Задание по технологиям защиты информации

На тему: Древние шифры

Горшков Ян

**Задание:**

Построить графическую программу, позволяющую шифровать и дешифровать исходный текст моноалфавитным шифром и квадратом Полибия

**Теоретические сведения:**

Моноалфавитный шифр — класс методов шифрования, которые сводятся к созданию по определённому алгоритму таблицы шифрования, в которой для каждой буквы открытого текста существует единственная сопоставленная ей буква шифр-текста. Само шифрование заключается в замене букв согласно таблице. Для расшифровки достаточно иметь ту же таблицу, либо знать алгоритм, по которой она генерируется.

К шифрам простой замены относятся многие способы шифрования, возникшие в древности или средневековье, как, например, Атбаш (также читается как этбаш) или Шифр Цезаря. Для вскрытия подобных шифров используется частотный криптоанализ. Пример такого шифра изображен на рисунке 1.

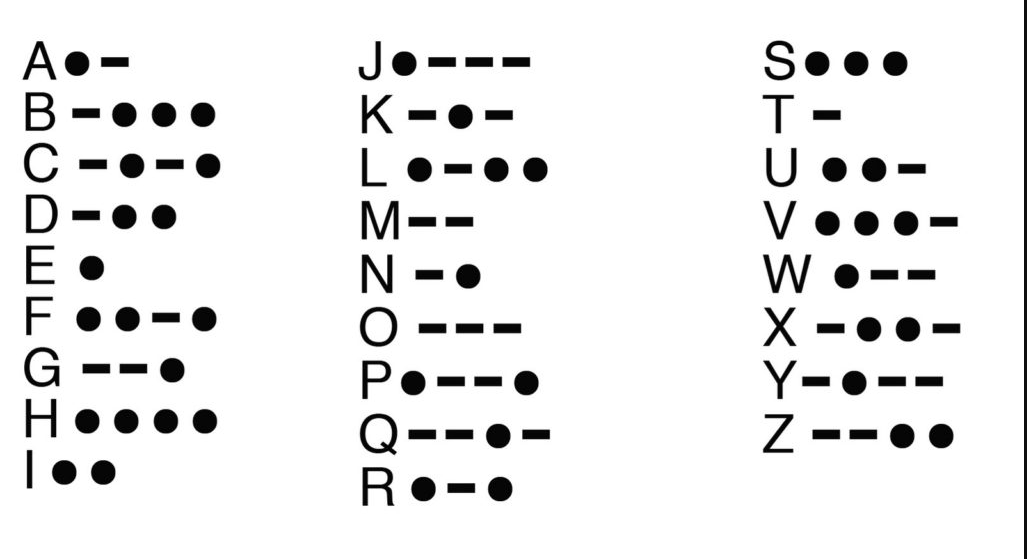


Рисунок 1 – Азбука Морзе как моноалфавитный шифр

В криптографии квадрат Полибия, также известный как шахматная доска Полибия — оригинальный код простой замены, одна из древнейших систем кодирования, предложенная Полибием. Данный вид кодирования изначально применялся для греческого алфавита, но затем был распространен на другие языки.

К каждому языку отдельно составляется таблица шифрования с количеством пронумерованных строк и столбцов, параметры которой зависят от его мощности (количества букв в алфавите). Берутся два целых числа, произведение которых ближе всего к количеству букв в языке — получаем нужное число строк и столбцов. Затем вписываем в таблицу все буквы алфавита подряд — по одной в каждую клетку. При нехватке клеток можно вписать в одну две буквы (редко употребляющиеся или схожие по употреблению). Квадрат Полибия для английского алфавита изображен на рисунке 2.

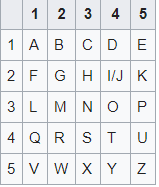


Рисунок 2 – Латинский алфавит в квадрате Полибия

Существует 2 методы шифрования при помощи квадрата Полибия:

В первом методе для шифрования на квадрате находили букву текста и вставляли в шифровку нижнюю от неё в том же столбце. Если буква была в нижней строке, то брали верхнюю из того же столбца. Таким образом слова SOMETEXT будут преобразованы в XTRKYKCY. Принцип данного метода изображен на рисунке 3.

