Министерство образования Российской Федерации

Пензенский Государственный Университет

Кафедра Вычислительная техника

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по курсу «Программирование»

на тему «Стеганография»

Выполнил:

студент группы 21ВВ1

Лобанов Рамиль

Принял:

Токарев А.Н

ОТЛ

7.06.22

Пенза 2022

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники

Кафедра “Вычислительная техника”

“УТВЕРЖДАЮ”

Зав. кафедрой ВТ

_____ М.А. Митрохин

«____»_____20__г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Лобанов Р.И. Группа 21ВВ1

Тема проекта: ***Разработка программы сложной структуры методом нисходящего программирования.***

Исходные данные (технические требования) на проектирование

Приложения для шифрования данных методом стеганографии. Открытие файлов изображения, внедрение в него зашифрованного текста, сохранение изображения на диск. Реализация одной из функций на языке ассемблер.

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio.

Объем работы по курсу

1. **Расчетная часть**

Разработка программы.

2. **Графическая часть**

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. **Экспериментальная часть**

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания "22" февраля

Дата защиты проекта "____" _____

Руководитель Токарев

Задание получил "22" февраля 2022г.

Студент Лошков Р.И.

Содержание

Введение 3
Постановка задачи 4
Выбор решения 5
Описание разработки программы 6
Отладка и тестирование. 6
Описание программы 7
Руководство пользователя 10
Список используемых источников 17
Приложение А. Листинг программы 18
Приложение В. Заставка 23

Введение

Спрятать секретное послание можно практически в любой цифровой объект — текстовый документ, лицензионный ключ, расширение файла. Например, редакторы сайта Genius.com, посвященного разбору треков рэп-исполнителей, использовали два вида апострофов в текстах таким образом, чтобы вместе они составляли словосочетание red handed («с поличным») на азбуке Морзе. Таким образом они защищали от копирования уникальный контент ресурса.

Однако один из самых удобных «контейнеров» — медиафайлы (картинки, аудио, видео и так далее). Они обычно достаточно большие сами по себе, а значит, и «довесок» может быть не таким маленьким, как в случае, скажем, с документом Word.

Секретную информацию можно записать в метаданные файла или же прямиком в основное содержимое. Возьмем, например, картинку. С точки зрения компьютера она представляет собой набор из сотен тысяч точек-пикселей. У каждого пикселя есть «описание» — информация о его цвете.

Постановка задачи

Необходимо разработать программу для зашифрования текста в bmp изображении.

Программа должна иметь интуитивно понятный интерфейс для пользователя. Необходимо определить названия пунктов меню, построить алгоритмы работы с массивами, файлами изображений для помещения в них текста, изучить функции для работы с файлами, чтобы сохранять результат в файл и загружать из файла.

Многомодульность программы. Программа должна быть поделена на логические модули. Это упростит поиск ошибок при отладке и тестировании программы, а также позволит легко расширять функционал программы. Использование сложных типов данных – массивов, файлов. Это необходимо для более простой и интуитивной обработки данных в коде программы. Режим работы видеосистемы – текстовый/графический. Для начала необходимо определиться с типом интерфейса и с элементами управления, затем необходимо изучить способы их реализации.

Ассемблерная вставка необходима для обеспечения высокой эффективности программы

Устройство ввода-вывода – клавиатура и мышь. Необходимо различать и идентифицировать действия, произведенные с их помощью, это облегчит использование программы.

Интерфейс должен быть построен на основе меню. Это необходимо для создания интуитивно понятного интерфейса для пользователя.

Заставка необходима для того, чтобы пользователь, запустивший программу, смог получить достаточную информацию о ней.

Выбор решения

При запуске программы открывается приветственный экран. Далее появляется интерфейс, состоящий из 4 кнопок, поля для ввода текста, поля для картинки, поля ключа

1) Кнопка “Выбрать текст”. При нажатии на неё открывается диалог выбора текстового файла. После выбора файла пользователем программа помещает текст файла в текстовое поле для ввода

2) Кнопка “Выбрать изображении”. При нажатии на неё открывается диалог выбора bmp файла. После выбора файла пользователем программа помещает изображение в поле для картинки

3)Кнопка “Зашифровать”. При нажатии на неё программа шифрует внутри bpm файла текст из поля для ввода. После открывается диалог сохранения нового bpm файла. При выборе места сохранения и имени файла программа сохраняет файл

4) Кнопка “Расшифровать”. При нажатии на неё открывается диалог выбора bpm файла. После выбора файла пользователем программа, взяв ключ из поля для ключа, расшифровывает файл и помещает расшифрованный текст в поле для ввода

5)Поле для текста. Пользователь может поместить в поле текст, который необходимо зашифровать, а также редактировать текст, помещенный в поле с помощью кнопки “Выбрать текст”.

6)Поле для картинки. В поле помещается картинка, выбранная пользователем, которая, при нажатии кнопки “Зашифровать”, заменяется на картинку, с зашифрованным текстом

7)Поле ключа. В поле пользователь вводит ключ, необходимый для расшифровки картинки. Ключ помещается в это поле программой после нажатия кнопки “Зашифровать”

Описание разработки программы

Разработанная программа состоит из нескольких модулей, которые реализовывались в следующем порядке:

1)Main.cpp

2)Form.h

3)mashinniy.cpp

4)zastavka.h

Отладка и тестирование

В качестве среды разработки была выбрана программа MicrosoftVisualStudio 2010. Программа обладает всеми средствами необходимыми при разработке и отладке программы. Для отладки использовались несколько возможностей VisualStudio: точка останова, трассировка, анализ содержимого переменных.

Тестирование проводилось во время разработки и также после завершения разработки. В ходе нее было выявлено огромное количество проблем, связанных с работой с файлами, работой с памятью.

Описание программы

Приложение coursovaya.exe является основным модулем программы. При запуске программы происходит вывод заставки-приветствия, после чего появляется графический интерфейс взаимодействия с программой. Описание состояний программы выполнено в таблице ниже.

