Министерство образования и науки Российской Федерации

КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

КАФЕДРА РАДИОФИЗИКИ

ОТЧЕТ

Криптография

Задача №1

Студент 4 курса,

группы 06-110

А.Ф.Хаматзянова

Казань – 2015

Формулировка задания:

Написать программу, реализующую шифр замены.

Требования к программе:

* В начале работы программы пользователь выбирает тип операции: шифрование или расшифрование.
* Программа позволяет зашифровывать и расшифровывать текст на русском языке, записанный в текстовом файле в стандартной кодировке ОС Windows.
* Зашифрованный текст сохраняется в отдельном текстовом файле.
* Таблица замены (ключ шифрования) генерируется случайным образом и сохраняется в файле key.txt.
* Расшифрование файла производится на основе таблицы, которая считывается из файла key.txt.
* Расшифрованный текст сохраняется в отдельном файле.

Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace first

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

List<char> alfa = new List<char>();

int i;

List<char> key = new List<char>();

List<int> usedIndex = new List<int>();

List<char> textToEnc = new List<char>();

List<char> textToDec = new List<char>();

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

{

//Исходный текст

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.Filter =

"Txt files |\*.txt";

openFileDialog1.Title = "Текст для шифрования";

DialogResult dr = openFileDialog1.ShowDialog();

if (dr == System.Windows.Forms.DialogResult.OK)

{

using (FileStream reader = File.OpenRead(openFileDialog1.FileName))

{

byte[] array = new byte[reader.Length];

reader.Read(array, 0, array.Length);

// декодируем байты в строку

string textFromFile = System.Text.Encoding.Default.GetString(array);

foreach (char s in textFromFile)

{

textToEnc.Add(s);

}

}

}

// Папка для ключа и зашифрованного текста

string Dir = "";

FolderBrowserDialog folderBrowserDialog1 = new FolderBrowserDialog();

//folderBrowserDialog1.RootFolder = Environment.SpecialFolder.Personal;

folderBrowserDialog1.ShowNewFolderButton = true;

DialogResult dialogresult = folderBrowserDialog1.ShowDialog();

folderBrowserDialog1.Description = "Выбор папки для записи ключа и результата шифрования";

if (dialogresult == DialogResult.OK)

{

Dir = folderBrowserDialog1.SelectedPath;

}

// Алфавит

for ( i = 1040; i < 1104; i++)

{

alfa.Add((char)i);

//добавляем Ё

if (i == 1045)

alfa.Add((char)1025);

//добавляем ё

if (i == 1077)

alfa.Add((char)1105);

}

//добавляем пробел

alfa.Add((char)32);

//генерируем ключ

Random random = new Random();

while (key.Count != 67)

{

int a = random.Next(0,67);

if (!(usedIndex.Contains(a)))

{

key.Add(alfa[a]);

usedIndex.Add(a);

}

}

string tempkey = "";

foreach (char s in alfa)

{

tempkey = tempkey + s;

}

foreach (char s in key)

{

tempkey = tempkey + s;

}

MessageBox.Show(tempkey);

using (StreamWriter writer = new StreamWriter(File.Open(Path.Combine(Dir, "key.txt"), FileMode.Create)))

{

foreach (char s in alfa)

writer.Write(s);

writer.WriteLine();

foreach (char s in key)

writer.Write(s);

writer.Close();

}

//шифрование

List <char> result = new List<char>();

for (i = 0; i != textToEnc.Count; i++)

{

char s = textToEnc[i];

bool flag = false;

for (int j = 0; j !=alfa.Count; j++)

{

if (flag == false)

{

if (alfa.Contains(s))

{

int index = alfa.IndexOf(s);

result.Add(key[index]);

flag = true;

}

else

{

result.Add(s);

flag = true;

}

}

}

}

//запись результата в файл

using (StreamWriter writer = new StreamWriter(File.Open(Path.Combine(Dir, "encodedText.txt"), FileMode.Create)))

