Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.04 – «Программная инженерия»

Дисциплина: «Защита информации»

Профиль: «Разработка программно-информационных систем»

Семестр 5

ОТЧЕТ

по лабораторной работе №1

Тема: «Шифры перестановки и замены»

Выполнил: студент группы РИС-22-1б

Баяндин К. С. \_\_\_\_\_\_\_\_\_\_\_

Проверил: старший преподаватель кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_

Пермь, 2024

**ЦЕЛЬ РАБОТЫ**

Получить практические навыки по применению шифров перестановки и шифров простой замены.

**ЗАДАНИЕ**

Вариант 8

Реализовать шифрование текстового сообщения, используя систему шифрования Цезаря.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**Система шифрования Цезаря**

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки). Свое название этот шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке с Цицероном (около 50 г. до н.э.).

При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу. Заменяющая буква определялась путем смещения по алфавиту от исходной буквы на К букв. При достижении конца алфавита выполнялся циклический переход к его началу. Цезарь использовал шифр замены при смещении К = 3. Такой шифр замены можно задать таблицей подстановок, содержащей соответствующие пары букв открытого текста и шифртекста. Совокупность возможных подстановок для К = 3 показана в следующей таблице.

|  |  |  |
| --- | --- | --- |
| A -> D | J -> M | S -> V |
| B -> E | K -> N | T -> W |
| С -> F | L -> O | U -> X |
| D -> G | M -> P | V -> Y |
| Е -> H | N -> Q | W -> Z |
| F -> I | O -> R | X -> A |
| G -> J | P -> S | Y -> B |
| Н -> K | Q -> T | Z -> C |
| I -> L | R -> U |  |

Например, послание Цезаря

VENI VIDI VICI

в переводе на русский означает "Пришел, Увидел, Победил"), направленное его другу Аминтию после победы над понтийским царем Фарнаком, сыном Митридата, выглядело бы в зашифрованном виде так:

YHQL YLGL YLFL

Концепция, заложенная в систему шифрования Цезаря, оказалась весьма плодотворной, о чем свидетельствуют ее многочисленные модификации. Несколько таких модификаций будут рассмотрены ниже.

**ХОД РАБОТЫ**

Для работы был выбран фреймворк Windows Forms на C#.

Были созданы 3 поля: 2 для ввода – первый для текста, второй для порядка сдвига, и 1 для вывода. Также была создана кнопка *Шифрование*, при нажатии на которую текст из 1 поля ввода преобразуется в шифр системы Цезаря при смещении на *K* – число, определяющее порядок сдвига по алфавиту, которое задается во 2 поле ввода. Результат выводится в поле вывода текста (рисунок 1).

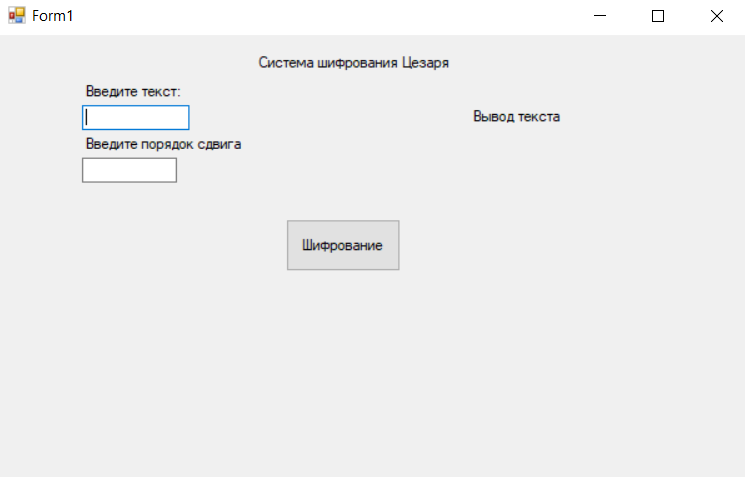


Рисунок 1 – Форма для шифрования.

Логика шифрования представлена в функции *Code().* Создаются переменные *inputText* и *K* для текста и порядка сдвига соответственно. С помощью цикла перебираются все символы *inputText*, каждый символ преобразуется в числовое значение в соответствии с таблицей ASCII, к коду символа добавляется порядок (*K*) и происходит обратная операция перевода из числового кода в символ. В случае пробела суммирование порядка не происходит. Также есть проверки на граничные символы (последние в алфавите, переход по порядку которых приведет к выходу из алфавита), которые не дают выйти за границы алфавита.

Программа работает с латиницей и кириллицей.

Пример работы программы представлен на рисунке 2.

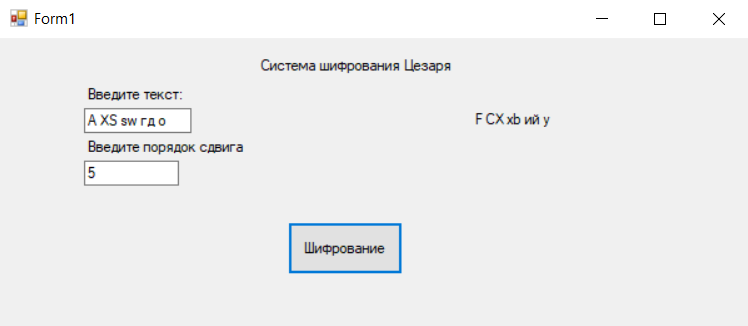


Рисунок 2 – Пример работы программы.

**ПРИЛОЖЕНИЕ А**

**Листинг 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;

namespace \_1lab

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

// Выводим текст в метку

label1.Text = shifr();

}

private string shifr()

{

// Получаем текст из текстового поля

string inputText = textBox1.Text;

string K = textBox2.Text;

int k = int.Parse(K);

char[] arr = new char[inputText.Length];

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

{

int index = (int)inputText[i];

if (inputText[i] == ' ')

arr[i] = ' ';

else

{

index += k;

char sim = (char)index;

arr[i] = sim;

if (arr[i] == '[')

arr[i] = 'A';

if (arr[i] == '\\')

arr[i] = 'B';

if (arr[i] == ']')

arr[i] = 'C';

if (arr[i] == '{')

arr[i] = 'a';

if (arr[i] == '|')

arr[i] = 'b';

if (arr[i] == '}')

arr[i] = 'c';

if (arr[i] == 'а')

arr[i] = 'А';

if (arr[i] == 'б')

arr[i] = 'Б';

if (arr[i] == 'в')

arr[i] = 'В';

}

}

string shifr = new string(arr);

return shifr;

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

}

}