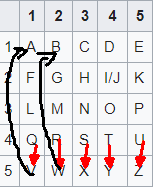


Рисунок 3 – Первый метод

Во втором методе сообщение преобразуется в координаты по квадрату Полибия, и координаты записываются вертикально. Результат данного этапа изображен на рисунке 4.

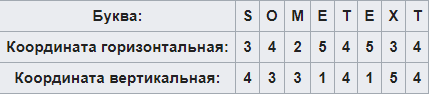


Рисунок 4 – Первый этап шифрования

Затем координаты считывают по строкам, в итоге получая пары цифр. Например: 34 25 45 34 43 31 41 54. Последним шагом остается записать данные пары в таблицу, и получить новые буквы. Последний шаг изображен на рисунке 5.

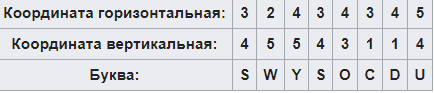


Рисунок 5 – Результат шифрования

**Разработанная программа:**

Программа разрабатывалась на языке C# при помощи Windows Forms. Интерфейс программы изображен на рисунке 6.

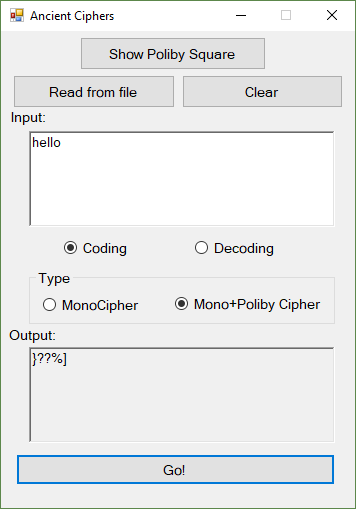


Рисунок 6 – Интерфейс программы

Программа позволяет шифровать и дешифровать текст английском языке. Шифруемый текст может быть введен как вручную в текстовое поле, так и прочитан из текстового файла. Кнопка “Clear” позволяет очистить текстовое поле. Программа работает как шифратор, так и дешифратор в зависимости от выбора пользователя. Также, программа может использовать как моноалфавитный шифр, так и совмещенный моноалфавитный с квадратом Полибия. Для просмотра квадрата Полибия выделена специальная кнопка. Результат нажатия изображен на рисунке 7.

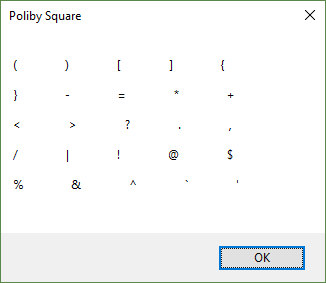


Рисунок 7 – Квадрат Полибия для моноалфавитного шифра

**Вывод:**

В данной практической работе была разработана программа шифратор-дешифратор, работающая по принципу моноалфавитного шифра и квадрата Полибия. Данная работа позволила понять принцип шифрования, реализовать его при помощи языка программирования, и проверить работоспособность на практике.

**Листинг:**

MainForm.cs:

using System;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace OldCodes

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

MonoPoliby coder = new MonoPoliby();

//Выбор режима

if (radioButton\_Code.Checked)

{

//Выбор вида шифрации

if (radioButton\_Mono.Checked)

richTextBox\_Out.Text = coder.encodeToMonoCipher(richTextBox\_In.Text.ToLower());

else

richTextBox\_Out.Text = coder.encodeToPoliby(coder.encodeToMonoCipher(richTextBox\_In.Text.ToLower()));

}

else

{

//Выбор вида дешифрации

if (radioButton\_Mono.Checked)

richTextBox\_Out.Text = coder.decodeFromMonoCipher(richTextBox\_In.Text.ToLower());

else

richTextBox\_Out.Text = coder.decodeFromMonoCipher(coder.decodeFromPoliby(richTextBox\_In.Text.ToLower()));

}

}

/\*

\* Вывод квадрата Полибия

\*/

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

{

MonoPoliby coder = new MonoPoliby();

MessageBox.Show(coder.getPolibySquare(),"Poliby Square");

}

/\*

\* Чтение текста из файла в текстовое поле

\*/

private void button\_Read\_Click(object sender, EventArgs e)

