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 №8**

|  |
| --- |
| **Шифр Гронсфельда** |
| *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 Гронсфельда на любом языке программирования.

**Теория:**

Ключ

Длина ключа (*K*) должна быть равной длине исходного текста. Для этого циклически записывают ключ до тех пор, пока его длина не будет соответствовать длине исходного текста.

### Шифрование

Каждый символ *Mi* открытого текста *M* нужно на *Ki* (соответствующий символ ключа *K*) шагов сдвинуть вправо. Или пользуясь таблицей Гронсфельда (*Tx y*, где *x* — номер строки, а *y* — номер столбца и отсчет ведется с нуля): каждый символ *Ci* шифротекста *C* находится на пересечении столбца *y*, первый (заголовочный) символ которого равен соответствующему символу открытого текста *Mi*, и *Ki*-й (соответствующей цифры ключа) строки — (*TKi y*)

### Дешифрирование

Каждый символ (*Ci*) зашифрованого текста *C* нужно на *Ki* (соответствующий символ ключа *K*) шагов сдвинуть влево.Или пользуясь таблицей Гронсфельда (*Tx y*, где *x* — номер строки, а *y* — номер столбца и отсчет ведется с нуля): нужно в *Ki* (*i*-ая цифра ключа *K*) строке найти символ, который равен соответствующему символу шифротекста (*TKi y = Ci*), и первый (заголовочный) элемент столбца будет *i*-ый символ открытого текста.

**Исходный код:**

**Алфавит:**

string ALF\_RU = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

string alf\_ru = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

string ALF\_EN = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

string alf\_en = "abcdefghijklmnopqrstuvwxyz";

**Алгоритм зашифрования:**

private void f8\_ButtonEncrypt\_Click(object sender, EventArgs e)

{

string Source = f8\_fieldOriginal.Text.Replace("\r\n", "\r");

char[] message = Source.ToCharArray();

var entered\_key = Regex.Replace(f8\_fieldKey.Text, "[^0-9]", "");

f8\_fieldKey.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++)

{

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

{

message[i] = ALF\_RU[(ALF\_RU.IndexOf(message[i]) + (entered\_key[keyPos] - '0')) % ALF\_RU.Length];

}

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

{

message[i] = alf\_ru[(alf\_ru.IndexOf(message[i]) + (entered\_key[keyPos] - '0')) % alf\_ru.Length];

}

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

{

message[i] = ALF\_EN[(ALF\_EN.IndexOf(message[i]) + (entered\_key[keyPos] - '0')) % ALF\_EN.Length];

}

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

{

message[i] = alf\_en[(alf\_en.IndexOf(message[i]) + (entered\_key[keyPos] - '0')) % alf\_en.Length];

}

keyPos++;

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

}

string result = new string(message);

f8\_fieldCiphertext.Text = result.Replace("\r", "\r\n"); ;

metka\_exit:;

}

**Алгоритм расшифрования:**

private void f8\_ButtonDecrypt\_Click(object sender, EventArgs e)

{

string Source = f8\_fieldCiphertext.Text.Replace("\r\n", "\r");

char[] message = Source.ToCharArray();

var entered\_key = Regex.Replace(f8\_fieldKey.Text, "[^0-9]", "");

f8\_fieldKey.Text = entered\_key;

if (entered\_key.Length < 1)

{

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

goto metka\_exit;

}

int keyPos = 0;

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

{

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

{

message[i] = ALF\_RU[(ALF\_RU.IndexOf(message[i]) - (int)(entered\_key[keyPos] - '0') + ALF\_RU.Length) % ALF\_RU.Length];

}

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

{

message[i] = alf\_ru[(alf\_ru.IndexOf(message[i]) - (int)(entered\_key[keyPos] - '0') + alf\_ru.Length) % alf\_ru.Length];

}

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

{

message[i] = ALF\_EN[(ALF\_EN.IndexOf(message[i]) - (int)(entered\_key[keyPos] - '0') + ALF\_EN.Length) % ALF\_EN.Length];

}

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

{

message[i] = alf\_en[(alf\_en.IndexOf(message[i]) - (int)(entered\_key[keyPos] - '0') + alf\_en.Length) % alf\_en.Length];

}

keyPos++;

