**MyForm.cpp**

#include "MyForm.h",

#include <Windows.h>

using namespace Project1;

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Application::Run(gcnew MyForm);

return 0;

}

**MyForm.h**

#include <iostream>,

#include <fstream>

#include <string>

#include <ctime>

#include <thread>

#include <Windows.h>

#include <locale.h>

#include "Vector.cpp"

#include "../../Dll1/funct.h"

#pragma once

using namespace std;

#define THREADCOUNT 1

HANDLE myMutex;

vector<string> parts;

string end\_data;

static void WriteToCSV(vector<string> arr)

{

ofstream out; // ïîòîê äëÿ çàïèñè

out.open("2.csv");

end\_data = "";// îêðûâàåì ôàéë äëÿ çàïèñè

if (out.is\_open())

{

for (int i = 0; i < arr.size(); i++)

{

string str = arr[i];

for (int j = 0; j < str.size(); j++)

{

end\_data += str[j];

end\_data += ',';

out << str[j] << ',';

}

out << endl;

end\_data += "\n";

}

}

out.close();

}

static string ReadFromFile(string path)

{

string line;

ifstream in(path); // îêðûâàåì ôàéë äëÿ ÷òåíèÿ

if (in.is\_open())

{

getline(in, line);

}

in.close(); // çàêðûâàåì ôàéë

return line;

}

static vector<string> DivideLine(string line)

{

vector<string> parts;

for (int i = 0; i < line.length(); i += 10) {

parts.push\_back(line.substr(i, 10));

}

return parts;

}

DWORD WINAPI WriteToFile(LPVOID lpParam)

{

DWORD dwCount = 0, dwWaitResult; // Çàïðîñèòü âëàäåíèå ìüþòåêñîì

while (dwCount < 2)

{

dwWaitResult = WaitForSingleObject(

myMutex, // handle íà ìüþòåêñ

INFINITE); // íåò òàéìàóòà

switch (dwWaitResult)

{

case WAIT\_OBJECT\_0: // Ïîòîê çàâëàäåë ìüþòåêñîì

\_\_try {

WriteToCSV(parts),

GetCurrentThreadId();

dwCount++;

}

\_\_finally {

if (!ReleaseMutex(myMutex))// Îñâîáîäèòü ìüþòåêñ

{

//error

}

}

break;

case WAIT\_ABANDONED: // Åñëè îøèáêà

return FALSE;

}

}

return TRUE;

};

namespace Project1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Summary for MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: Add the constructor code here

//

}

protected:

/// <summary>

/// Clean up any resources being used.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::Label^ label7;

protected:

private:

/// <summary>

/// Required designer variable.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

void InitializeComponent(void)

{

this->button1 = (gcnew System::Windows::Forms::Button());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->label5 = (gcnew System::Windows::Forms::Label());

this->label6 = (gcnew System::Windows::Forms::Label());

this->label7 = (gcnew System::Windows::Forms::Label());

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(371, 432);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(188, 52);

this->button1->TabIndex = 2;

this->button1->Text = L"Íà÷àòü âûïîëíåíèå ïðîãðàììû";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->Location = System::Drawing::Point(13, 13);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(395, 20);

this->label1->TabIndex = 0;

this->label1->Text = L"Êóçíåöîâà Åêàòåðèíà Åâãåíüåâíà, ãðóïïà 503";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label2->Location = System::Drawing::Point(224, 46);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(24, 20);

this->label2->TabIndex = 3;

this->label2->Text = L" ";

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label3->Location = System::Drawing::Point(13, 46);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(204, 20);

this->label3->TabIndex = 4;

this->label3->Text = L"Òåêóùàÿ äàòà è âðåìÿ:";

//

// label4

//

this->label4->AutoSize = true;

this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label4->Location = System::Drawing::Point(13, 75);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(165, 20);

this->label4->TabIndex = 5;

this->label4->Text = L"Èñõîäíûå äàííûå:";

//

// label5

//

this->label5->AutoSize = true;

this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label5->Location = System::Drawing::Point(13, 106);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(106, 20);

this->label5->TabIndex = 7;

this->label5->Text = L"íåò äàííûõ";

//

// label6

//

this->label6->AutoSize = true;

this->label6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label6->Location = System::Drawing::Point(13, 137);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(241, 20);

this->label6->TabIndex = 8;

this->label6->Text = L"Äàííûå â êîíå÷íîì ôàéëå:";

