Федеральное государственное бюджетное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ»**

*Институт ЭУИС*

*Кафедра ИСТАС*

**Практическая работа №2**

по дисциплине:

«Защита информации»

**Вариант: Сложение по модулю 2.**

|  |
| --- |
| Сдал: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |
| Торговцев В.П. ИЭУИС 4-2 |
| Принял: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |
| Алексеевская Я.А. |

г. Москва

2020 г.

**Описание программы.**

На рисунке 1 представлен вид главного окна программы.

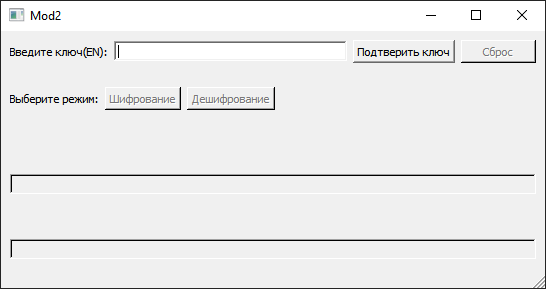


Рисунок 1 - Главное окно

Сверху расположено поле ввода для Ключа, клавиша подтверждения ключа и кнопка сброса ключа.

Ниже расположен блок выбора режима работы программы в виде двух кнопок – «Шифрование» и «Дешифрование».

Под Ними расположены 2 поля ввода, которые становятся активными в зависимости от выбранного режима работы программы, верхнее поле ввода предназначено для незашифрованного текста, нижнее – для зашифрованного.

Программа работает только с латиницей и цифрами, для этого установлено ограничение на ввод.

Некоторые элементы программы активны не сразу, но они становятся доступны после совершения некоторых подготовительных действий.

**Ссылка на скачивание исходных файлов и программы:** <https://drive.google.com/open?id=1FVFLD3BrP4dp22OXAa0GYPPbpYJVZmk_>

**Инструкция.**

Запуск:

Вариант 1. Открыть файл Torgovtsev\_Practice\_2.exe в папке «ProjectForWindows».

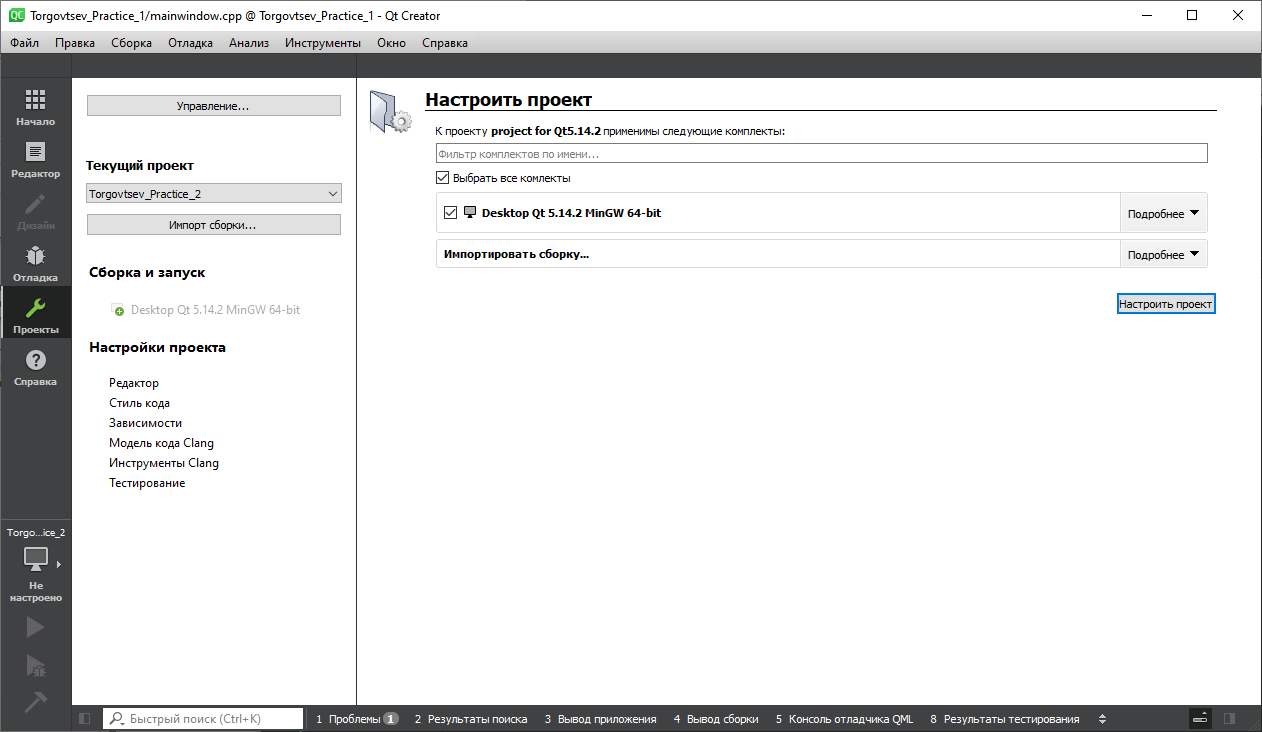
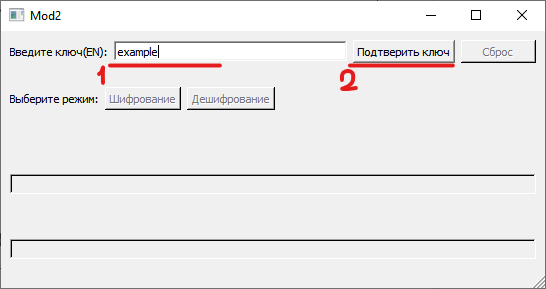
Вариант 2. Если установлен Qt, запустить Torgovtsev\_Practice\_2.pro в папке «project for Qt5.14.2», нажать кнопку «Настроить проект» и запустить проект комбинацией клавиш Ctrl+R.

