using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Security.Cryptography;

namespace Тест

{

public partial class Form1 : Form

{

int count\_question = 0, ci = -1, time\_s = 0, pass\_mark;

bool check; // проверка для прохождения теста

string[] AS; // массив строк

string full\_name, hash = "5VID@A67", test\_name;

bool[] true\_question; // массив верно и неверно отвеченных вопросов

List<int> true\_answers = new List<int>(); // список правильных ответов

List<int> list = new List<int>(); // список выбранных ответов

private static string key = "<RSAKeyValue><Modulus>4z7qydSNly92pdMuidf6YCjHbkr6iqB6GD5JlJ4i0wwX0zIZ+14Mfmb3P67Y5FmeaE9pIgdg0/cgkMiicgz7vbbhWevAJTh18Hg2zm7blLJgqggNXugCM+vHcL00uEFVT4AZC1TeB4Pwn2m4/YI1H/T7M3CcGxYu469yvE9LpNE=</Modulus><Exponent>AQAB</Exponent><P>7gmkYgqp2dT6qCZn16ch9HmV4kCwuEtIUpkgaYlE1gkrUDppeI/JLXqBSnGwrdKGplnLdxjNxhFDl+DeZu+6aw==</P><Q>9GTMtTPs4qrk5ynoK+y1e/+TnQL+s//RDGeVQ0Do3e+Q7LixXD9QEgSM8Kcqs2dl1y3cUObg1mplmWF3Ti/ksw==</Q><DP>YZZqYhdlEv0R1iYIpIzMiuHd5OgIi0Wx3MsSUrIIGoefzkBkRR0NqSt0HgUGHf0VXtWpaKYulmk1gOqbwy+9CQ==</DP><DQ>UO+w+7GemdKOzB+MzD8wO0iA0lbTkzj9OmZzgDpW5GkaTcs8sZgmJFFte6Vg9tkfWQvPEtvjSh8iVpF4uWQTEQ==</DQ><InverseQ>VhAxaWGU+VU+r4EYtsfBMu9ZaEiO3GPfHz9L3PFM6YsGBlG8OXNz+e+JC2B36q5BOuttazDh9x8++9muZUIG3A==</InverseQ><D>Ed8G0+yVXghrikr7Y/fKRvmuBWx8G57XO0TawIdXMcXHseJEzNHupZovY6C6ierKXGWyXOy7nq/Wrcw5ikKfIXxptQnRToVNKrNGPTbuo2GldSRMyDIwijP8BQYbet33I9WqDCnmArL7Q3GpQkXZu4hvW9JcyuMfR0BZShhh0ek=</D></RSAKeyValue>";

public Form1()

{

InitializeComponent();

string path = "C:\\Test.txt"; // путь для считывание вопросов и ответов с текстого файла

//считывание файла построчно, подсчет количества вопросов в файле

using (StreamReader reader = new StreamReader(path, System.Text.Encoding.GetEncoding(1251)))

{

string line;

while ((line = reader.ReadLine()) != null) //подсчет количества вопросов

if (line != "" && line[0] == '-') count\_question++;

}

//разделение вопросов между собой в тексте

using (StreamReader reader = new StreamReader(path, System.Text.Encoding.GetEncoding(1251)))

{

string line;

if ((line = reader.ReadLine()) != null)

{

label1.Text = test\_name = line; //считывание первой строки в файле, запись данных как название теста в окно вывода текста

}

if ((line = reader.ReadLine()) != null) //считывание второй строки в файле, сохранение процента прохождения теста

{

label1.Text = label1.Text + "\n" + line;

line = line.Substring(16);

line = line.Substring(0, line.Length - 1);

pass\_mark = int.Parse(line);

}

// Выделение памяти для массива из count строк и расчет времени в зависимости количества вопросов

AS = new string[count\_question];

time\_s = (count\_question \* 2 / 3) \* 60;

//распределение вопросов из файла рандомно

for (int i = 0; i < count\_question; i++)

{

Again: var rand = new Random();

int j = rand.Next(count\_question);

if (AS[j] == null)

while ((line = reader.ReadLine()) != null)

{

if (line != "")

AS[j] += line;

else break;

}

else goto Again;

}

count\_question = count\_question \* 2 / 3;

true\_question = new bool[count\_question];

}

}

private void button1\_Click(object sender, EventArgs e)

{

ci--;

if (ci == 0) button1.Enabled = false;

if (ci > -1 && ci < count\_question)

{

//проверка выбранных ответов

list.Clear();

foreach (int j in checkedListBox1.CheckedIndices)

list.Add(j);

check = true;

if (list.Count() == true\_answers.Count())

{

for (int j = 0; j < list.Count(); j++)

if (list[j] != true\_answers[j])

check = false;

}

else check = false;

true\_question[ci] = check;

//работа с вопросом

string question;

question = AS[ci];

int ind = question.IndexOf('/');

question = question.Substring(1, ind - 1);

label1.Text = question;

//работа с ответами вопроса

string answer;

answer = AS[ci];

answer = answer.Remove(0, ind + 1);

checkedListBox1.Items.Clear();

string[] answers = answer.Split(new char[] { '/' });

true\_answers.Clear();

int i = 0;

foreach (string s in answers)

{

if (s[0] == '\*') true\_answers.Add(i);

checkedListBox1.Items.Add(s.Trim(new char[] { '\*', ' ' }));//.Trim(new char[] { '\*', ' ' })

i++;

}

button2.Text = "Следующий вопрос";

}

}

private void button2\_Click(object sender, EventArgs e)