Таблица 1.

Действие пользователя	Действие программы
Нажата кнопка «Выбрать текст»	Запускается диалог выбора файла с текстом
Нажата кнопка «Выбрать изображение»	Запускается диалог выбора файла с изображением
Нажата кнопка «Зашифровать»	Программа зашифровывает текст с изображением и запускает диалог сохранения нового файла с изображением
Нажата кнопка «Расшифровать»	Программа запускает диалог выбора файла с изображением, расшифровывает его и помещает расшифрованный текст в поле для ввода

Далее на рисунке 1 представлена схема взаимодействия программы.

Рисунок.1.1

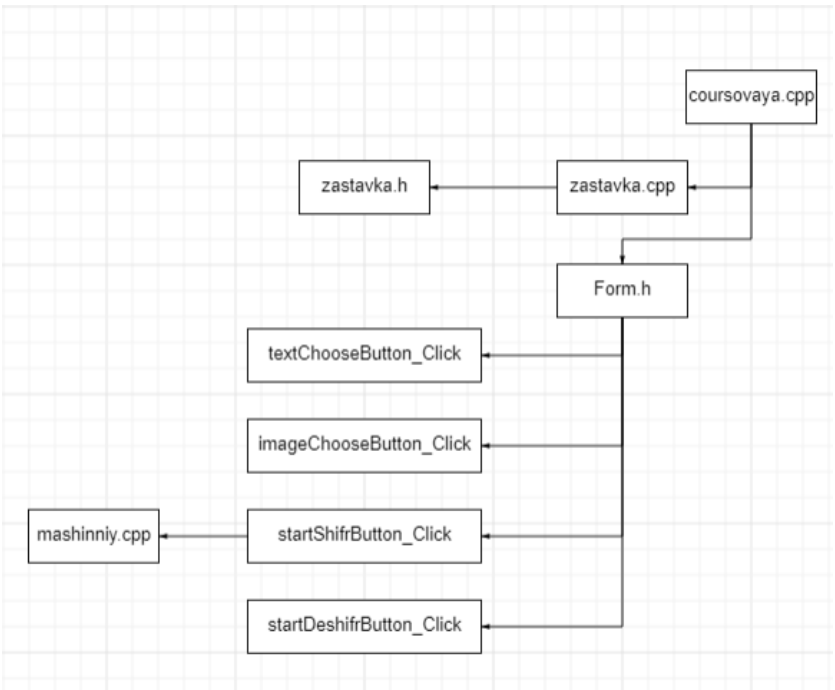


Рисунок - схема взаимодействия программы

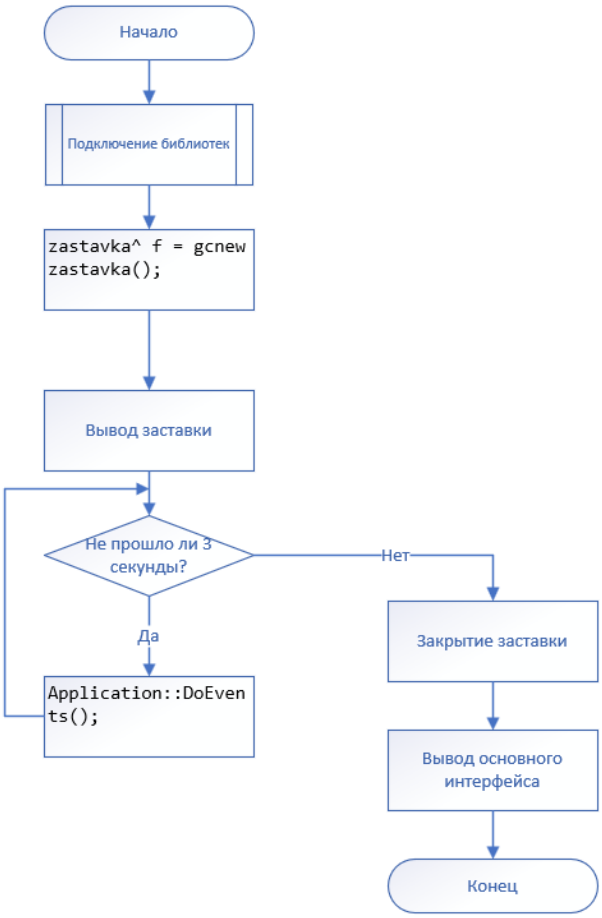
Ниже, на рисунке 2 представлена схема данных.

