



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Студент Романов Семен Константинович

Группа ИУ7-75Б

Предмет Защита информации

Студент

подпись, дата

Романов С. К.

фамилия, и.о.

Преподаватель

подпись, дата

Чиж И. С.

фамилия, и.о.

2023 г.

ВВЕДЕНИЕ

Цель лабораторной работы — разработать программу шифровальной машины «Энигма» [1].

Задачи лабораторной работы:

- 1) провести анализ работы шифровальной машина «Энигма»;
- 2) описать алгоритм шифрования;
- 3) релизовать описанный алгоритм.

1 Аналитическая часть

Шифровальная машина «Энигма» состоит из трех основных частей:

- 1) роторы — диски обладающие 26 гранями, где каждая грань представляла собой нумерацию английского алфавита;
- 2) рефлектор — статический механизм, позволяющий машине также расшифровать текст;
- 3) коммутатор — набор парных шифров.

1.1 Алгоритм работы машины

На вход «Энигме» подается строка, которая разбивается на символы. Далее символ проходит через коммутационную панель, который меняет символ в соответствии с настройкой. После прохождения панели, символ проходит через три диска и попадает на рефлектор. После работы рефлектора, символ отправляется обратно на диск и окончательно шифруется через коммутатор. Затем один ротор совершает оборот, если ротор обернулся 26 раз, то поворачивается следующий.

2 Конструкторская часть

2.1 Разработка алгоритма

На рисунке 1 приведена схема работы шифровальной машины Энигма.

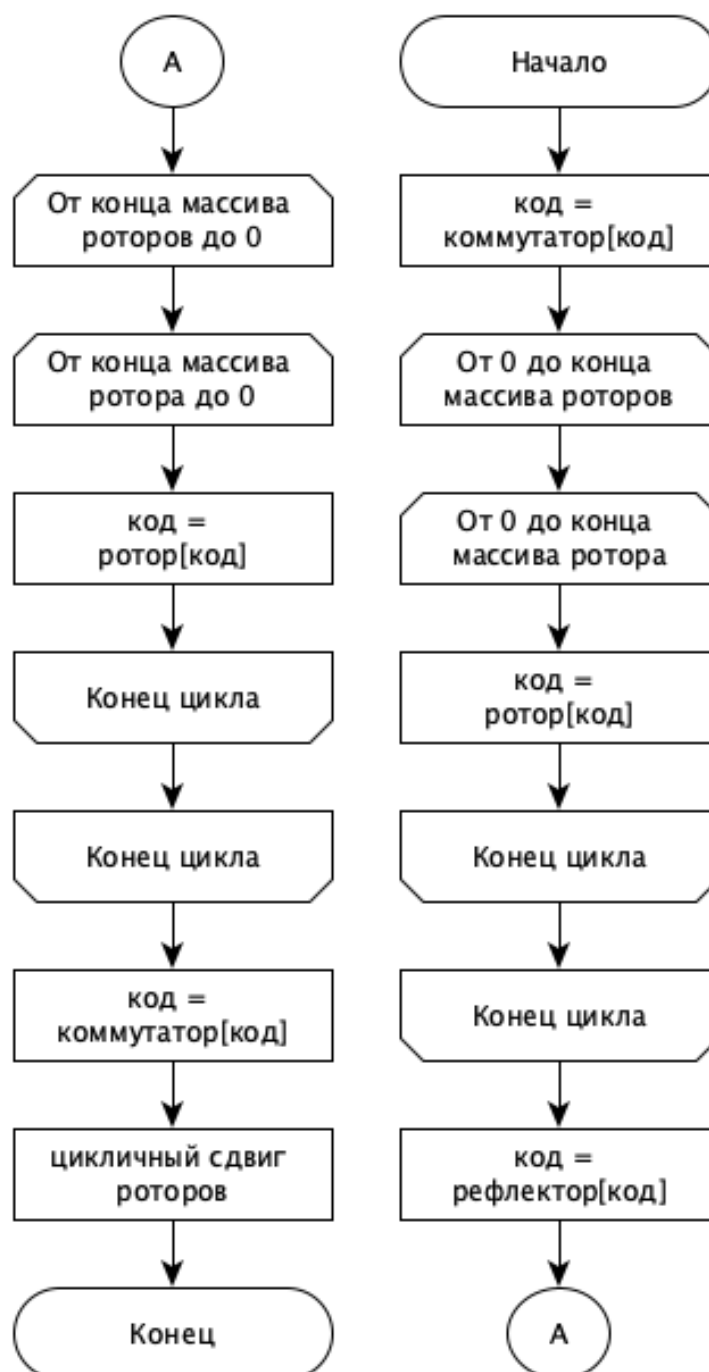


Рисунок 1 – Схема работы шифровальной машина Энигма

3 Технологическая часть

3.1 Средства реализации

Для реализации ПО был выбран язык C++ [2]. В данном языке есть все требующиеся инструменты для данной лабораторной работы. В качестве среды разработки была выбрана среда NeoVim[3].