Рисунок 2 – Запуск программы

Ход программы:

1. Введите ключ в поле сверху, затем нажмите на кнопку «Подтвердить ключ». Рисунок 3.

нажатии на кнопку «Подтвердить ключ» без введенного ключа вы получите сообщение – «Введите Ключ!» Рисунок 4.

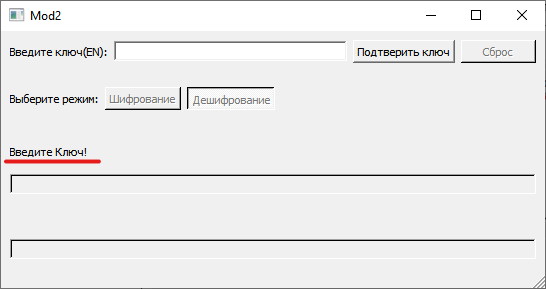
Рисунк 3 – Ввод ключа

Рисунок 4 – Ошибка при вводе ключа

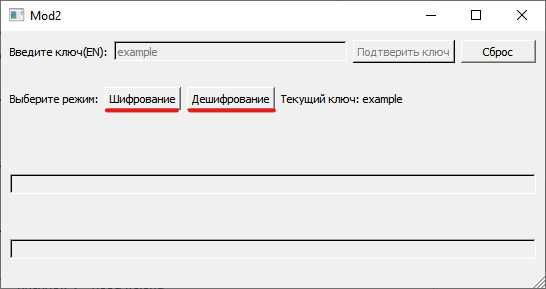
1. Ниже, под полем ввода ключа, станет активен выбор режима программы, выберите один из двух режимов – «Шифрование» и «Дешифрование». Рисунок 5.

Рисунок 5 – Выбор режима работы программы

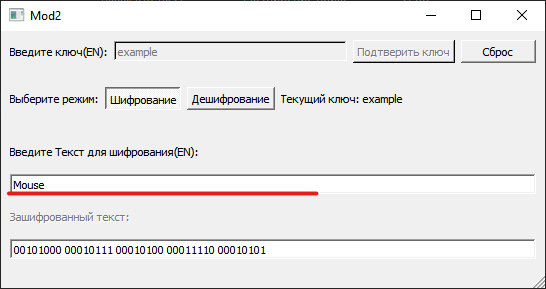
1. В зависимости от того, какой режим работы программы вы выбрали, переходите к одному из двух представленных ниже пунктов – 3.1 – Шифрование, 3.2 – Дешифрование
   1. После выбора режима Шифрования вам станет активно среднее поле ввода. Для того, чтобы зашифровать сообщение просто введите его в активное поле, и вы увидите результат в поле под надписью: «Зашифрованный текст:». Рисунок 6.