{

foreach (char s in result)

writer.Write(s);

writer.Close();

}

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

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

{

//чтение исходного файла

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.Filter =

"Txt files |\*.txt";

openFileDialog1.Title = "Текст для дешифрации";

DialogResult dr = openFileDialog1.ShowDialog();

if (dr == System.Windows.Forms.DialogResult.OK)

{

string tmptext = "";

StreamReader streamReaderText = new StreamReader(openFileDialog1.FileName);

while (!streamReaderText.EndOfStream) //Цикл длиться пока не будет достигнут конец файла

{

tmptext += streamReaderText.ReadLine(); //В переменную str по строчно записываем содержимое файла

}

streamReaderText.Close();

foreach (char s in tmptext)

{

textToDec.Add(s);

}

}

// Папка для ключа и зашифрованного текста

string Dir = "";

FolderBrowserDialog folderBrowserDialog1 = new FolderBrowserDialog();

//folderBrowserDialog1.RootFolder = Environment.SpecialFolder.Personal;

folderBrowserDialog1.ShowNewFolderButton = true;

DialogResult dialogresult = folderBrowserDialog1.ShowDialog();

folderBrowserDialog1.Description = "Выбор папки для записи результата дешифрования";

if (dialogresult == DialogResult.OK)

{

Dir = folderBrowserDialog1.SelectedPath;

}

//чтение ключа

OpenFileDialog openFileDialog2 = new OpenFileDialog();

openFileDialog2.Filter =

"Txt files |\*.txt";

openFileDialog2.Title = "Ключ для дешифрования";

DialogResult dr2 = openFileDialog2.ShowDialog();

List<char> temp = new List<char>();

string tmp = "";

StreamReader streamReader = new StreamReader(openFileDialog2.FileName);

while (!streamReader.EndOfStream) //Цикл длиться пока не будет достигнут конец файла

{

tmp+= streamReader.ReadLine(); //В переменную str по строчно записываем содержимое файла

}

streamReader.Close();

temp = tmp.ToList();

alfa = temp.GetRange(0, 67);

key = temp.GetRange(67, 67);

//декодирование

List<char> result = new List<char>();

for (i = 0; i != textToDec.Count; i++)

{

char s = textToDec[i];

bool flag = false;

for (int j = 0; j != key.Count; j++)

{

if (flag == false)

{

if (key.Contains(s))

{

int index = key.IndexOf(s);

result.Add(alfa[index]);

flag = true;

}

else

{

result.Add(s);

flag = true;

}

}

}

}

//запись результата в файл

using (StreamWriter writer = new StreamWriter(File.Open(Path.Combine(Dir, "decodedText.txt"), FileMode.Create)))

{

foreach (char s in result)

writer.Write(s);

writer.Close();

}

}

}

}

Результат работы программы:

Результатом работы программы являются 3 файла:

* key.txt – ключ;
* encodedText.txt – зашифрованный текст;
* decodedText.txt – расшифрованный текст.