3.2 Реализация алгоритма

Листинг 1 – Реализация алгоритма Энигмы.

```
1 uint8_t Enigma::encrypt(Encoder& encoder, uint8_t code)
2 {
3     code = encoder.encode(code);
4     uint64_t rotor_queue = 1;
5
6     if(code > size_rotor)
7     {
8         throw std::out_of_range("Code bigger than size of rotor");
9     }
10    code = commutator.inner[code];
11
12    for(auto& rotor : rotors)
13    {
14        code = rotor.inner[code];
15    }
16    code = reflector.inner[code];
17
18    for(int i = num_rotors - 1; i >= 0; --i)
19    {
20        try
21        {
22            code = find_rotor(i, code);
23        }
24        catch(const std::overflow_error& e)
25        {
26            std::cout << e.what() << std::endl;
27        }
28    }
```

Листинг 2 – Реализация алгоритма Энигмы, продолжение.

```
1
2     counter++;
3     for(int i = 0; i < num_rotors; ++i)
4     {
5         if(counter % rotor_queue == 0)
6         {
7             rotor_shift(i);
8         }
9         rotor_queue *= size_rotor;
10    }
11    code = commutator.inner[code];
12
13    return code;
14 }
15
16 std::vector<uint8_t> Enigma::encrypt(Encoder& encoder, std::vector<uint8_t>&
    codes)
17 {
18     std::vector<uint8_t> res;
19     for(auto& symbol : codes)
20     {
21         try
22         {
23             uint8_t encoded_ch = encrypt(encoder, symbol);
24             uint8_t decoded_ch = encoder.decode(encoded_ch);
25             res.push_back(decoded_ch);
26         }
27         catch(std::overflow_error& e)
28         {
29             std::cout << e.what() << std::endl;
30         }
31     }
32
33     return res;
34 }
```

3.3 Тестовые данные

В таблице 1 приведены тесты, описанные в листинге 4 для алгоритма шифрования Энигмы. Применена методология черного ящика. Тесты пройдены *успешно*.

Листинг 3 – Реализация функциональных тестов.

```
1 TEST(Enigma, Encode)
2 {
3     Encoder encoder = setup_encoder();
4     Enigma enigma = setup_enigma();
5     std::string new_message{"test"};
6     std::vector<uint8_t> message(new_message.begin(), new_message.end());
7
8     ASSERT_FALSE(enigma.encrypt(encoder, message) == message);
9 }
10
11 TEST(Enigma, Decode)
12 {
13     Encoder encoder = setup_encoder();
14     Enigma enigma_encrypt = setup_enigma();
15     Enigma enigma_decrypt = setup_enigma();
16     std::string new_message{"test"};
17     std::vector<uint8_t> message(new_message.begin(), new_message.end());
18     std::vector<uint8_t> enc_msg = enigma_encrypt.encrypt(encoder, message);
19
20     ASSERT_TRUE(enigma_decrypt.encrypt(encoder, enc_msg) == message);
21 }
22
23 TEST(Enigma, Encode_Empty)
24 {
25     Encoder encoder = setup_encoder();
26     Enigma enigma = setup_enigma();
27     std::string new_message{""};
28     std::vector<uint8_t> message(new_message.begin(), new_message.end());
29
30     ASSERT_TRUE(enigma.encrypt(encoder, message) == message);
31 }
```

Листинг 4 – Реализация функциональных тестов, продолжение.

```

1 TEST(Enigma, Encode_One)
2 {
3     Encoder encoder = setup_encoder();
4     Enigma enigma = setup_enigma();
5     std::string new_message{"test"};
6     std::vector<uint8_t> message(new_message.begin(), new_message.end());
7
8     ASSERT_FALSE(enigma.encrypt(encoder, message) == message);
9 }
10
11 TEST(Enigma, Decode_One)
12 {
13     Encoder encoder = setup_encoder();
14     Enigma enigma_encrypt = setup_enigma();
15     Enigma enigma_decrypt = setup_enigma();
16     std::string new_message{"F"};
17     std::vector<uint8_t> message(new_message.begin(), new_message.end());
18     std::vector<uint8_t> enc_msg = enigma_encrypt.encrypt(encoder, message);
19
20     ASSERT_TRUE(enigma_decrypt.encrypt(encoder, enc_msg) == message);
21 }

```

Таблица 1 – Функциональные тесты

Входная строка	Выходная строка
<i>TEST</i>	<i>FGxKNR</i>
<i>FGxKNR</i>	<i>TEST</i>
<i>F</i>	<i>R</i>
<i>R</i>	<i>F</i>

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе:

- 1) проведен анализ работы шифровальной машина «Энигма»;
- 2) описан алгоритм шифрования;
- 3) реализован описанный алгоритм;

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Список литературы

1. Enigma german code device. <https://www.britannica.com/topic/Enigma-German-code-device>. дата обращения: 17.09.2023.
2. Язык программирования C++. <https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference?view=msvc-170>. дата обращения: 17.09.2023.
3. Neovim. <https://neovim.io/>. дата обращения: 17.09.2023.