Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| *институт* |
| Кафедра Прикладной математики и компьютерной безопасности |
| *кафедра* |

**ЛАБОРАТОРНАЯ РАБОТА №9**

|  |
| --- |
| **Шифр Вижинера** |
| *тема* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Руководитель | | |  |  |  | В.И.Вайнштейн |
|  | |  |  | *подпись, дата* |  | *инициалы, фамилия* |
| Студент | КИ15-01 №031508683 | |  |  |  | М.С.Димаксян |
|  | *номер группы, зачетной книжки* | |  | *подпись, дата* |  | *инициалы, фамилия* |

Красноярск 2019

**Задание:**

Реализовать шифр Вижинера на любом языке программирования.

**Теория:**

Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Например, предположим, что исходный текст имеет такой вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

LEMONLEMONLE

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN

Ключ: LEMONLEMONLE

Зашифрованный текст: LXFOPVEFRNHR

**Исходный код:**

**Алфавит:**

string ALF\_RU = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

string alf\_ru = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

string ALF\_EN = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

string alf\_en = "abcdefghijklmnopqrstuvwxyz";

**Преобразование ключа:**

private int Keyword\_symb\_to\_Int\_Rotate(char symb)

{

int num = -1;

bool rus = false;

if (alf\_ru.IndexOf(symb) > num) { num = alf\_ru.IndexOf(symb); rus = true; }

if (ALF\_RU.IndexOf(symb) > num) { num = ALF\_RU.IndexOf(symb); rus = true; }

if (alf\_en.IndexOf(symb) > num) num = alf\_en.IndexOf(symb);

if (ALF\_EN.IndexOf(symb) > num) num = ALF\_EN.IndexOf(symb);

if (rus)

num = num % 33;

else if (num != -1)

num = num % 26;

return num;

}

**Алгоритм зашифрования:**

private void f9\_ButtonEncrypt\_Click(object sender, EventArgs e)

{

string Source = f9\_fieldOriginal.Text.Replace("\r\n", "\r");

char[] message = Source.ToCharArray();

var entered\_key = Regex.Replace(f9\_fieldKeyword.Text, "[^a-zA-Zа-яА-ЯёЁ]", "");

f9\_fieldKeyword.Text = entered\_key;

if (entered\_key.Length == 0)

{

MessageBox.Show("Некорректный ключ.", "Ошибка");

goto metka\_exit;

}

int keyPos = 0;

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

{

int one\_time\_shift = Keyword\_symb\_to\_Int\_Rotate(entered\_key[keyPos]); //Сдвиг для текущей буквы

if (ALF\_RU.IndexOf(message[i]) >= 0)

{

message[i] = ALF\_RU[(ALF\_RU.IndexOf(message[i]) + one\_time\_shift) % ALF\_RU.Length];

}

if (alf\_ru.IndexOf(message[i]) >= 0)

{

message[i] = alf\_ru[(alf\_ru.IndexOf(message[i]) + one\_time\_shift) % alf\_ru.Length];

}

if (ALF\_EN.IndexOf(message[i]) >= 0)

{

message[i] = ALF\_EN[(ALF\_EN.IndexOf(message[i]) + one\_time\_shift) % ALF\_EN.Length];

}

if (alf\_en.IndexOf(message[i]) >= 0)

{

message[i] = alf\_en[(alf\_en.IndexOf(message[i]) + one\_time\_shift) % alf\_en.Length];

}

keyPos++;

if (keyPos == entered\_key.Length) keyPos = 0;

}

string result = new string(message);

f9\_fieldCiphertext.Text = result.Replace("\r", "\r\n"); ; ;

metka\_exit:;

}

**Алгоритм расшифрования:**

private void f9\_ButtonDecrypt\_Click(object sender, EventArgs e)

{

string Source = f9\_fieldCiphertext.Text.Replace("\r\n", "\r");

char[] message = Source.ToCharArray();

var entered\_key = Regex.Replace(f9\_fieldKeyword.Text, "[^a-zA-Zа-яА-ЯёЁ]", "");

f9\_fieldKeyword.Text = entered\_key;

if (entered\_key.Length == 0)

{

MessageBox.Show("Некорректный ключ.", "Ошибка");

goto metka\_exit;

}

int keyPos = 0;

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

{

int one\_time\_shift = Keyword\_symb\_to\_Int\_Rotate(entered\_key[keyPos]); //Сдвиг для текущей буквы

if (ALF\_RU.IndexOf(message[i]) >= 0)

{

message[i] = ALF\_RU[(ALF\_RU.IndexOf(message[i]) - one\_time\_shift + ALF\_RU.Length) % ALF\_RU.Length];

}

if (alf\_ru.IndexOf(message[i]) >= 0)