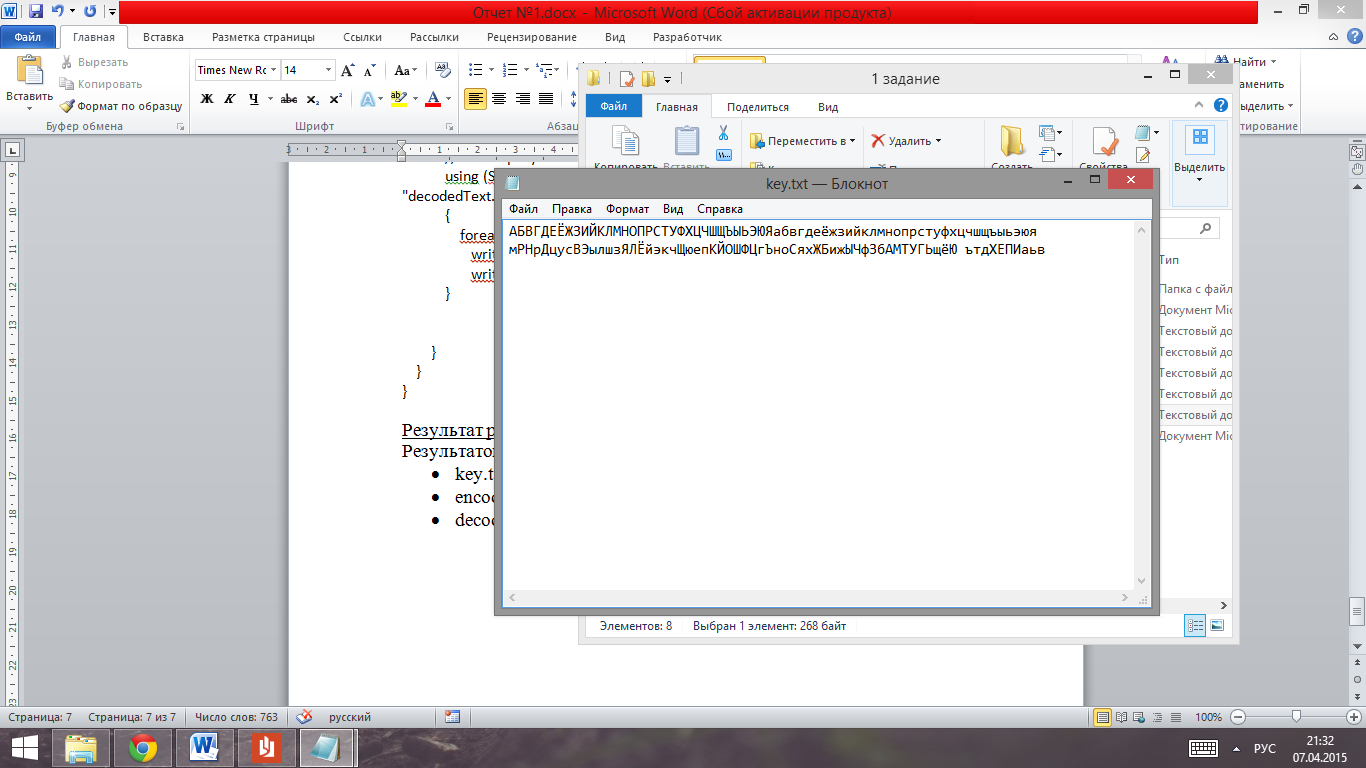


Рисунок 1. Содержимое файла key.txt

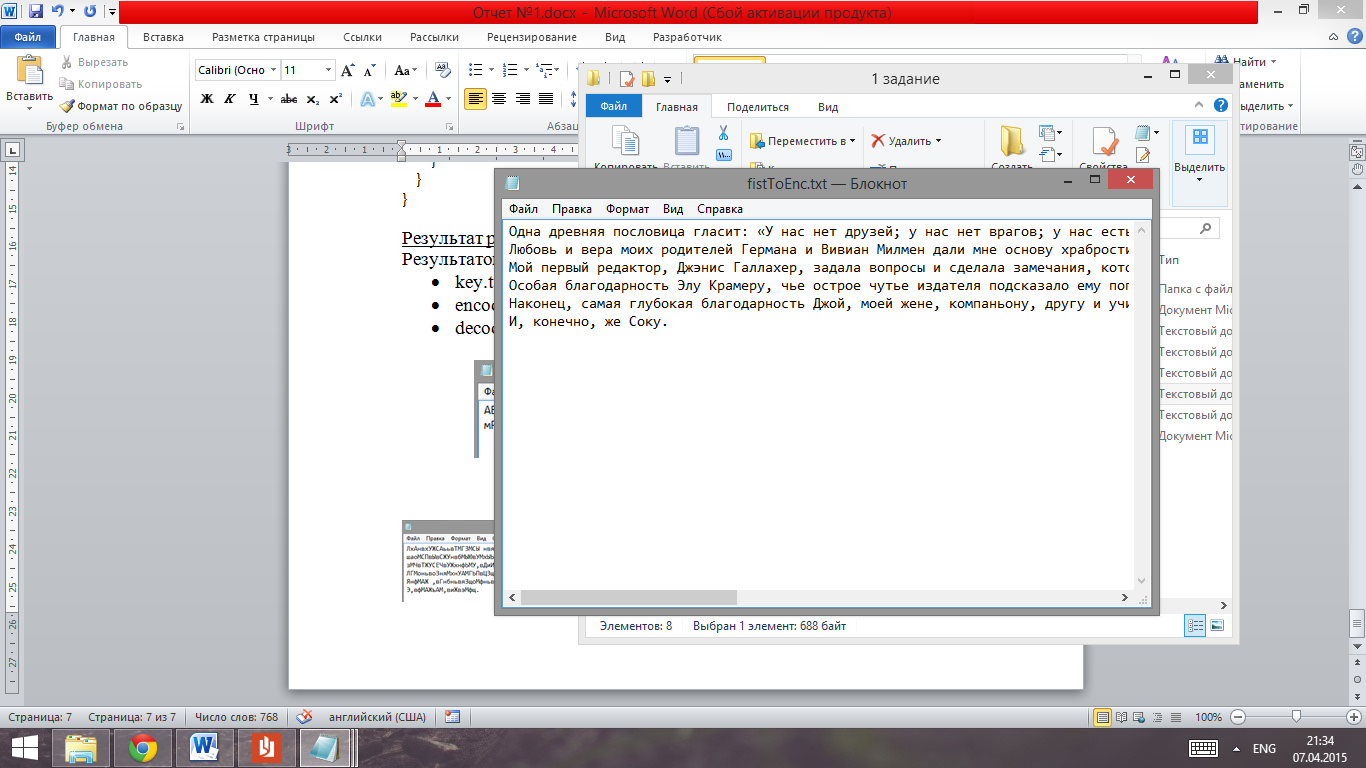


