**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1**

по дисциплине «Основы защита информации»

на тему: «Шифрование и дешифрование на основе классических криптосистем»

Вариант 13

Выполнил: студент гр. ИТП-31

Матарас А.В.

Принял: доцент

Кравченко О.А.

Гомель 2021

**Цель работы:** освоить основные понятия криптологии, научиться применять простейшие криптосистемы.

**Задание**

Разработать приложение, выполняющее шифрование и дешифрование текста в соответствии с заданным алгоритмом (табл.1).

Протестировать работу приложения на не менее чем трех текстах от 10 до 20 символов.

Таблица 1

|  |  |
| --- | --- |
| Криптографическая система | Вариант |
| **Задание 1** | |
| Шифр Цезаря (шаг сдвига соответствует варианту). | Четные варианты |
| Цифровая система тайнописи (соответствие цифр и букв выбрать самостоятельно). | Нечетные варианты |
| **Задание 2** | |
| Квадрат Полибия (для латинского алфавита) | 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31 |
| Квадрат Полибия (для русского алфавита) | 2, 5, 8, 11, 14, 17, 20, 23, 26, 29 |
| Магический квадрат | 3, 6, 9, 12, 15, 18, 21, 24, 27, 30 |

На рисунке 1 представлено шифрование методом «Цифровая система тайнописи».

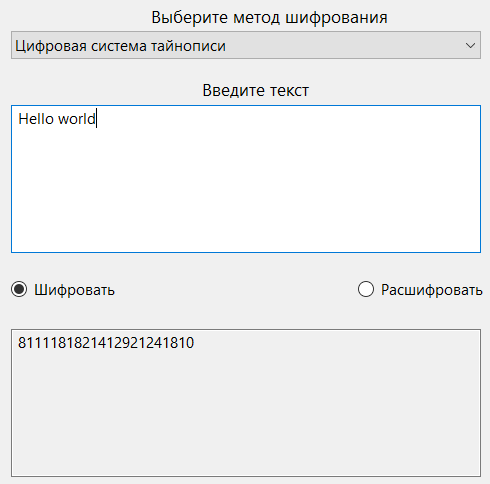


Рисунок 1 – Результат шифрования

На рисунке 2 представлено дешифрование методом «Цифровая система тайнописи».

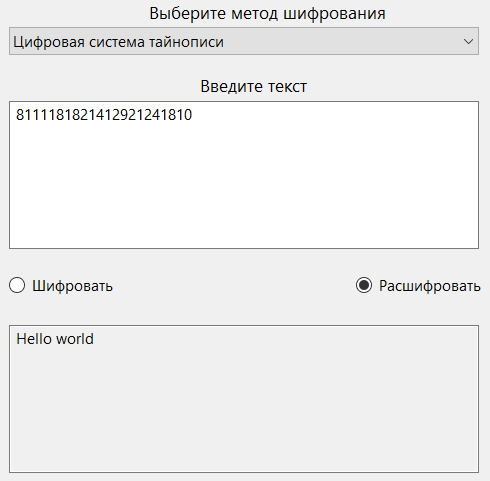


Рисунок 2 – Результат дешифрования

На рисунке 3 представлено шифрование методом «Квадрат Полибия» (для латинского алфавита).

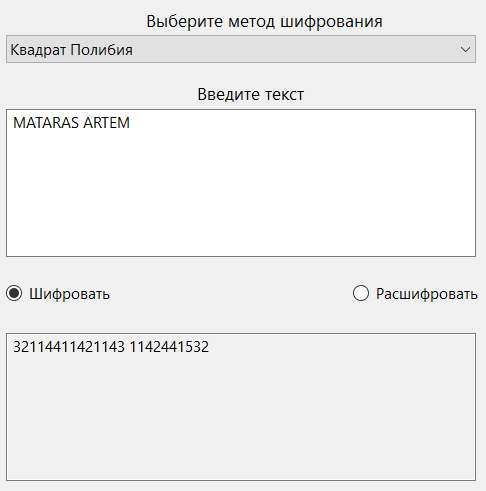


Рисунок 3 – Результат шифрования

На рисунке 4 представлено дешифрование методом «Квадрат Полибия» (для латинского алфавита).