{

message[i] = alf\_ru[(alf\_ru.IndexOf(message[i]) - one\_time\_shift + alf\_ru.Length) % alf\_ru.Length];

}

if (ALF\_EN.IndexOf(message[i]) >= 0)

{

message[i] = ALF\_EN[(ALF\_EN.IndexOf(message[i]) - one\_time\_shift + 2\*ALF\_EN.Length) % ALF\_EN.Length];

}

if (alf\_en.IndexOf(message[i]) >= 0)

{

message[i] = alf\_en[(alf\_en.IndexOf(message[i]) - one\_time\_shift + 2\*alf\_en.Length) % alf\_en.Length];

}

keyPos++;

if (keyPos == entered\_key.Length) keyPos = 0;

}

string result = new string(message);

f9\_fieldOriginal.Text = result.Replace("\r", "\r\n"); ; ;

metka\_exit:;

}

**Результат работы программы:**

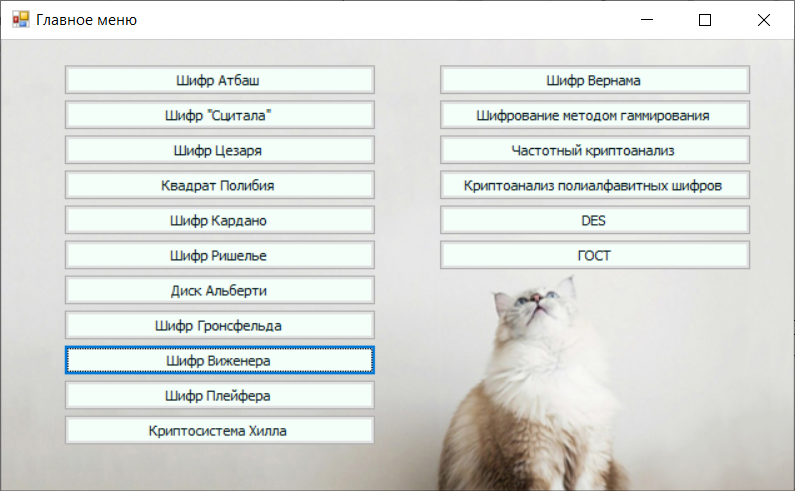


Рисунок 1. Главное меню программы.

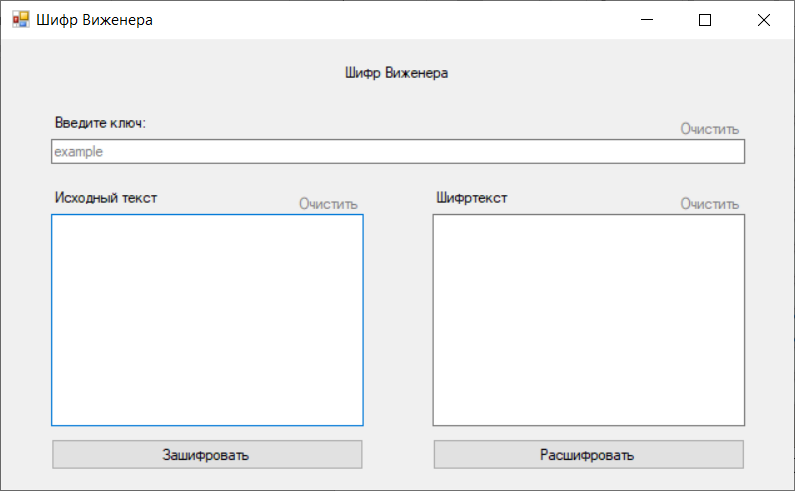


Рисунок 2. Окно шифра Вижинера.

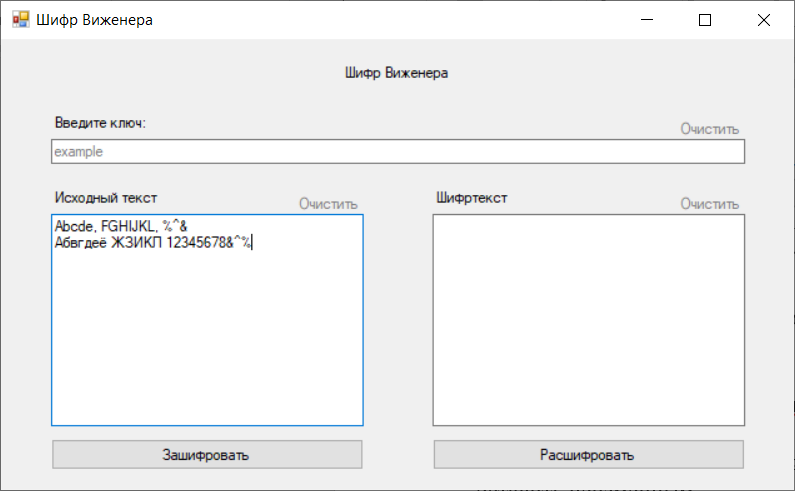


Рисунок 3. Исходный текст.