Рисунок 2. Содержимое файла для шифрования

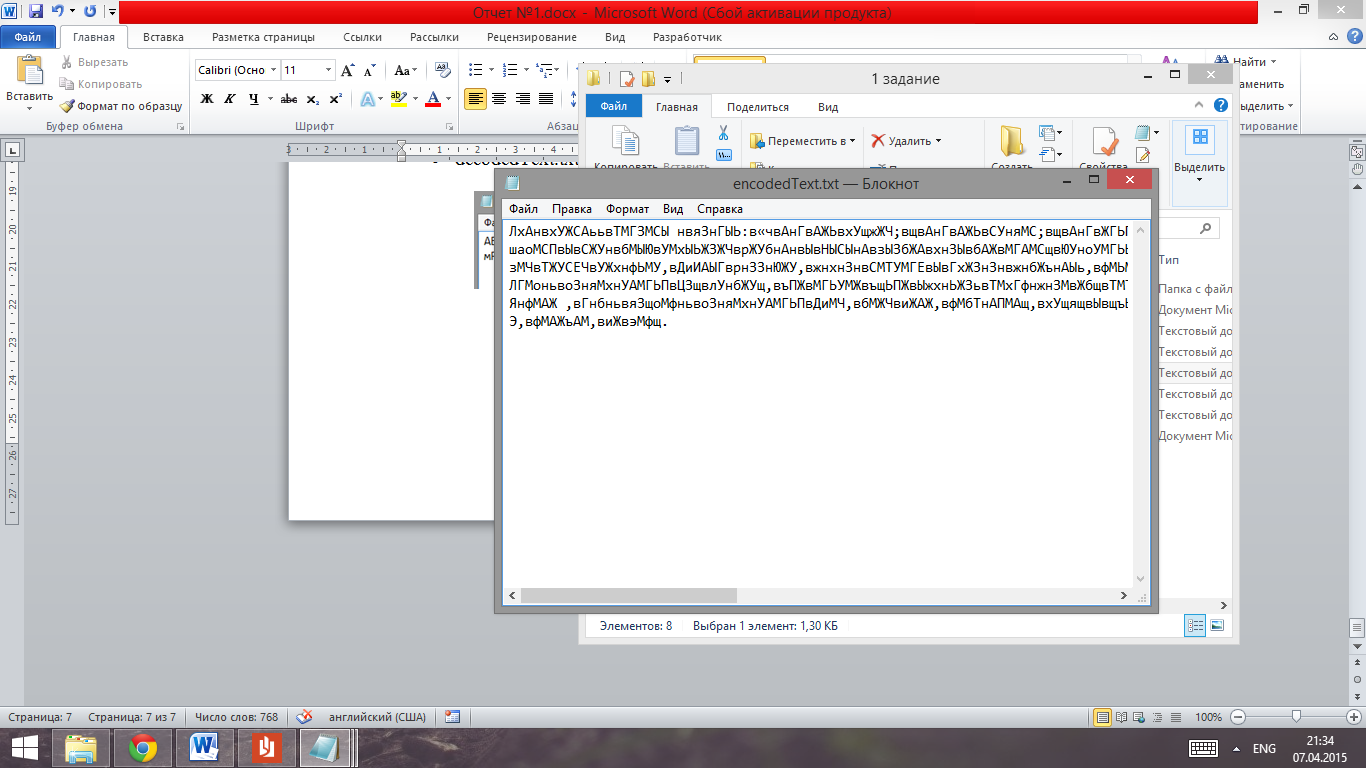