//

// label7

//

this->label7->AutoSize = true;

this->label7->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10.2F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label7->Location = System::Drawing::Point(13, 168);

this->label7->Name = L"label7";

this->label7->Size = System::Drawing::Size(106, 20);

this->label7->TabIndex = 9;

this->label7->Text = L"íåò äàííûõ";

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(949, 496);

this->Controls->Add(this->label7);

this->Controls->Add(this->label6);

this->Controls->Add(this->label5);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->button1);

this->Controls->Add(this->label1);

this->Name = L"MyForm";

this->Text = L"MyForm";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) //êíîïêà íà÷àòü âûïîëíåíèå

{

string str = MyFunc::GetTime();

String^ strg = gcnew String(str.c\_str());

label2->Text = strg;

string line = ReadFromFile("1.txt");

String^ text = gcnew String(line.c\_str());

label5->Text = text;

parts = DivideLine(line);

HANDLE aThread;

DWORD ThreadID;

myMutex = CreateMutex( // Ñîçäàíèå ìüþòåêñà

NULL, // àòðèáóòû áåçîïàñíîñòè ïî óìîë÷àíèþ

FALSE, // áåç âëàäåëüöà

NULL); // áåç èìåíè

if (myMutex == NULL)

{

MessageBox::Show(L"Îøèáêà ñîçäàíèÿ ìüþòåêñà", L"Îøèáêà!");

}

aThread = CreateThread(

NULL, // àòðèáóòû áåçîïàñíîñòè ïî óìîë÷àíèþ

0, // ðàçìåð ñòåêà ïî óìîë÷àíèþ

(LPTHREAD\_START\_ROUTINE)WriteToFile,

NULL, // íåò àðãóìåíòîâ ïðîöåññà

0, // default creation flags

&ThreadID); // ïîëó÷èòü id ïîòîêà

if (aThread == NULL)

{

MessageBox::Show(L"Îøèáêà ñîçäàíèÿ ïîòîêà", L"Îøèáêà!");

}

// Çàêðûòü ïîòîê è ìüþòåêñ

WaitForMultipleObjects(THREADCOUNT, &aThread, TRUE, INFINITE);

CloseHandle(aThread);

CloseHandle(myMutex);

system("attrib +R 2.csv"); // äîáàâëÿåì àòðèáóò òîëüêî äëÿ ÷òåíèÿ

strg = gcnew String(end\_data.c\_str());

label7->Text = strg;

button1->Enabled = FALSE;

}

};

}

**Vector.cpp**

template <typename T> class vector,

{

private:

T\* \_array; //óêàçàòåëü íà ìàññèâ

int \_arraySize; //ðàçìåð ìàññèâà

public:

void push\_back(T data)

{

T\* \_result = new T[++this->\_arraySize]; //ñîçäàåì íîâûé ìàññèâ ðàçìåðîì èñõîäíîãî+1

for (int index = 0; index < this->\_arraySize; index++)

{

if (index != this->\_arraySize - 1) //åñëè íå ïîñëåäíèé ýëåìåíò ìàññèâà

{

\_result[index] = this->\_array[index]; //ïåðåïèñûâàåì çíàåíèå â íîâûé ìàññèâ

}

else

{

\_result[index] = data; //çàïèñûâàåì äàííûå â ïîñëåäíèé ýëåìåíò

break;

}

}

delete[] this->\_array; //óäàëÿåì èñõîäíûé ìàññèâ

this->\_array = \_result; //ïðèñâàèâàåì ññûëêó íà íîâûé ìàññèâ

}

void clear()

{

if (this->\_array != nullptr)

{

::ZeroMemory(this->\_array, this->\_arraySize); //çàïîëíÿåò íóëÿìè áëîê ïàìÿòè

}

this->\_arraySize = 0;

this->\_array = new T[this->\_arraySize];

}

T operator [](int index) //ïîëó÷èòü çíà÷åíèå ïî èíäåêñó

{

return this->\_array[index];

}

int size() //ïîëó÷èòü ðàçìåð âåêòîðà

{

return this->\_arraySize;

}

vector() //êîíñòðóêòîð ïî óìîë÷

{

this->\_arraySize = 0;

this->\_array = new T[this->\_arraySize];

}

};