{

if(textBox1.Text == "" || textBox1.Text == "Введите ФИО полностью") MessageBox.Show("ФИО не введено! Введите данные");

else

{

ci++;

if (ci == 0)

{

//включение временного промежутка и видимости элемента для ответов

timer1.Enabled = true;

checkedListBox1.Visible = true;

button1.Text = "Предыдущий вопрос";

full\_name = textBox1.Text;

textBox1.Visible = false;

//работа с вопросом

button2.Text = "Следующий вопрос";

string question;

question = AS[ci];

int ind = question.IndexOf('/');

question = question.Substring(1, ind - 1);

label1.Text = question;

//работа с ответами вопроса

string answer;

answer = AS[ci];

answer = answer.Remove(0, ind + 1);

string[] answers = answer.Split(new char[] { '/' });

int i = 0;

foreach (string s in answers)

{

if (s[0] == '\*') true\_answers.Add(i);

checkedListBox1.Items.Add(s.Trim(new char[] { '\*', ' ' }));//.Trim(new char[] { '\*', ' ' })

i++;

}

}

if (ci > 0 && ci < count\_question)

{

if (ci == 1) button1.Enabled = true;

//проверка выбранных ответов

list.Clear();

foreach (int j in checkedListBox1.CheckedIndices)

list.Add(j);

check = true;

if (list.Count() == true\_answers.Count())

{

for (int j = 0; j < list.Count(); j++)

if (list[j] != true\_answers[j])

check = false;

}

else check = false;

true\_question[ci - 1] = check;

//работа с вопросом

string question;

question = AS[ci];

int ind = question.IndexOf('/');

question = question.Substring(1, ind - 1);

label1.Text = question;

//работа с ответами вопроса

string answer;

answer = AS[ci];

answer = answer.Remove(0, ind + 1);

checkedListBox1.Items.Clear();

string[] answers = answer.Split(new char[] { '/' });

true\_answers.Clear();

int i = 0;

foreach (string s in answers)

{

if (s[0] == '\*') true\_answers.Add(i);

checkedListBox1.Items.Add(s.Trim(new char[] { '\*', ' ' }));//.Trim(new char[] { '\*', ' ' })

i++;

}

//проверка, на последний ли вопрос выбирают ответ

if (ci == count\_question - 1) button2.Text = "Завершить тест";

}

if (ci == count\_question)

{

//проверка выбранных ответов

list.Clear();

foreach (int j in checkedListBox1.CheckedIndices)

list.Add(j);

check = true;

if (list.Count() == true\_answers.Count())

{

for (int j = 0; j < list.Count(); j++)

if (list[j] != true\_answers[j])

check = false;

}

else check = false;

true\_question[ci - 1] = check;

//завершение теста

label1.Text = "";

time\_s = count\_question \* 60 - time\_s;

if (time\_s % 60 < 10)

label2.Text = "Тест завершен за: " + (time\_s / 60).ToString() + ":0" + (time\_s % 60).ToString();

else

label2.Text = "Тест завершен за: " + (time\_s / 60).ToString() + ":" + (time\_s % 60).ToString();

button2.Enabled = false;

button2.Text = "";

button1.Enabled = false;

button1.Text = "";

checkedListBox1.Items.Clear();

checkedListBox1.Visible = false;

int count = 0;

for (int j = 0; j < true\_question.Count(); j++)

if (true\_question[j] == true) count++;

if (count \* 100 / count\_question > pass\_mark)

{

label1.ForeColor = Color.Green;

label1.Text = "Пройден!\n" + count + " из " + count\_question;

timer1.Stop();

}

else

{

label1.ForeColor = Color.Red;

label1.Text = "Провален!\n" + count + " из " + count\_question;

timer1.Stop();

}

DateTime now = DateTime.Now;

//шифрование данных

byte[] encContent;

RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();

rsa.FromXmlString(key);

using(StreamWriter encryptWriter = new StreamWriter("C:\\result.txt", true))

{

encContent = rsa.Encrypt(Encoding.UTF8.GetBytes(full\_name), true);

encryptWriter.WriteLineAsync(Convert.ToBase64String(encContent));

encContent = rsa.Encrypt(Encoding.UTF8.GetBytes(test\_name), true);

encryptWriter.WriteLineAsync(Convert.ToBase64String(encContent));

encContent = rsa.Encrypt(Encoding.UTF8.GetBytes(now.ToString("d")), true);

encryptWriter.WriteLineAsync(Convert.ToBase64String(encContent));

encContent = rsa.Encrypt(Encoding.UTF8.GetBytes((count \* 100 / count\_question).ToString()), true);

encryptWriter.WriteLineAsync(Convert.ToBase64String(encContent));

encContent = rsa.Encrypt(Encoding.UTF8.GetBytes((time\_s / 60).ToString() + ":" + (time\_s % 60).ToString()), true);

encryptWriter.WriteLineAsync(Convert.ToBase64String(encContent));

}

}

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

// пуск отсчета времени

time\_s--;

if(time\_s % 60 < 10)

label2.Text = "До окончания: " + (time\_s / 60).ToString() + ":0" + (time\_s % 60).ToString();

else

label2.Text = "До окончания: " + (time\_s / 60).ToString() + ":" + (time\_s % 60).ToString();

label2.Visible = true;

// проверка, если вышло время, тест останавливается

if(time\_s == 0)

{

label1.Text = "";

button2.Enabled = false;

button2.Text = "";

checkedListBox1.Items.Clear();

label1.ForeColor = Color.Red;

label1.Text = "Провален";

timer1.Stop();

}

}

private void textBox1\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox1.ForeColor = Color.Black;

}

}

}