Рисунок 3. Содержимое файла encodedText.txt

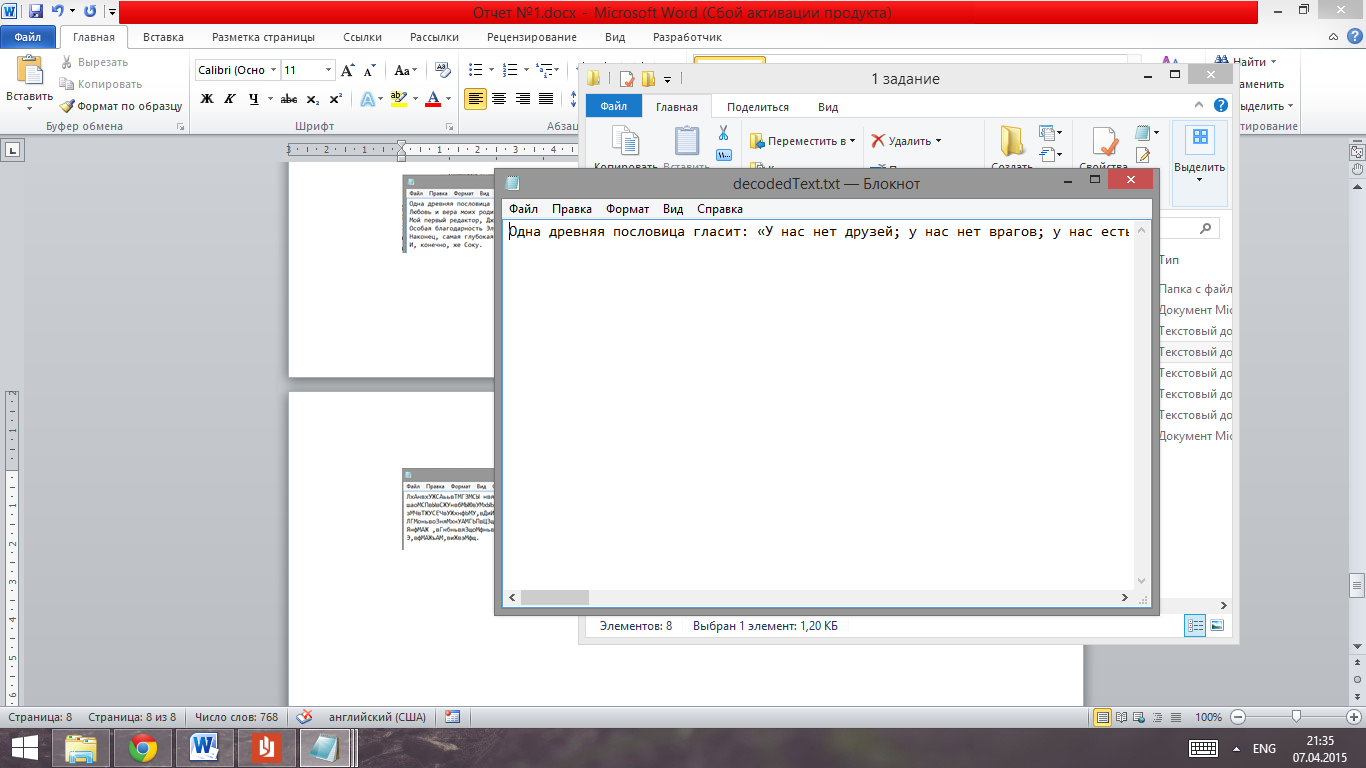


