# Практическая работа № 2

**Шифры перестановки**

**Цели:**

1. Рассмотреть алгоритм шифра Цезаря и выполнить его программную реализацию.

2. Закрепить навыки применения технологии Windows Forms. **Приложение 2.1. Программная реализация шифра замены**

***Задание:*** Разработать приложение с пользовательским интерфейсом, которое позволит осуществлять шифрование и дешифрование текста с использованием шифра Цезаря.

***Константы:*** отсутствуют.

***Переменные:***

1. text – для ввода и вывода текста (тип string)

***Исходные данные:***

1) text – для ввода и вывода текста

***Результат:*** зашифрованное и расшифрованное сообщение

***Код приложения:***

***Листинг 1 – Код приложения MainForm***

namespace CaesarCodeApplication

{

public partial class MainForm : Form

{

CaesarCoder coder = new CaesarCoder();

private string alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ ,.!?";

private int key = 5;

public MainForm()

{

InitializeComponent();

alphabetTextBox.Text = alphabet;

keyTextBox.Text = key.ToString();

}

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

{

// Проверка на наличие алфавита

string alphabet = alphabetTextBox.Text;

if (String.IsNullOrEmpty(alphabet))

{

MessageBox.Show("Алфавит не задан!", "Проверка алфавита", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//Проверка на повторяющиеся символы в алфавите

if (!coder.SetAlphabet(alphabet.ToUpper()))

{

MessageBox.Show("Алфавит содержит повторяющиеся символы!", "Проверка алфавита",MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//Проверка наличия ключа и его тип

int key = 0;

try

{

key = Convert.ToInt32(keyTextBox.Text);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ключ",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

//проверка. Значение ключа не должно превышать мощность алфавита

int n = alphabet.Length;

if (key >= n)

{

MessageBox.Show("Ключ не может быть больше или равным мощности алфавита(" + n.ToString() + ")!", "Ключ", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

// Проверка на наличие исходного сообщения

string text = sourceTextBox.Text;

if (String.IsNullOrEmpty(text))

{

MessageBox.Show("Не задан текст для шифрования!", "Текст для шифрования", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

sourceTextBox.Text = text.ToUpper();

//Шифрование текста

string result = coder.Code(key, text.ToUpper());

сodingResultTextBox.Text = result;

//перенос шифрограммы в поле <<Дешифрование>>

if (resultTransferCheckBox.Checked)

{

source2TextBox.Text = сodingResultTextBox.Text;

}

}

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

{

// Стандартные проверки

string alphabet = alphabetTextBox.Text;

if (String.IsNullOrEmpty(alphabet))

{

MessageBox.Show("Алфавит не задан!", "Проверка алфавита", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (!coder.SetAlphabet(alphabet.ToUpper()))

{

MessageBox.Show("Алфавит содержит повторяющиеся символы!", "Проверка алфавита",MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

int key = 0;

try

{

key = Convert.ToInt32(keyTextBox.Text);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ключ",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

int n = alphabet.Length;

if (key >= n)

{

MessageBox.Show("Ключ не может быть больше или равным мощности алфавита(" + n.ToString() + ")!", "Ключ", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string text = source2TextBox.Text;

if (String.IsNullOrEmpty(text))

{

MessageBox.Show("Не задан текст для шифрования!", "Текст для шифрования", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

sourceTextBox.Text = text.ToUpper();

//Дешифрование текста

string result = coder.Decode(key, text.ToUpper());

decodingResultTextBox.Text = result;

}

}

}

Рисунок 1,2,3 – Результат работы приложения MainForm

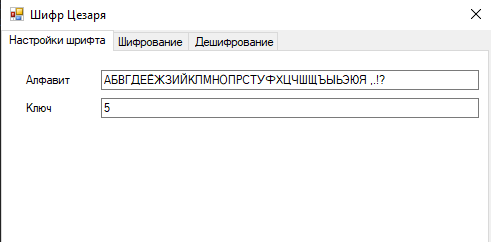


Рисунок 1

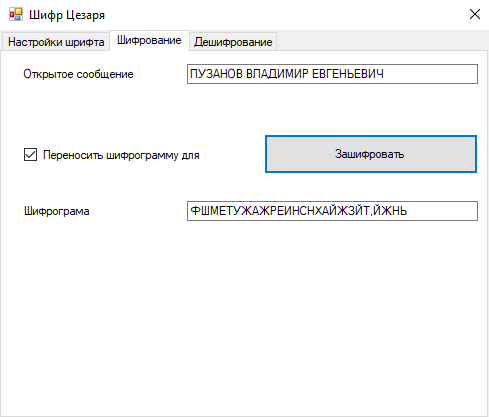


Рисунок 2

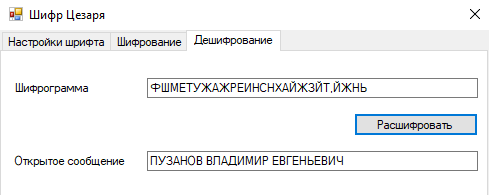


Рисунок 3

**Приложение CaesarCoder**

***Код приложения:***

namespace CaesarCodeApplication

{

class CaesarCoder

{

private Dictionary<char, int> Alphabet = new Dictionary<char, int>();

private Dictionary<int, char> Alphabet2 = new Dictionary<int, char>();

public bool SetAlphabet(string alphabet)

{

//Проверка алфавита на пустоту

if (String.IsNullOrEmpty(alphabet))

{

return false;

}

//отчистка словарей

Alphabet.Clear();

Alphabet2.Clear();

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

{

try

{

Alphabet.Add(alphabet[i], i);

Alphabet2.Add(i, alphabet[i]);

}

catch

{

Alphabet.Clear();

Alphabet2.Clear();

return false;

}

}

return true;

}

public string Code(int k, string text)

{

//Подготовка переменной хранящей результат

string res = "";

if (Alphabet.Count == 0)

{

return res;

}

if (String.IsNullOrEmpty(text))

{

return res;

}

int n = Alphabet.Count;

int value;

int value2;

char symbol;

char symbol2;

//шифрование

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

{

symbol = text[i];

try

{

value = Alphabet[symbol];

value2 = (value + k) % n;

symbol2 = Alphabet2[value2];

res += symbol2;

}

catch

{

res = "";

System.Windows.Forms.MessageBox.Show("Символ " + symbol + " несодержится в алфавите!", "Шифрование", System.Windows.Forms.MessageBoxButtons.OK, System.Windows.Forms.MessageBoxIcon.Error);

return res;

}

}

return res;

}

public string Decode(int k, string text)

{

//Подготовка переменной хранящей результат

string res = "";

if (Alphabet.Count == 0)

{

return res;

}

if (String.IsNullOrEmpty(text))

{

return res;

}

int n = Alphabet.Count;

int value;

int value2;

char symbol;

char symbol2;

//дешифрование

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

{

symbol = text[i];

try

{

value = Alphabet[symbol];

value2 = (value - k + n) % n;

symbol2 = Alphabet2[value2];

res += symbol2;

}

catch

{

res = "";

System.Windows.Forms.MessageBox.Show("Символ " + symbol + " несодержится в алфавите!", "Шифрование", System.Windows.Forms.MessageBoxButtons.OK, System.Windows.Forms.MessageBoxIcon.Error);

return res;

}

}

return res;

}

}

}

Выполнил студент Пузанов В. Е., ФИТУ 010304-КМСб-о22

Проверил ст. преподаватель каф. ПМ Черноиван Д.Н.