СЕВАСТОПОЛЬСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ЯДЕРНОЙ ЭНЕРГИИ И ПРОМЫШЛЕННОСТИ

Кафедра КЭЭМ

Отчет №2

По дисциплине: “ТСПП ”

**Вариант №20**

Выполнил:

Студент 421 группы

**Ульянченко Л.А.**

Проверил:

**Артёменко М.А.**

Севастополь

2014

**Задание**: Написать программный продукт для шифровки/дешифровки текста. Программный продукт должен позволять шифровать и дешифровать сообщения. При необходимости ПП должен предоставлять возможность изменять и сохранять параметры шифрования. ПП должен обладать интуитивно-понятным интерфейсом. Для выполнения шифрования [желательно создать класс], Использовать алгоритм Атбаш.

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;

namespace L2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

Encode txt = new Encode(textBox1.Text,int.Parse(textBox3.Text));

if (checkBox2.Checked)

txt.Ecode();

else

txt.Dcode();

textBox2.Text = new string(txt.GetEncodeText());

}

private void checkBox1\_Click(object sender, EventArgs e)

{ if (checkBox1.Checked)

textBox3.Enabled = true;

else

textBox3.Enabled = false;

}

private void checkBox2\_Click(object sender, EventArgs e)

{

if (checkBox2.Checked)

{

label1.Text = "До шифр.:";

label2.Text = "После шифр.:";

}

else

{

label1.Text = "Шифр.:";

label2.Text = "После дешифр.:";

}

}

}

public class Encode

{

private char[] Text;

private char[] DecodeText;

private char[] EncodeText;

private int offset;

bool Mod;

public Encode(string a,int i)

{

Text = a.ToCharArray();

offset = i;

}

public void Ecode()

{

EncodeText = Text;

for (int i = 0; i <EncodeText.Length; i++)

{

if (EncodeText[i].Equals(' ')) continue;

EncodeText[i] = (char)((int)EncodeText[i] + offset);

}

Mod = true;

}

public void Dcode()

{

DecodeText=Text;

for (int i = 0; i < DecodeText.Length; i++)

{

if (DecodeText[i].Equals(' ')) continue;

DecodeText[i] = (char)((int)DecodeText[i] - offset);

}

Mod = false;

}

public char[] GetEncodeText()

{

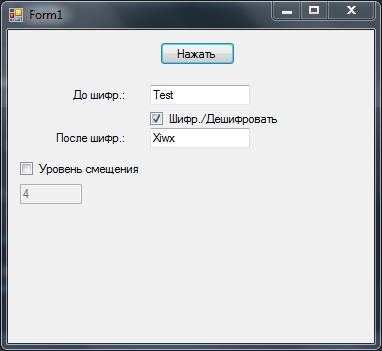
if (Mod) return EncodeText;

return DecodeText;

}

}

}

Шифрование:  


Дешифровка:  
