Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Лабораторная работа № 6**

«ИЗУЧЕНИЕ УСТРОЙСТВА И ФУНКЦИОНАЛЬНЫХ ОСОБЕННОСТЕЙ ШИФРОВАЛЬНОЙ МАШИНЫ «ЭНИГМА»»

Выполнил:

Студент: Герман А.Е.

ФИТ 3 курса 5 группы

Преподаватель: Савельева М.Г

Минск 2024

# **1 Описание приложения**

Данное консольное приложение, разработанное на языке программирования C#, предназначено для шифрования текста по с помощью одной из реализаций шифровальной машины «Энигма».

Основные функциональности приложения:

* Шифрование текста на языке алфавита;
* Смена «отступа» в роторах в процессе работы программы.

# **2 Методика выполнения расчетов**

В данной лабораторной работе была поставлена цель создания приложения, имитирующее работу машины «Энигма». На листингах 2.1 и 2.2 представлен классы, реализующий данную функциональность.

|  |
| --- |
| class Rotor  {  public char[] Alphabet { get; set; }  public int Shift { get; set; }  public int countOfShifts { get; set; }  public int DoShift(int turnsCount)  {  int \_turnsCount = 0;ф  if (countOfShifts < Alphabet.Length)  {  countOfShifts += Shift;  }  else  {  countOfShifts = Alphabet.Length - countOfShifts;  \_turnsCount++;  }  for (int j = 0; j < Shift + turnsCount; j++)  {  char temp = ' ';  for (int i = 0; i < Alphabet.Length - 1; i++)  {  if (i == 0)  {  temp = Alphabet[Alphabet.Length - 1];  Alphabet[Alphabet.Length - 1] = Alphabet[i];  Alphabet[i] = Alphabet[i + 1];  }  else  {  Alphabet[i] = Alphabet[i + 1];  }  if (i == Alphabet.Length - 2)  {  Alphabet[Alphabet.Length - 2] = temp;  }  }  }  return \_turnsCount;  }  public void PickStartPosition(string startPositionAlpha)  {  char chr = Char.Parse(startPositionAlpha.Replace("System.Windows.Controls.ListBoxItem:", "").Trim());  int oldIndex = Array.IndexOf(Alphabet, chr);  for (int j = 0; j < oldIndex; j++)  {  char temp = ' ';  for (int i = 0; i < Alphabet.Length - 1; i++)  {  if (i == 0)  {  temp = Alphabet[Alphabet.Length - 1];  Alphabet[Alphabet.Length - 1] = Alphabet[i];  Alphabet[i] = Alphabet[i + 1];  }  else  {  Alphabet[i] = Alphabet[i + 1];  }  if (i == Alphabet.Length - 2)  {  Alphabet[Alphabet.Length - 2] = temp;  }  }  }  } |

Листинг 2.1 – Класс **Rotor**, реализующий функционал ротора в машине «Энигма»

|  |
| --- |
| private void Encrypt\_Click(object sender, RoutedEventArgs e)  {  string originalMessage = Orig\_message\_input.Text.ToLower().Replace(" ","");  char encryptedChar;  string encryptedMessage = "";  L.PickStartPosition(Start\_position\_Rot\_1.Items[Start\_position\_Rot\_1.SelectedIndex].ToString());  M.PickStartPosition(Start\_position\_Rot\_2.Items[Start\_position\_Rot\_2.SelectedIndex].ToString());  R.PickStartPosition(Start\_position\_Rot\_3.Items[Start\_position\_Rot\_3.SelectedIndex].ToString());  for (int i = 0; i < originalMessage.Length; i++)  {  encryptedChar = R.Alphabet[originalAlphabet.IndexOf(originalMessage[i])];  encryptedChar = M.Alphabet[originalAlphabet.IndexOf(encryptedChar)];  encryptedChar = L.Alphabet[originalAlphabet.IndexOf(encryptedChar)];  char encrypt\_symbolFromRefl;  if (!reflector.Alphabet.TryGetValue(encryptedChar, out encrypt\_symbolFromRefl))  encryptedChar = reflector.Alphabet.First(key => key.Value == encryptedChar).Key;  else  encryptedChar = encrypt\_symbolFromRefl;  encryptedChar = L.Alphabet[originalAlphabet.IndexOf(encryptedChar)];  encryptedChar = M.Alphabet[originalAlphabet.IndexOf(encryptedChar)];  encryptedMessage = encryptedMessage + R.Alphabet[originalAlphabet.IndexOf(encryptedChar)];  L.DoShift(M.DoShift(R.DoShift(0)));  }  Encrypt\_Output.Text = encryptedMessage;  }  } |

Листинг 2.2 – Функция **Encrypt\_Click**, реализующий функционал для шифрования с помощью машины «Энигма»

# **3 Результаты работы приложения**

Для выполнения расчетов достаточно необходимо запустить приложение. Рисунки 3.1 - 3.5 показывают необходимые расчеты и вызовы методов, требуемые в данной лабораторной работе.