Рисунок 1.2

Рисунок – схема данных

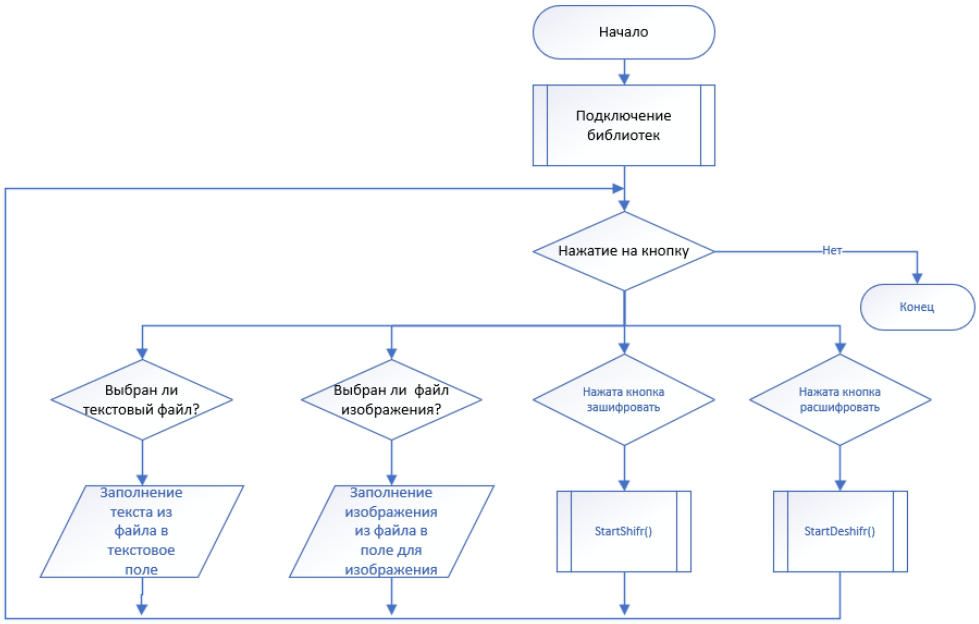
Схема программы. Класс Main

Рисунок 1.3



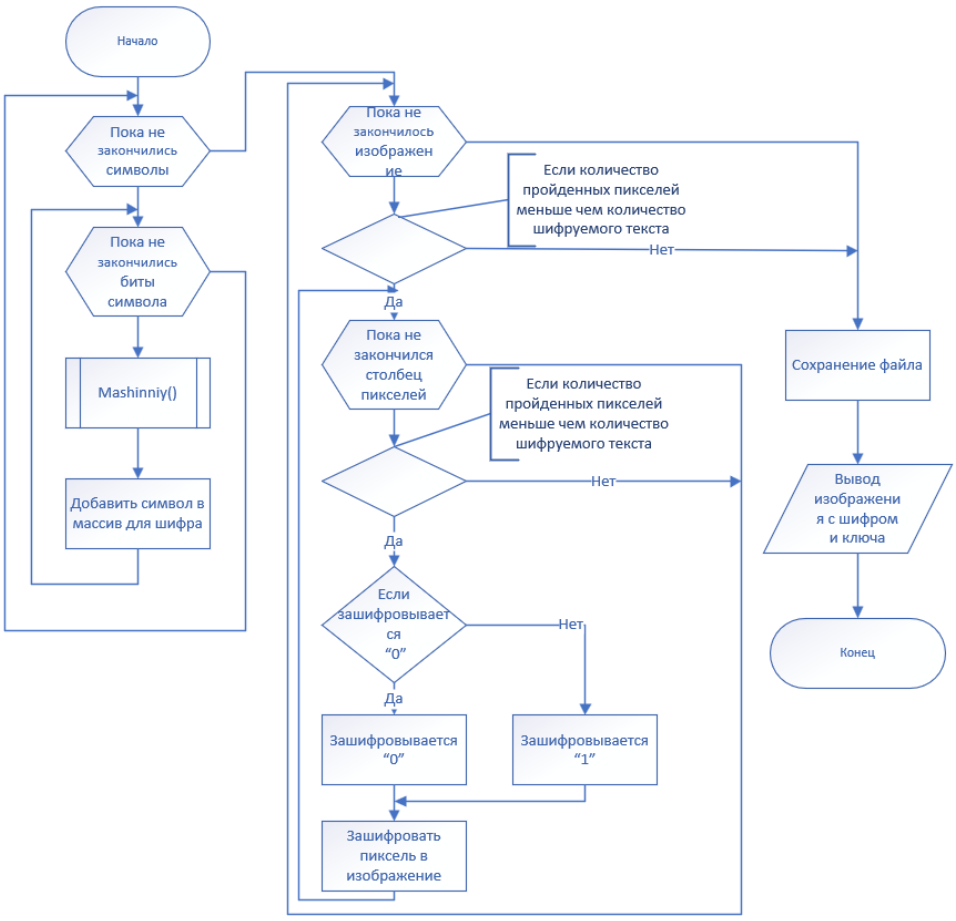
Класс Form1.h

Рисунок 1.4



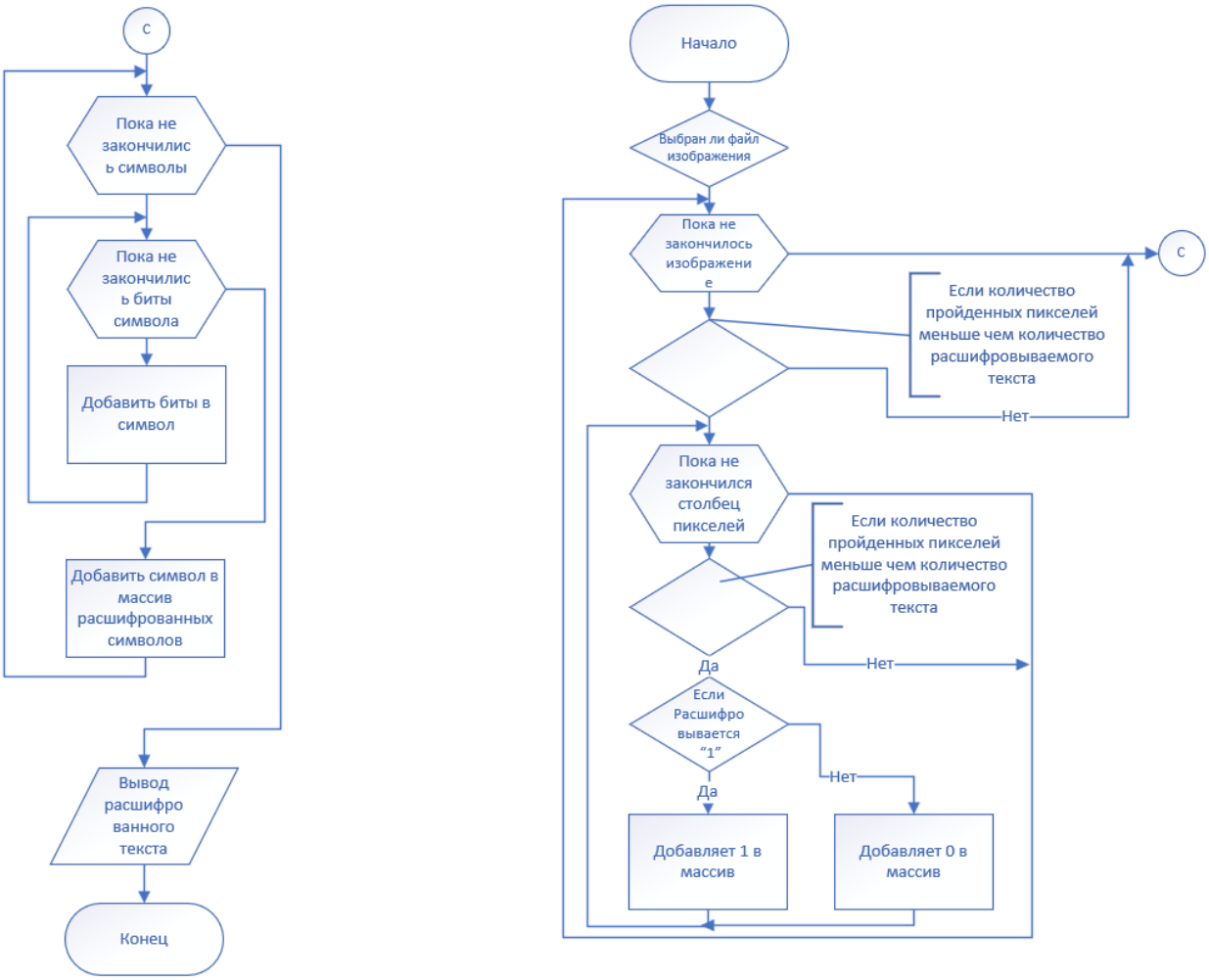
Функция startShifr

Рисунок 1.5



Функция startDeshifr

Рисунок 1.6

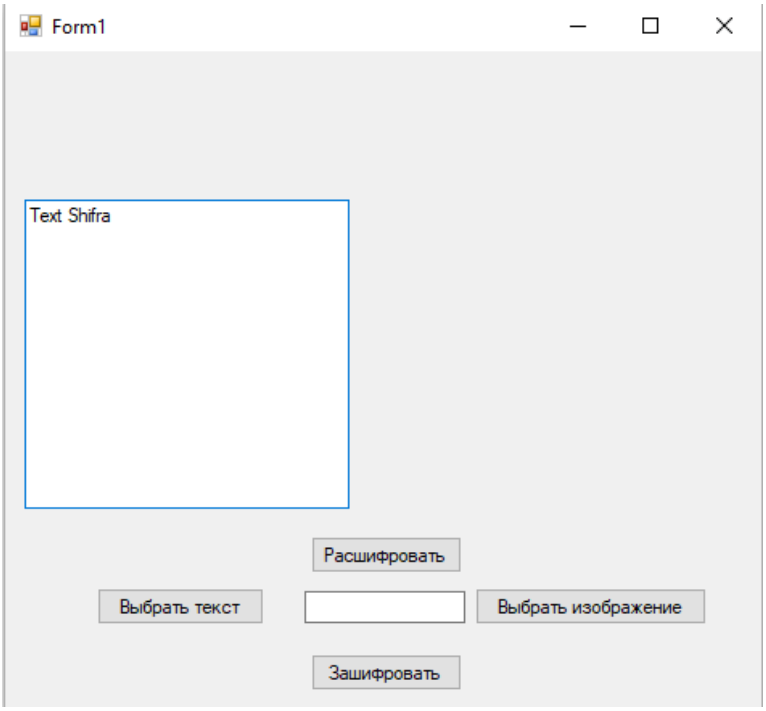


Руководство пользователя

Программа Steganografiya.exe предназначена для зашифровывания текста в bmp-изображения и расшифровывания текста из bmp-изображения. Программа имеет интуитивно понятный интерфейс и поддерживает такие операции как ввод текста шифра вручную, ввод текста шифра из файла, выбор файла изображения, зашифровывание и расшифровывание bmp-изображения

Для ввода текста шифра вручную необходимо кликнуть на поле для текста и набрать текст шифра

Рисунок 2.1.1



Для ввода текста шифра из файла необходимо нажать кнопку Выбрать текст, после этого откроется диалог выбора файла. Необходимо выбрать текстовый файл с текстом шифра. После этого текст из файла будет помещен в поле для текста