Рисунок 4. Содержимое файла decodedText.txt

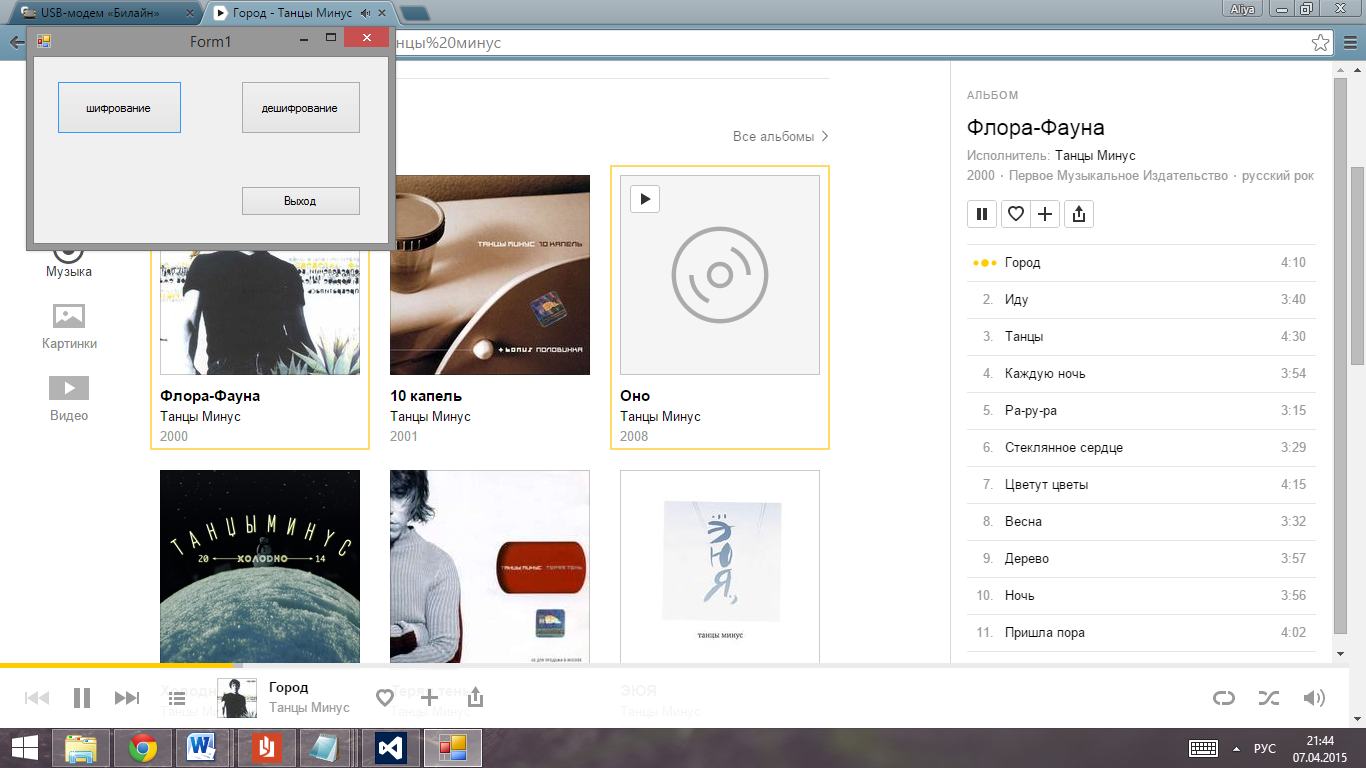


Рисунок 5. Окно