## **3.1 Шифрование в «Энигма»**

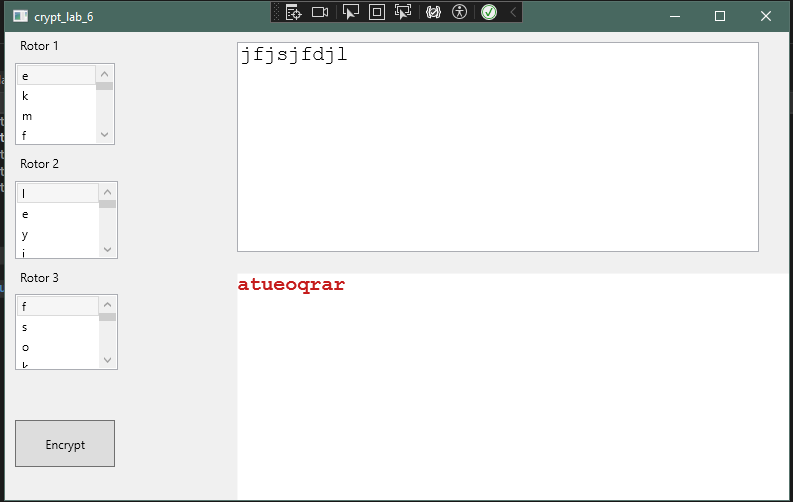
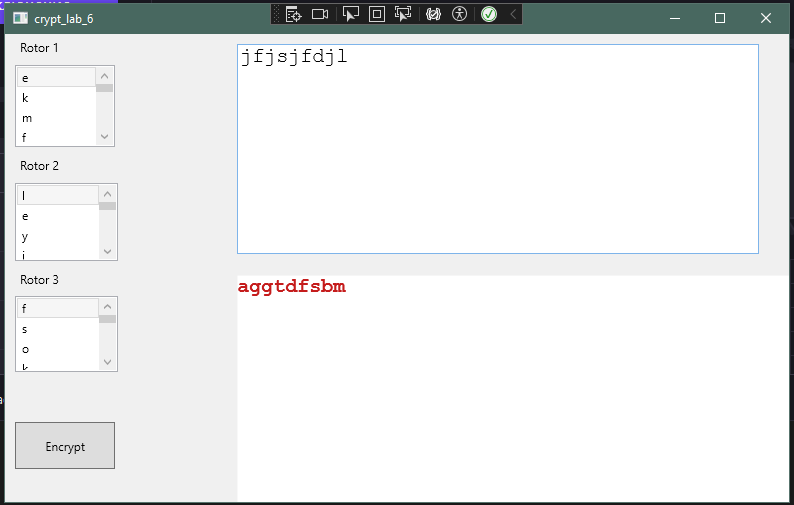
 

Рисунок 3.1 – Результат работы методов класса **Encrypt\_Click** с входным текстом (два раза, для демонстрации работы «Энигмы»).

Спецификация взята с ресурса: https://www.codesandciphers.org.uk/enigma/rotorspec.htm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вариант задания | L | M | R | Re | Li-Mi-Ri |
| 5 | I | Beta | Gamma | B | 3-1-3 |

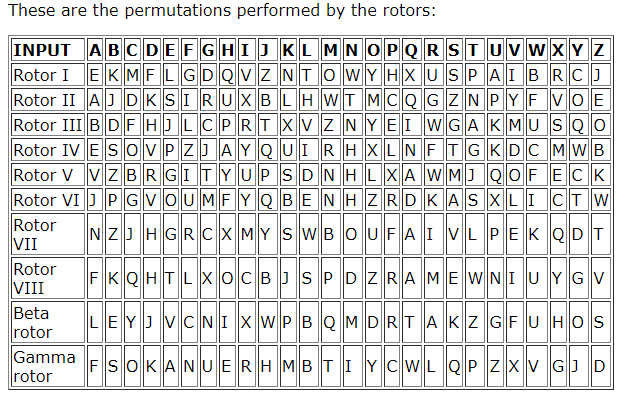


Рисунок 3.2 – Cпецификация для роторов

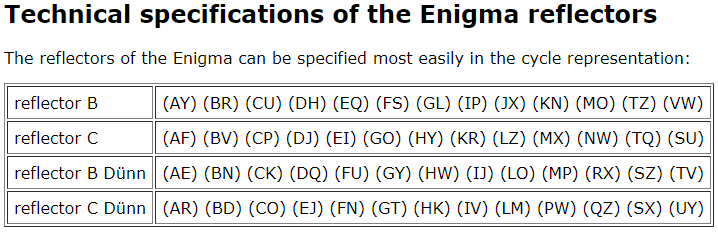


Рисунок 3.3 – Cпецификация для рефлекторов

## **3.2 Самостоятельный выбор отступа для шифрования**

## 

Рисунок 3.4 – Результат работы методов класса **Encrypt\_Click** с входным текстом

Слева есть отображения алфавита в роторах, путём нажатия на букву, мы ставим её первой и таким образом выполняем отступ. Пример на рисунке 3.4, где в 3 роторе была выбрана буква “s”, а во втором «j», что сильно изменило шифротекст в отличие от двух вариантов 3.1. Первый ротор остался без изменений.

## **3.3 Подсчёт сложности шифрования**

Рисунок 3.5 –Вычисления вариативности шифрования символа в машине «Энигма»

# **4 Вывод**

В ходе изучения теоретических материалов лабораторной работы и выполнения её практической части были изучены основы теории построения, а также реализации шифровальной машины Энигма. Было установлено, что криптостойкость машины составляет 1,07\*1023 комбинаций.