Рисунок 2.2.1

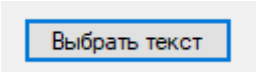


Рисунок 2.2.2

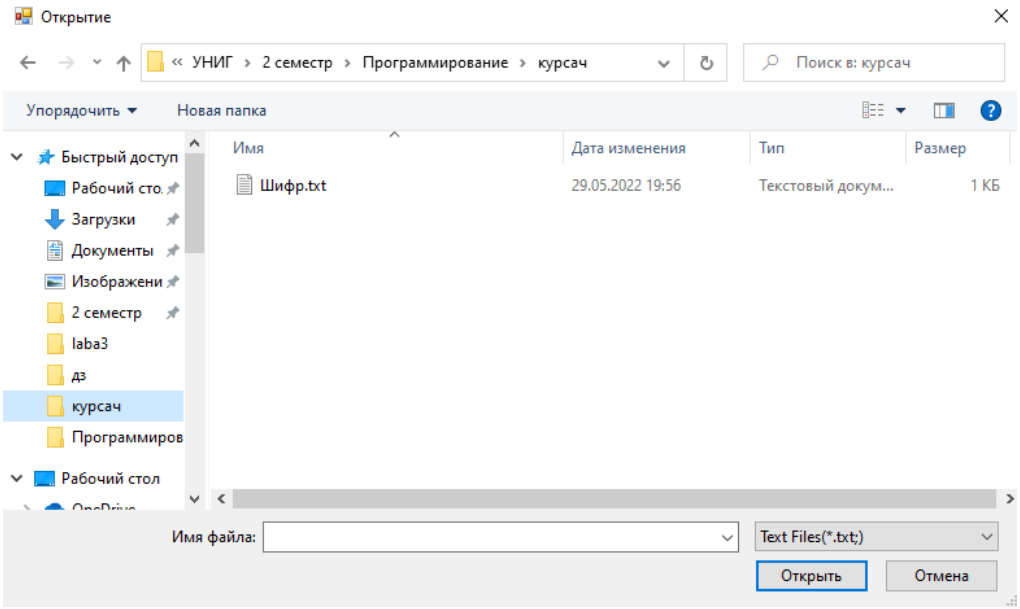
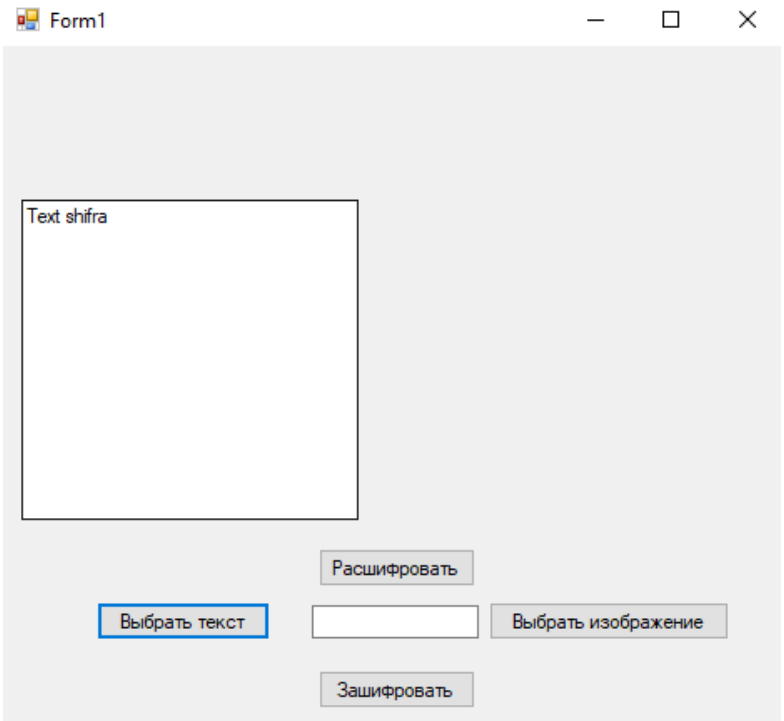


Рисунок 2.2.3



Для выбора изображения необходимо нажать на кнопку Выбрать изображение после этого откроется диалог выбора файла. Необходимо выбрать файл с bmp-изображением. После этого изображение будет помещено в поле для изображения

Рисунок 2.3.1

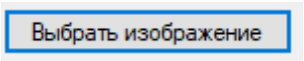


Рисунок 2.3.2

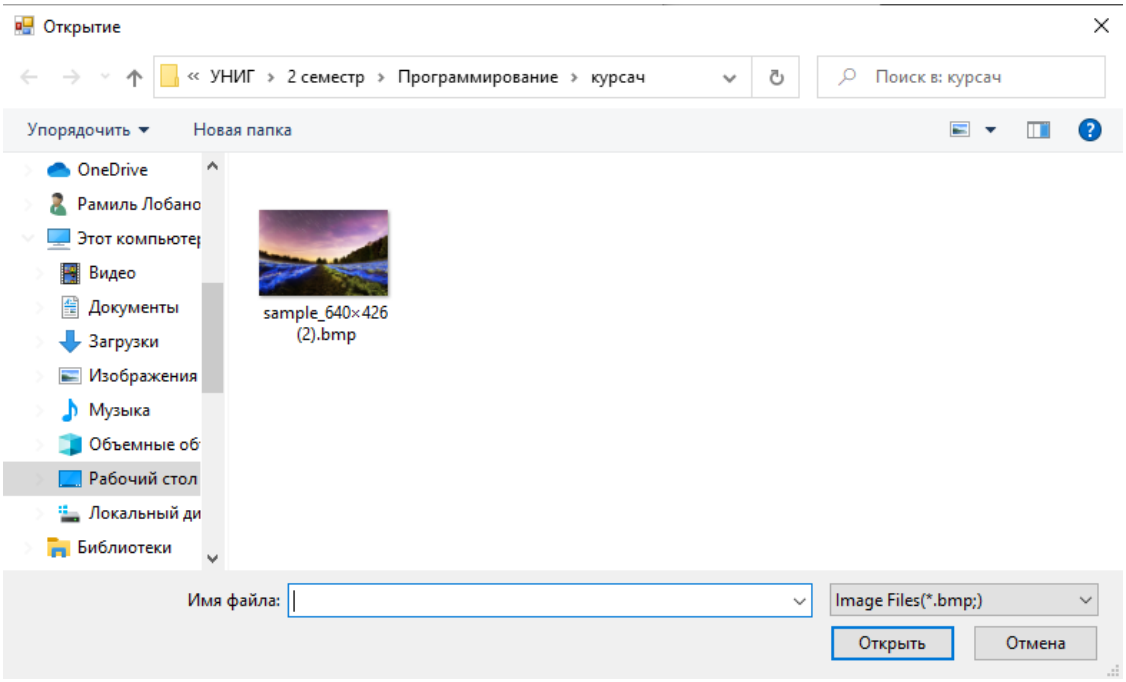
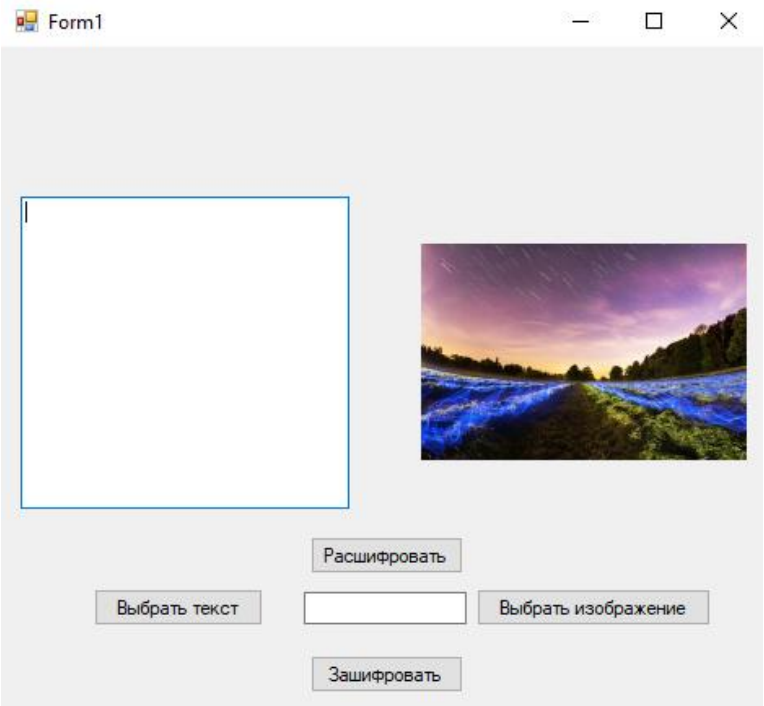