работы программы

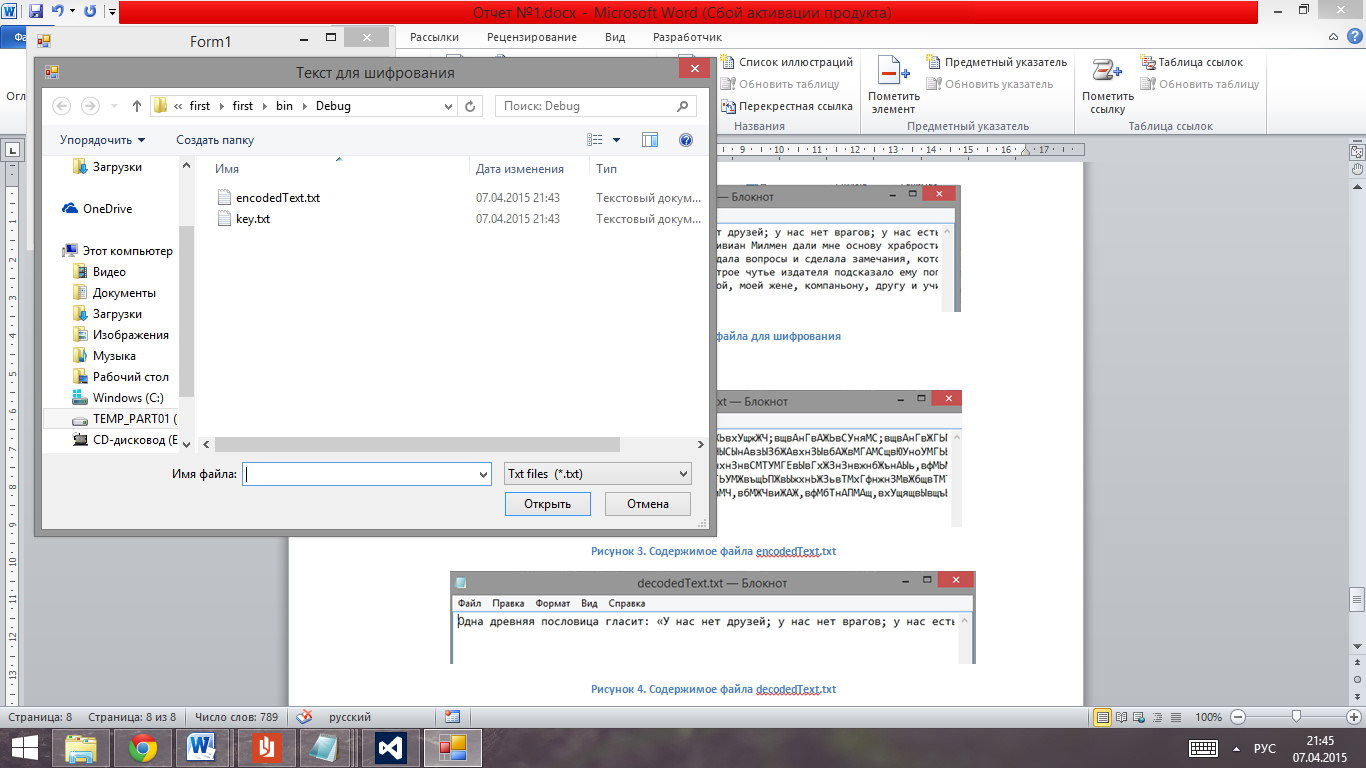


Рисунок 6. Окно выбора текста для шифрования