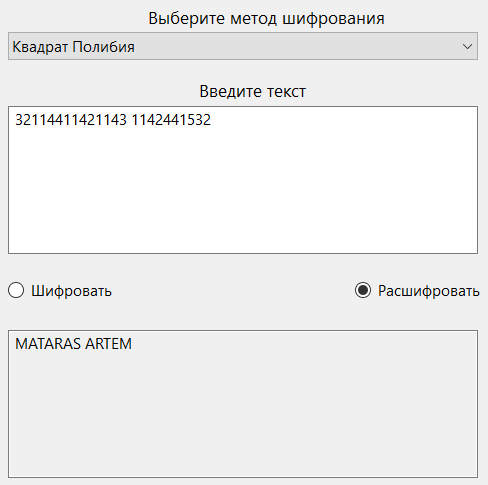


Рисунок 4 – Результат дешифрования

Листинг программы в приложении А.

**Вывод:** в ходе выполнения лабораторной работы получены навыки шифрования и дешифрования текста, освоены основные методы шифрования: «шифр Цезаря», «цифровая система тайнописи», «квадрат Полибия», «магический квадрат».

**Приложение А**

**Листинг программы**

**Класс *DigitalCipherStrategy*** // Цифровая система тайнописи

using System;

using System.Collections.Generic;

using System.Text;

namespace СipherLibrary

{

public class DigitalCipherStrategy : CipherStrategy

{

Dictionary<char, int> tokens;

Dictionary<int, char> inversedTokens;

/// <summary>

/// Цифровая система тайнописи

/// </summary>

public DigitalCipherStrategy()

{

tokens = new Dictionary<char, int>();

inversedTokens = new Dictionary<int, char>();

for (int i = 40; i < 100; i++) // Добавляю в словарь символы и их числовое представление

{

tokens.Add((char)(i - 9), i);

inversedTokens.Add(i, (char)(i - 9));

}

for(int i = 10; i < 40; i++)

{

tokens.Add((char)(i + 90), i);

inversedTokens.Add(i, (char)(i + 90));

}

}

public string Decrypt(string textToDecrypt)

{

StringBuilder builder = new StringBuilder(textToDecrypt.Length);

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

{

if(i == textToDecrypt.Length - 1)

{

builder.Append(i.ToString());

}

try

{

int num = Convert.ToInt32(textToDecrypt.Substring(i, 2));

builder.Append(inversedTokens[num]);

}

catch

{

builder.Append(textToDecrypt[i]);

i -= 1;

}

}

return builder.ToString();

}

public string Encrypt(string textToEncrypt)

{

StringBuilder builder = new StringBuilder(textToEncrypt.Length \* 2);

foreach (char c in textToEncrypt)

{

try

{

builder.Append(tokens[c].ToString());

}

catch

{

builder.Append(c);

}

}

return builder.ToString();

}

}

}

**Класс *PolybiusSquareStrategy*** //Квадрат Полибия

using System;

using System.Collections.Generic;

using System.Text;

namespace СipherLibrary

{

public class PolybiusSquareStrategy : CipherStrategy

{

char[,] charMatrix;

public PolybiusSquareStrategy()

{

charMatrix = new char[,]

{

{'A', 'B', 'C', 'D', 'E' },

{'F', 'G', 'H', 'I', 'K' },

{'L', 'M', 'N', 'O', 'P' },

{'Q', 'R', 'S', 'T', 'U' },

{'V', 'W', 'X', 'Y', 'Z' }

};

}

public string Decrypt(string textToDecrypt)

{

StringBuilder builder = new StringBuilder(textToDecrypt.Length);

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

{

if (i == textToDecrypt.Length - 1)

{

builder.Append(i.ToString());

}

try

{

int row = Convert.ToInt32(textToDecrypt.Substring(i, 1));

int column = Convert.ToInt32(textToDecrypt.Substring(i + 1, 1));

builder.Append(charMatrix[row - 1, column - 1]);

}

catch

{

builder.Append(textToDecrypt[i]);

i -= 1;

}

}

return builder.ToString();

}

public string Encrypt(string textToEncrypt)

{

textToEncrypt = textToEncrypt.ToUpper();

StringBuilder builder = new StringBuilder();

foreach(char c in textToEncrypt)

{

if (c == 'J')

builder.Append("24");

string coords = getSymbolCoordsInStringFormat(c);

if(coords.Equals(""))

{

builder.Append(c);

}

else

{

builder.Append(coords);

}

}

return builder.ToString();

}

private string getSymbolCoordsInStringFormat(char c)

{

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

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

{

if (c == charMatrix[i, j])

return ((i + 1) \* 10 + (j + 1)).ToString();

}

return "";

}

}

}