Рисунок 2.3.3



Для зашифровывания текста необходимо нажать кнопку Зашифровать (при этом поле для текста и поле изображения должны быть не пустыми) после этого откроется диалог сохранения файла. Необходимо выбрать имя файла и сохранить изображение. При этом в поле для текста помещается подсказка, а в поле ключа помещается ключ

Рисунок 2.4.1

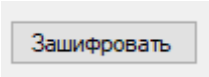


Рисунок 2.4.2

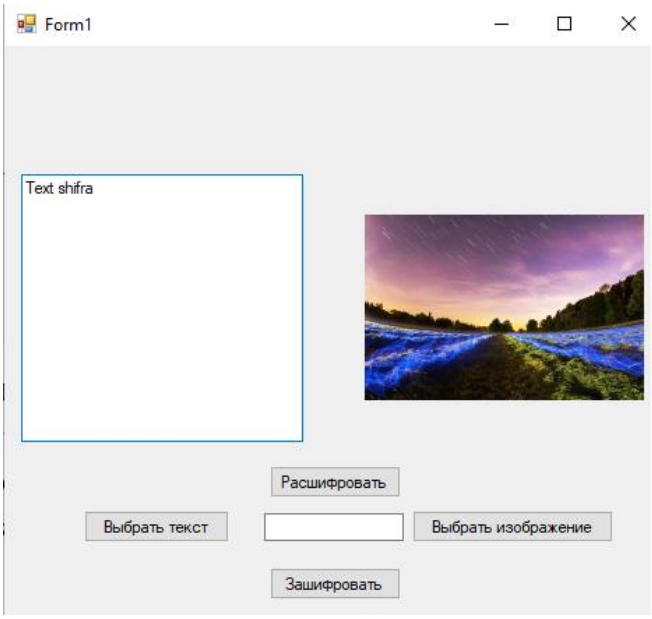


Рисунок 2.4.3

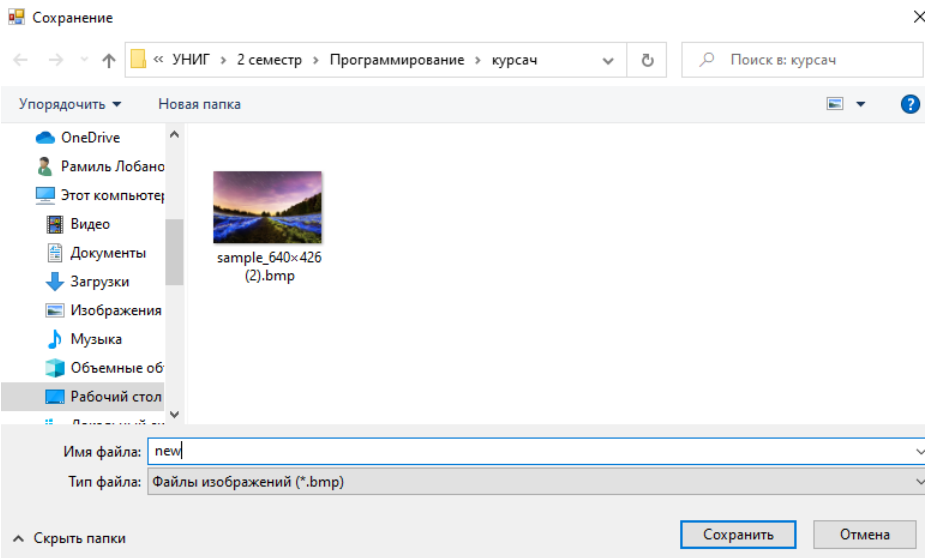
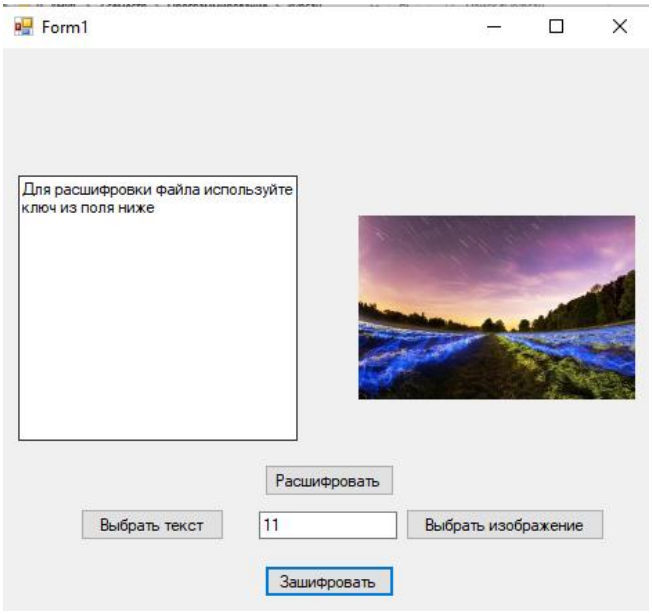


Рисунок 2.4.4



Для расшифровывания текста необходимо нажать кнопку Расшифровать (при этом поле ключа должно быть не пустым и должно быть заполнено ключом, выданным программой при зашифровании данного изображения) после этого откроется диалог выбора файла изображения. Необходимо выбрать файл. После в поле для текста помещается текст, зашифрованный в картинке.