Рисунок 6 – Процесс шифрования

* 1. После выбора режима Дешифрования вам станет активно нижнее поле ввода. Для того чтобы расшифровать сообщение просто начните вводить двоичный код(с пробелом через каждый байт) в активное нижнее поле, и вы увидите Расшифрованное сообщение в поле под надписью – «Исходный текст:». Рисунок 7.

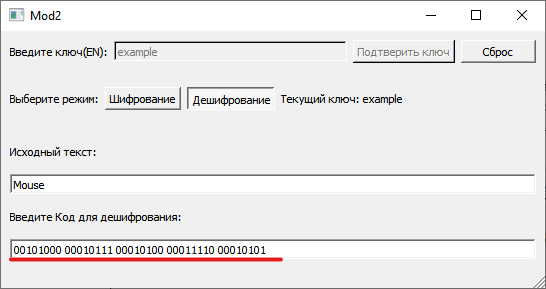


Рисунок 7 – Процесс дешифрования

1. Для того, чтобы ввести новый ключ и обнулить все шифрования нажмите на кнопку «Сброс». Рисунок 8.

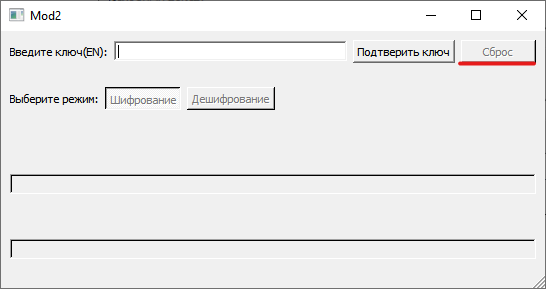


Рисунок 8 – Результат сброса

Исходный код программы:

**Файл “mainwindow.h”:**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

int keyLength = 0;

int \*\*letterSequence;

private slots:

void **on\_pushButtonEncrypt\_clicked**();

void **on\_lineEdit\_Encrypt\_textEdited**(const QString &arg1);

void **on\_pushButtonInputKey\_clicked**();

void **on\_pushButton\_Decrypt\_clicked**();

void **on\_lineEdit\_Decrypt\_textEdited**(const QString &arg1);

void **on\_pushButton\_ResetKey\_clicked**();

private:

Ui::MainWindow \*ui;

QString key = "";

};

#endif // MAINWINDOW\_H

**Файл “main.cpp”:**

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(*argc*, argv);

MainWindow w;

w.show();

return a.exec();

}

**Файл “mainwindow.cpp”:**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "QDebug"

#include <QRegExpValidator>

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

QRegExp rx("[A-z 0-9]{1,99}");//----Создание ограничения на ввод

QValidator \*validator = new QRegExpValidator(rx, this);

QRegExp rx2("[0-1 ]{1,99}");

QValidator \*validator2 = new QRegExpValidator(rx2, this);

ui->lineEditKey->setValidator(validator);

ui->lineEdit\_Decrypt->setValidator(validator2);

ui->lineEdit\_Encrypt->setValidator(validator);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButtonEncrypt\_clicked**()//---нажатие на кнопку выбора шифрования(шифрование)

{

ui->pushButtonEncrypt->setChecked(1);

ui->pushButton\_Decrypt->setChecked(0);

ui->lineEdit\_Encrypt->setEnabled(1);

ui->label\_Encrypt->setEnabled(1);

ui->label\_Encrypt->setText("Введите Текст для шифрования(EN):");

ui->lineEdit\_Decrypt->setEnabled(0);

ui->label\_Decrypt->setText("Зашифрованный текст:");

