

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Лабораторная работа №5

по дисциплине: «Методы и средства криптографической защиты информации»

На тему «Программная реализация простых шифров»

Выполнил обучающийся гр. ВКБ42

Михайлов Александр Сергеевич

          Проверила:

Сафарьян О.А

Ростов-на-Дону

2024

Цель работы: формирование умений шифрования с использованием алгоритма шифрования цифр.

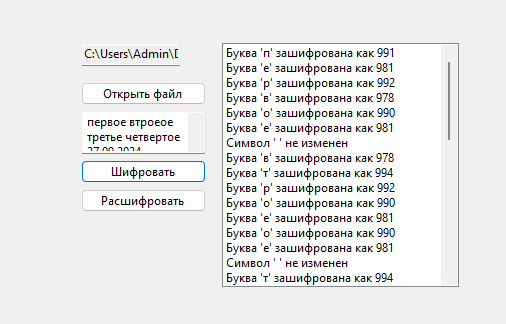
Задание:

Зашифруйте сообщение, используя шифр цифр.

Сообщение: сообщение

Ключ: любой файл

Программная реализация:



Вывод:

В ходе выполнения лабораторной работы были сформированы умения шифрования с использованием простых шифров.

Листинг кода

using System.Text;

namespace ШифровкаЦифр

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Text files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

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

{

string filePath = openFileDialog.FileName;

textBoxFilePath.Text = filePath;

string fileContent = File.ReadAllText(filePath);

textBoxContent.Text = fileContent;

}

}

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

{

{

string content = textBoxContent.Text;

string encryptedContent = EncryptContent(content);

// Получаем путь к исходному файлу

string originalFilePath = textBoxFilePath.Text;

string directory = Path.GetDirectoryName(originalFilePath);

string fileName = Path.GetFileNameWithoutExtension(originalFilePath);

string fileExtension = Path.GetExtension(originalFilePath);

// Создаем новое имя файла с добавлением "+1"

string newFileName = $"{fileName}+1{fileExtension}";

string newFilePath = Path.Combine(directory, newFileName);

// Записываем зашифрованный текст в новый файл

File.WriteAllText(newFilePath, encryptedContent);

MessageBox.Show("Файл успешно зашифрован и сохранен.");

}

}

private string EncryptContent(string content)

{

StringBuilder encryptedContent = new StringBuilder();

listBoxSteps.Items.Clear();

foreach (char c in content)

{

if (char.IsLetter(c))

{

char baseChar = char.IsUpper(c) ? 'A' : 'a';

int encryptedChar = c - baseChar + 1;

encryptedContent.Append(encryptedChar.ToString() + " ");

listBoxSteps.Items.Add($"Буква '{c}' зашифрована как {encryptedChar}");

}

else

{

encryptedContent.Append(c);

listBoxSteps.Items.Add($"Символ '{c}' не изменен");

}

}

return encryptedContent.ToString();

}

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Text files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

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

{

string filePath = openFileDialog.FileName;

string fileContent = File.ReadAllText(filePath);

string decryptedContent = DecryptContent(fileContent);

// Получаем путь к исходному файлу

string directory = Path.GetDirectoryName(filePath);

string fileName = Path.GetFileNameWithoutExtension(filePath);

string fileExtension = Path.GetExtension(filePath);

// Создаем новое имя файла с добавлением "+2"

string newFileName = $"{fileName}+2{fileExtension}";

string newFilePath = Path.Combine(directory, newFileName);

// Записываем расшифрованный текст в новый файл

File.WriteAllText(newFilePath, decryptedContent);

MessageBox.Show("Файл успешно расшифрован и сохранен.");

}

}

private string DecryptContent(string content)

{

StringBuilder decryptedContent = new StringBuilder();

listBoxSteps.Items.Clear();

string[] parts = content.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string part in parts)

{

if (int.TryParse(part, out int number))

{

char decryptedChar = (char)(number + 'A' - 1);

decryptedContent.Append(decryptedChar);

listBoxSteps.Items.Add($"Число {number} расшифровано как '{decryptedChar}'");

}

else

{

decryptedContent.Append(part);

listBoxSteps.Items.Add($"Символ '{part}' не изменен");

}

}

return decryptedContent.ToString();

}

}

}