Рисунок 2.5.1

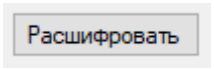


Рисунок 2.5.2

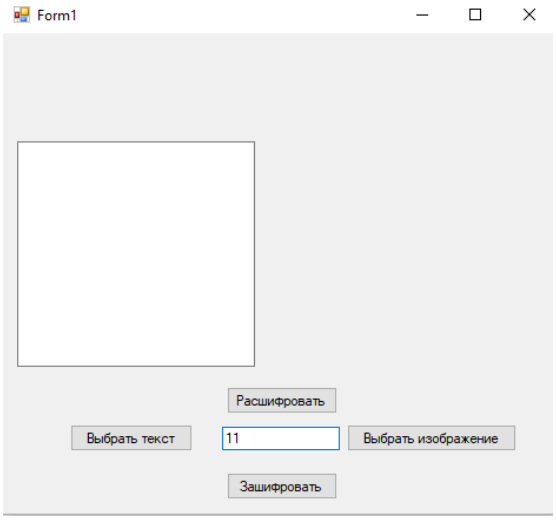


Рисунок 2.5.3

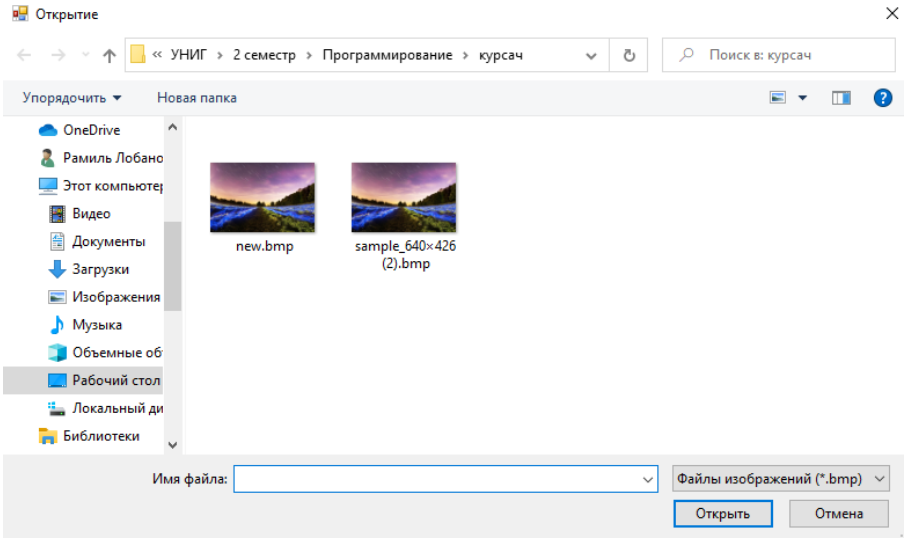
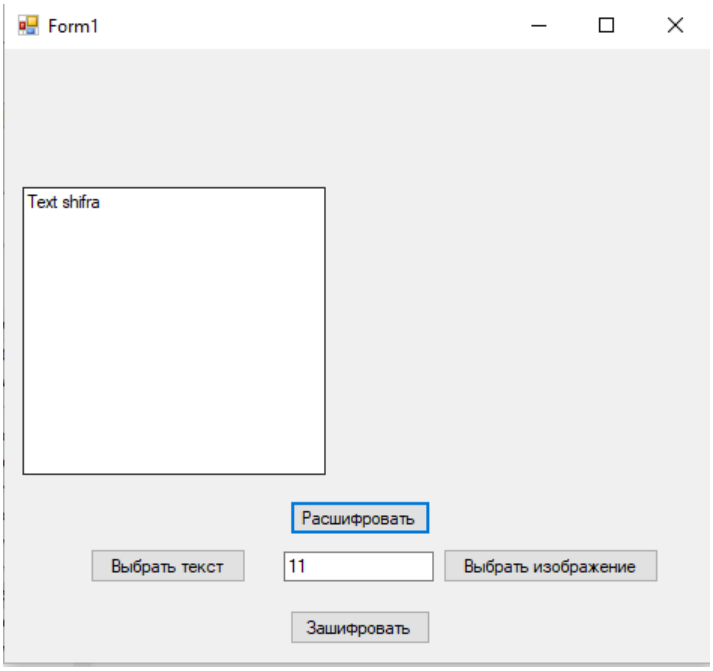


Рисунок 2.5.4



Заключение

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ. Была освоена работа с файлами, изучены функции работы с графическим интерфейсом и оперативной памятью. Также были получены основные навыки отладки и тестирования программ и программирования в среде VisualStudio 2010 на языках Си/C++ и Assembler.

В рамках курсовой работы была написана программа для зашифровывания и расшифровывания текста в bmp-изображении.

В дальнейшем программу можно улучшить, путем добавления дополнительного уровня шифрования, что позволит уменьшить шанс нахождения шифра сторонними лицами. Добавить дополнительные типы изображений (jpeg, png)

Список используемых источников

1. Прата С. Язык программирования C++. 2019 г.

2. А.А. Тюгашев. Языки программирования. Учебное пособие. 2018 г.

3. Руслан Аблязов. Программирование на ассемблере на платформе x86-64. 2016 г.

