | MB\_ICONERROR);

}

}

//----------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

if (ListBox1->Count == 0)

{

MessageBox (NULL, "Данные не загружены!", "Информация", MB\_OK);

return;

}

if (ListBox1->ItemIndex == -1)

{

MessageBox (NULL, "Дата не выбрана!", "Информация", MB\_OK);

return;

}

int SelectedItem = ListBox1->ItemIndex;

float MiddleTemp = (DataOfWeather [SelectedItem].TempMorning + DataOfWeather [SelectedItem].TempDay + DataOfWeather [SelectedItem].TempEvening + DataOfWeather [SelectedItem].TempNight) / 4;

float MinTemp = 273;

float MaxTemp = -273;

if (DataOfWeather [SelectedItem].TempMorning > MaxTemp) MaxTemp = DataOfWeather [SelectedItem].TempMorning;

if (DataOfWeather [SelectedItem].TempDay > MaxTemp) MaxTemp = DataOfWeather [SelectedItem].TempDay;

if (DataOfWeather [SelectedItem].TempEvening > MaxTemp) MaxTemp = DataOfWeather [SelectedItem].TempEvening;

if (DataOfWeather [SelectedItem].TempNight > MaxTemp) MaxTemp = DataOfWeather [SelectedItem].TempNight;

if (DataOfWeather [SelectedItem].TempMorning < MinTemp) MinTemp = DataOfWeather [SelectedItem].TempMorning;

if (DataOfWeather [SelectedItem].TempDay < MinTemp) MinTemp = DataOfWeather [SelectedItem].TempDay;

if (DataOfWeather [SelectedItem].TempEvening < MinTemp) MinTemp = DataOfWeather [SelectedItem].TempEvening;

if (DataOfWeather [SelectedItem].TempNight < MinTemp) MinTemp = DataOfWeather [SelectedItem].TempNight;

Label29->Caption = FloatToStr (MiddleTemp);

Label30->Caption = FloatToStr (MinTemp);

Label31->Caption = FloatToStr (MaxTemp);

return; // Выход из функции

}

//----------------------------------------------------------------------