}

void MainWindow::**on\_pushButton\_Decrypt\_clicked**()//---нажатие на кнопку выбора шифрования(дешифрование)

{

ui->pushButton\_Decrypt->setChecked(1);

ui->pushButtonEncrypt->setChecked(0);

ui->lineEdit\_Decrypt->setEnabled(1);

ui->label\_Decrypt->setEnabled(1);

ui->label\_Decrypt->setText("Введите Код для дешифрования:");

ui->lineEdit\_Encrypt->setEnabled(0);

ui->label\_Encrypt->setText("Исходный текст:");

}

void MainWindow::**on\_pushButtonInputKey\_clicked**()//-----нажатие на кнопку принятия ключа

{

if(ui->lineEditKey->text().length() == 0){ // Проверка на наличие ключа

ui->label\_Encrypt->setEnabled(1);

ui->label\_Encrypt->setText("Введите Ключ!");

}

else{

key = ui->lineEditKey->text();

ui->lineEditKey->setEnabled(0);

ui->pushButtonInputKey->setEnabled(0);

ui->pushButton\_ResetKey->setEnabled(1);

ui->label\_KeyReview->setText("Текущий ключ: " + key);

ui->pushButtonEncrypt->setEnabled(1);

ui->pushButton\_Decrypt->setEnabled(1);

}

}

void MainWindow::**on\_lineEdit\_Encrypt\_textEdited**(const QString &arg1)//---реакция на ввод текста в верхнее поле(шифрование)

{

QString out\_text;

int pos\_key = 0;

int key\_num, input\_num, out\_num;

for (int i = 0; i < arg1.length(); i++) {

key\_num = key[pos\_key++].unicode(); // Номер символа ключа в кодировке unicode.

input\_num = arg1[i].unicode(); // Номер символа исходного текста в кодировке unicode.

out\_num = key\_num ^ input\_num;// XOR номеров символов.

out\_text+=QString("%1").arg(out\_num, 8, 2, QChar('0'));

// out\_text+=QChar(out\_num); // Преобразование кода символа в символ.

out\_text+=" ";

if(pos\_key == key.length())

pos\_key = 0;

}

ui->lineEdit\_Decrypt->clear();

ui->lineEdit\_Decrypt->setText(out\_text);

ui->lineEdit\_Decrypt->setEnabled(1);

}

void MainWindow::**on\_lineEdit\_Decrypt\_textEdited**(const QString &arg1)//---реакция на ввод текста в нижнее поле(дешифрование)

{

QString Input = ui->lineEdit\_Decrypt->text();

if(Input[Input.length()-1]!= ' ') Input[Input.length()] = ' ';

QString out\_text, texet, text, p;

bool ok;

int pos\_key = 0;

int key\_num, input\_num, out\_num;

for (int i = 0; i < Input.length(); i++) {

if (Input[i] != " " && i!=Input.length()-1) texet+=Input[i];

else

{

key\_num = key[pos\_key++].unicode();

text=texet.toInt(&ok,2);

p = text;

input\_num = p[0].unicode();

out\_num = key\_num ^ input\_num;

out\_text+=QChar(out\_num);

if(pos\_key == key.length())

pos\_key = 0;

texet = "";

}

}

ui->lineEdit\_Encrypt->clear();

ui->lineEdit\_Encrypt->setText(out\_text);

ui->lineEdit\_Encrypt->setEnabled(1);

}

void MainWindow::**on\_pushButton\_ResetKey\_clicked**()

{

ui->lineEditKey->setText("");

ui->lineEdit\_Decrypt->setText("");

ui->lineEdit\_Decrypt->setEnabled(0);

ui->label\_Decrypt->setText("");

ui->lineEdit\_Encrypt->setText("");

ui->lineEdit\_Encrypt->setEnabled(0);

ui->label\_Encrypt->setText("");

ui->lineEditKey->setEnabled(1);

ui->pushButtonInputKey->setEnabled(1);

ui->label\_KeyReview->setText("");

ui->pushButtonEncrypt->setEnabled(0);

ui->pushButton\_Decrypt->setEnabled(0);

ui->pushButton\_ResetKey->setEnabled(0);

}