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

}

string result = new string(message);

f8\_fieldOriginal.Text = result.Replace("\r", "\r\n"); ;

metka\_exit:;

}

**Результат работы программы:**

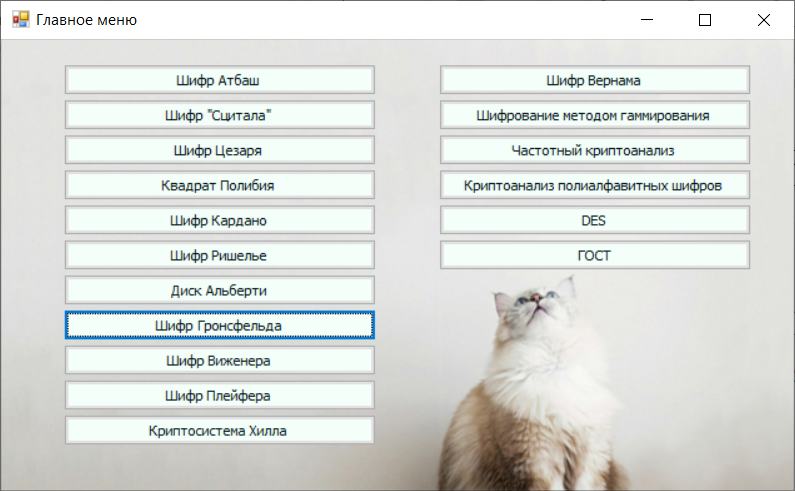


Рисунок 1. Главное меню программы.

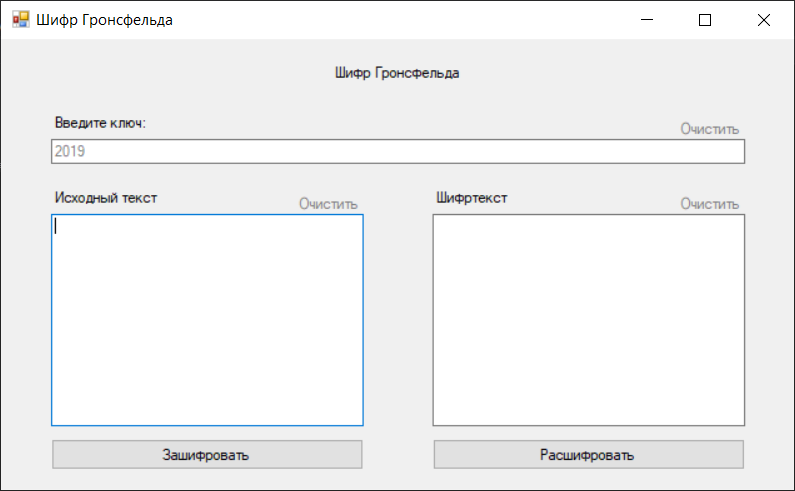


Рисунок 2. Окно шифра Гронсфельда.

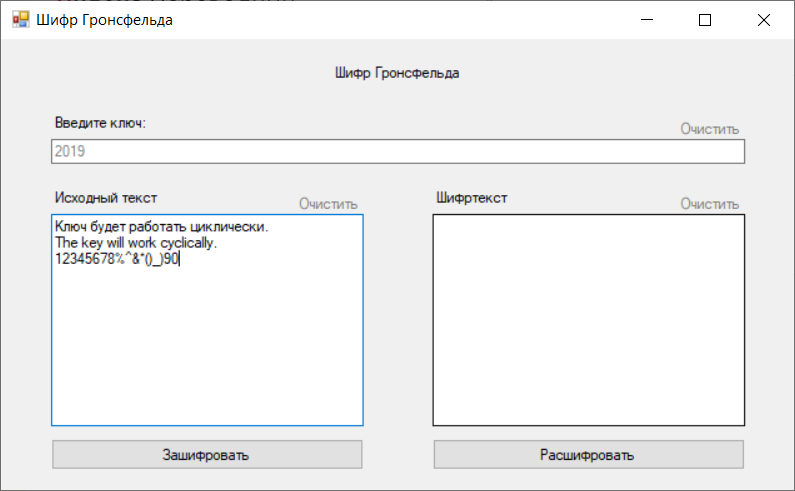


Рисунок 3. Исходный текст.