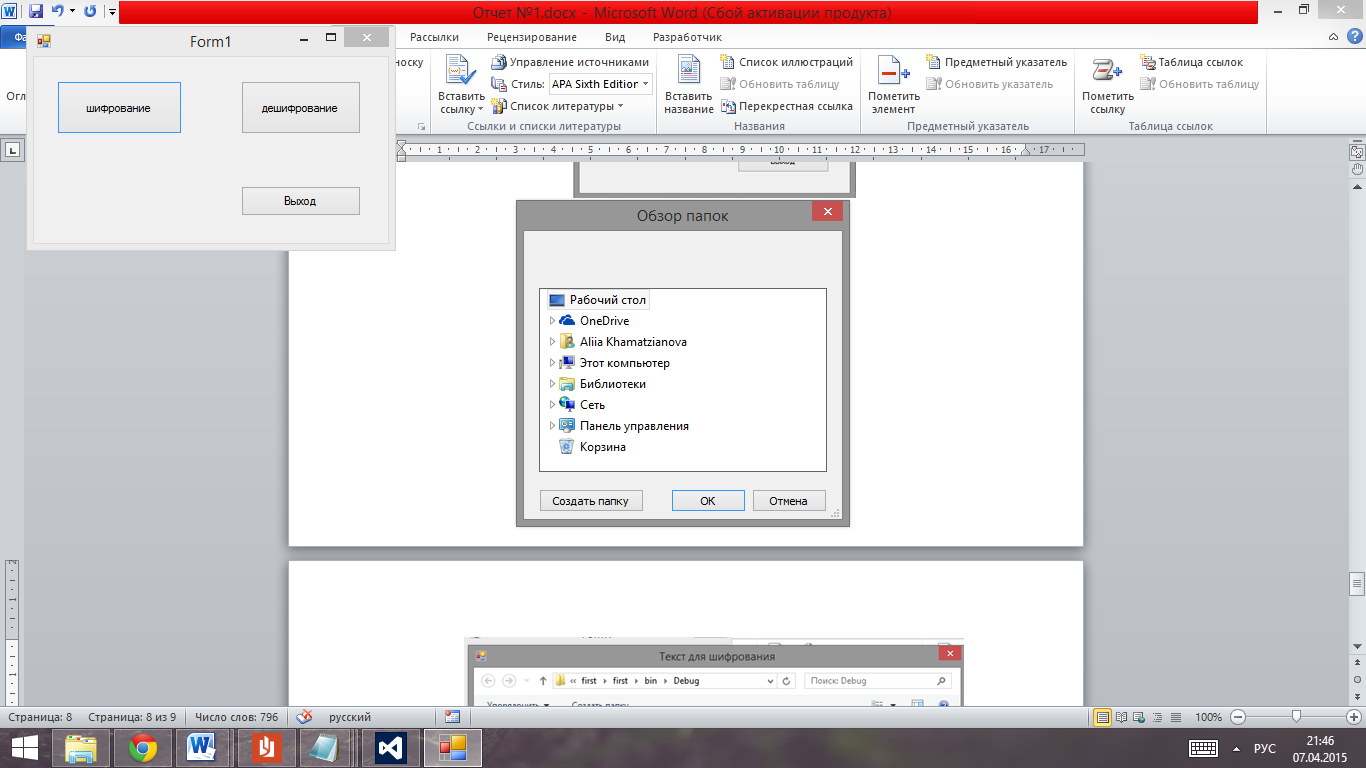


Рисунок 7. Окно выбора папки для сохранения результата