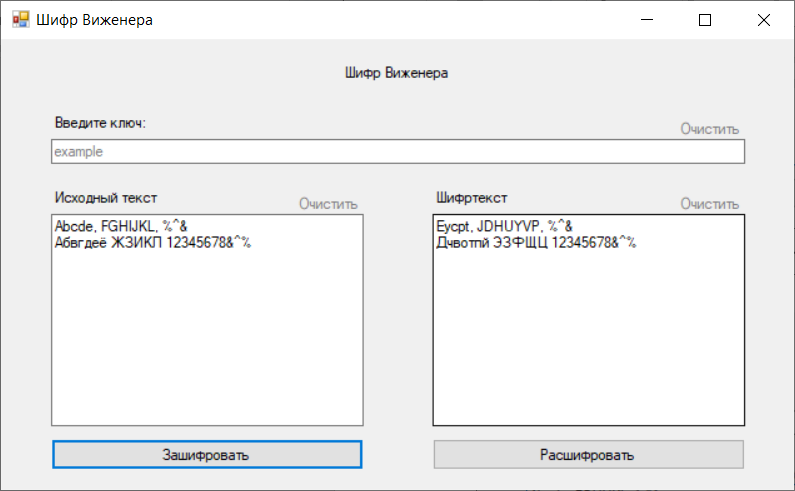


Рисунок 4. Шифртекст (результат зашифрования).

Очистим поле Исходный текст и расшифруем сообщение. Ключ необходимо оставить неизменным.

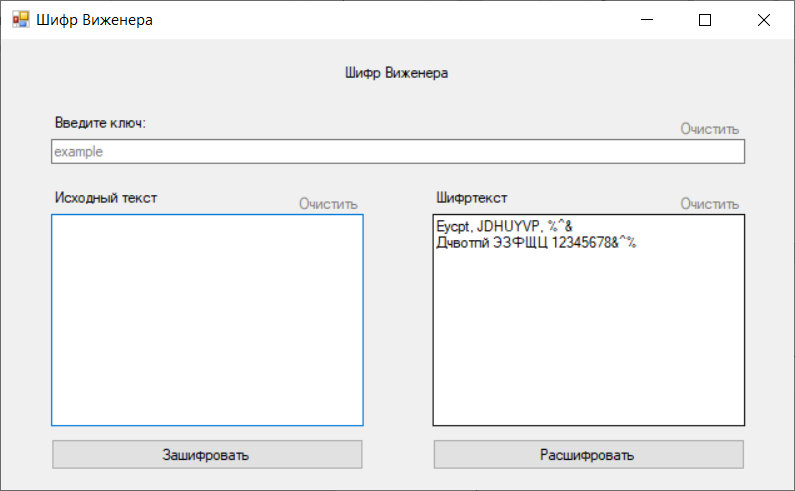


Рисунок 5. Очистка поля Исходный текст.

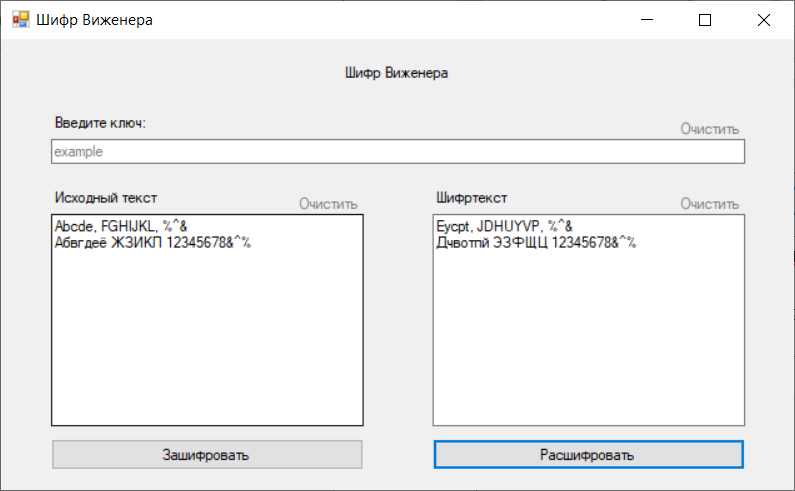


Рисунок 6. Исходный текст (результат расшифрования).

**Вывод:**

В ходе выполнения лабораторной работы я ознакомилась с алгоритмом шифра Вижинера, реализовала на практике программу по зашифрованию и расшифрованию данным шифром на языке C#. Данные навыки я могу применить при реализации других шифров и при дальнейшем использовании шифра Вижинера.