{

try

{

OpenFileDialog open = new OpenFileDialog();

if (open.ShowDialog() == DialogResult.OK)

{

string path = open.FileName;

richTextBox\_In.Text = File.ReadAllText(path,Encoding.Default);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

/\*

\* Очистка текстовых полей

\*/

private void button\_Clear\_Click(object sender, EventArgs e)

{

richTextBox\_In.Text = String.Empty;

richTextBox\_Out.Text = String.Empty;

}

}

}

MonoPoliby.cs:

using System;

using System.Collections.Generic;

namespace OldCodes

{

class MonoPoliby

{

private Dictionary<char, char> monoCipher = new Dictionary<char, char>();

private char[,] polibySquare = new char[5, 5];

public MonoPoliby()

{

monoCipher.Add('a', '(');

monoCipher.Add('b', ')');

monoCipher.Add('c', '[');

monoCipher.Add('d', ']');

monoCipher.Add('e', '{');

monoCipher.Add('f', '}');

monoCipher.Add('g', '-');

monoCipher.Add('h', '=');

monoCipher.Add('i', '\*');

monoCipher.Add('k', '+');

monoCipher.Add('l', '<');

monoCipher.Add('m', '>');

monoCipher.Add('n', '?');

monoCipher.Add('o', '.');

monoCipher.Add('p', ',');

monoCipher.Add('q', '/');

monoCipher.Add('r', '|');

monoCipher.Add('s', '!');

monoCipher.Add('t', '@');

monoCipher.Add('u', '$');

monoCipher.Add('v', '%');

monoCipher.Add('w', '&');

monoCipher.Add('x', '^');

monoCipher.Add('y', '`');

monoCipher.Add('z', '\'');

createPolibySquare();

}

/\*

\* Запись моноалфавитного шифра в квадрат Полибия

\*/

private void createPolibySquare()

{

int i = 0, j = 0;

foreach (var value in monoCipher.Values)

{

if (j == 5)

{

j = 0;

i++;

}

polibySquare[i, j] = value;

j++;

}

}

/\*

\* Метод кодирующий входную строку в моноалфавитный шифр

\*/

public string encodeToMonoCipher(string word)

{

string result = String.Empty;

foreach (char letter in word)

result += monoCipher[letter];

return result;

}

/\*

\* Метод декодирующий входную строку из моноалфавитного шифра

\* в англоязычную строку

\*/

public string decodeFromMonoCipher(string code)

{

string result = String.Empty;

foreach (var symbol in code)

foreach (var key in monoCipher.Keys)

if (monoCipher[key] == symbol)

result += key;

return result;

}

/\*

\* Кодирование из моношифра в шифр Полибия

\*/

public string encodeToPoliby(string monoCode)

{

string result = String.Empty;

string temp = String.Empty;

foreach (var symbol in monoCode)

for (int i = 0; i < 5; i++)

for (int j = 0; j < 5; j++)

if (polibySquare[i, j] == symbol)

{

temp += i.ToString() + j;

break;

}

string t1 = String.Empty,t2=String.Empty;

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

if (i % 2 == 0)

t1 += temp[i];

else

t2 += temp[i];

temp = t1 + t2;

for (int i = 0; i < temp.Length; i+=2)

result += polibySquare[int.Parse(temp[i].ToString()), int.Parse(temp[i+1].ToString())];

return result;

}

/\*

\* Декодирование из шифра Полибия в моношифр

\*/

public string decodeFromPoliby(string polibyCode)

{

string temp=String.Empty,result = String.Empty;

foreach (var symbol in polibyCode)

for (int i = 0; i < 5; i++)

for (int j = 0; j < 5; j++)

if (polibySquare[i, j] == symbol)

{

temp += i.ToString() + j;

break;

}

for (int i = 0; i < temp.Length / 2; i++)

result +=polibySquare[int.Parse(temp[i].ToString()), int.Parse(temp[i + temp.Length / 2].ToString())];

return result;

}

/\*

\* Преобразования квадрата ПОлибия для моноалфавитного шрифта в строку

\*/

public string getPolibySquare()

{

string square = string.Empty;

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

square += polibySquare[i, j].ToString() + " ";

}

square += "\n\n";

}

return square;

}

}

}