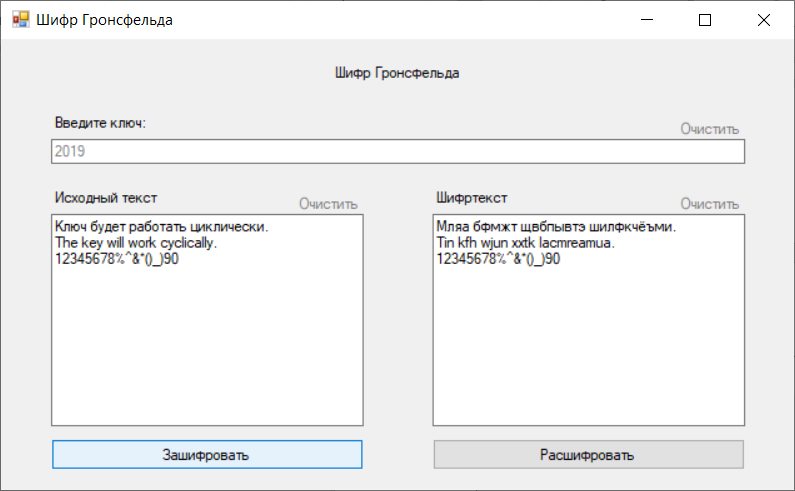


Рисунок 4. Шифртекст (результат зашифрования).

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

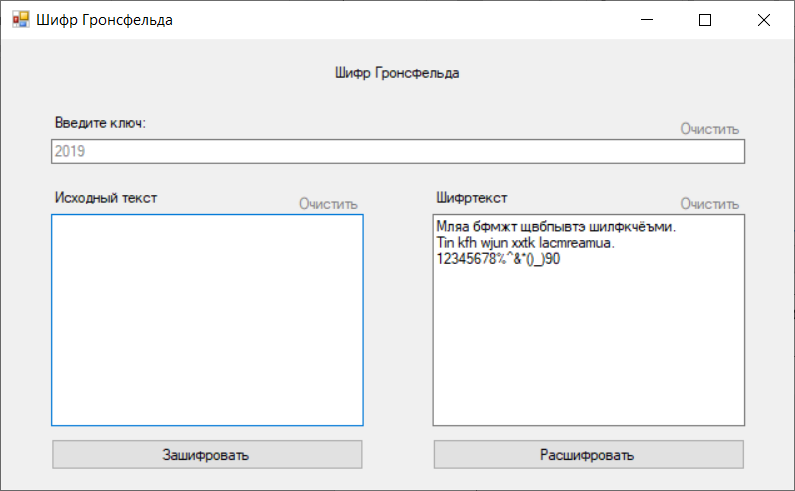


Рисунок 5. Очистка поля Исходный текст.

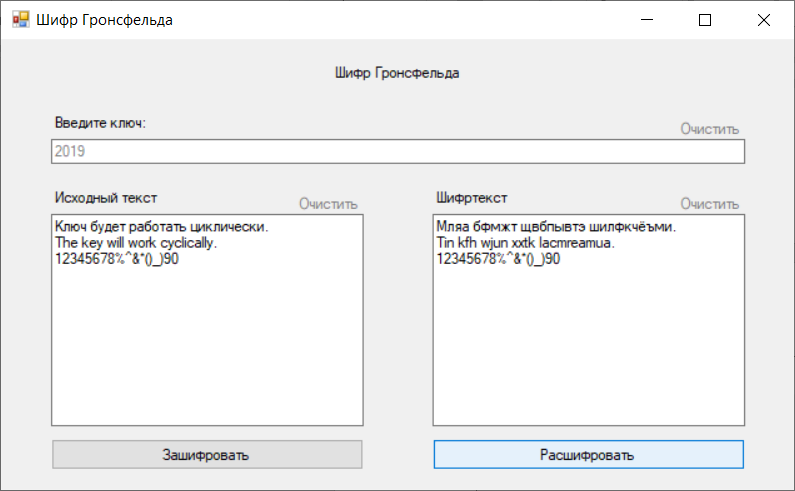


Рисунок 6. Исходный текст (результат расшифрования).

**Вывод:**

В ходе выполнения лабораторной работы я ознакомилась с алгоритмом шифра Гронсфельда, 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 Гронсфельда.