Приложение А

Листинги программы

Файл Main.cpp

```
#include "stdafx.h"
#include "Form1.h"
#include "zastavka.h"
#include <stdio.h>
#include "locale.h"
using namespace coursovaya;
[STAThreadAttribute]
void main(array<System::String ^> ^argv)
{
    Application::SetCompatibleTextRenderingDefault(false);
    zastavka^ f = gcnew zastavka();
    Application::EnableVisualStyles();
    DateTime end = DateTime::Now + TimeSpan::FromSeconds(4);
    f->Show();
    while(end >DateTime::Now){
        Application::DoEvents();
    }
    f->Close();

    Application::Run(gcnew Form1());
}
```

Файл zastavka.h

```
#pragma once
namespace coursovaya {
	using namespace System;
	using namespace System::Collections::Generic;
	using namespace System::ComponentModel;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace System::Linq;
	using namespace System::Text;
	using namespace System::Windows::Forms;
	using namespace System::IO;
	using namespace System::Collections;
	using namespace System::Text;
	public ref class zastavka : public System::Windows::Forms::Form
	{
	public:
		zastavka(void)
		{
			InitializeComponent();
		}

	protected:

		~zastavka()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::PictureBox^  pictureBox1;
	protected:
```

```
	private:
		System::ComponentModel::Container ^components;
#pragma region Windows Form Designer generated code
		void InitializeComponent(void)
		{
			System::ComponentModel::ComponentResourceManager^  resources = (gcnew
System::ComponentModel::ComponentResourceManager(zastavka::typeid));
			this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this-
>pictureBox1))->BeginInit();
			this->SuspendLayout();                    this->pictureBox1->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources->GetObject(L"pictureBox1.Image")));
			this->pictureBox1->Location = System::Drawing::Point(0, -1);
			this->pictureBox1->Name = L"pictureBox1";
			this->pictureBox1->Size = System::Drawing::Size(287, 264);
			this->pictureBox1->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::Zoom;
			this->pictureBox1->TabIndex = 0;
			this->pictureBox1->TabStop = false;
			this->pictureBox1->Click += gcnew System::EventHandler(this,
&zastavka::pictureBox1_Click);
			this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
			this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
			this->ClientSize = System::Drawing::Size(284, 261);
			this->Controls->Add(this->pictureBox1);
			this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::None;
			this->Name = L"zastavka";
			this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
			this->Text = L"zastavka";
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this-
>pictureBox1))->EndInit();
			this->ResumeLayout(false);

		}
#pragma endregion
	private: System::Void pictureBox1_Click(System::Object^  sender,
System::EventArgs^  e) {
			 }
	};
}
```

Файл Form.h

```
#pragma once
	#using <system.drawing.dll>
#include "mashinniy.h"
#include <cstdlib>
namespace coursovaya {
	using namespace System;
	using namespace System::Collections::Generic;
	using namespace System::ComponentModel;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace System::Linq;
	using namespace System::Text;
	using namespace System::Windows::Forms;
	using namespace System::IO;
	using namespace System::Collections;
	using namespace System::Text;
	public ref class Form1 : public System::Windows::Forms::Form
	{
	public:
```

```
		Form1(void)
		{
			InitializeComponent();
		}

	protected:
		~Form1()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::Button^  textChooseButton;
	private: System::Windows::Forms::Button^  imageChooseButton;
	protected:

	private: System::Windows::Forms::TextBox^  textBox1;
	private: System::Windows::Forms::PictureBox^  pictureBox1;
	private: System::Windows::Forms::Button^  startShifrButton;
	private: System::Windows::Forms::Button^  startDeshifrButton;
	private: System::Windows::Forms::TextBox^  keyBox;
	private:
		System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
		void InitializeComponent(void)
		{
			this->textChooseButton = (gcnew System::Windows::Forms::Button());
			this->imageChooseButton = (gcnew System::Windows::Forms::Button());
			this->textBox1 = (gcnew System::Windows::Forms::TextBox());
			this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
			this->startShifrButton = (gcnew System::Windows::Forms::Button());
			this->startDeshifrButton = (gcnew System::Windows::Forms::Button());
			this->keyBox = (gcnew System::Windows::Forms::TextBox());
			(cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this->pictureBox1))->BeginInit();
			this->SuspendLayout();
			this->textChooseButton->Location = System::Drawing::Point(57, 333);
			this->textChooseButton->Name = L"textChooseButton";
			this->textChooseButton->Size = System::Drawing::Size(104, 23);
			this->textChooseButton->TabIndex = 0;
			this->textChooseButton->Text = L"Выбрать текст";
			this->textChooseButton->UseVisualStyleBackColor = true;
			this->textChooseButton->Click += gcnew System::EventHandler(this, &Form1::textChooseButton_Click);
			this->imageChooseButton->Location = System::Drawing::Point(292, 333);
			this->imageChooseButton->Name = L"imageChooseButton";
			this->imageChooseButton->Size = System::Drawing::Size(144, 23);
			this->imageChooseButton->TabIndex = 1;
			this->imageChooseButton->Text = L"Выбрать изображение";
			this->imageChooseButton->UseVisualStyleBackColor = true;
			this->imageChooseButton->Click += gcnew System::EventHandler(this, &Form1::imageChooseButton_Click);
			this->textBox1->Location = System::Drawing::Point(12, 92);
			this->textBox1->Multiline = true;
			this->textBox1->Name = L"textBox1";
			this->textBox1->Size = System::Drawing::Size(202, 192);
			this->textBox1->TabIndex = 2;
			this->textBox1->TextChanged += gcnew System::EventHandler(this, &Form1::textBox1_TextChanged);
			this->pictureBox1->Location = System::Drawing::Point(258, 92);
			this->pictureBox1->Name = L"pictureBox1";
			this->pictureBox1->Size = System::Drawing::Size(200, 192);
```

```
                    this->pictureBox1->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::Zoom;
                    this->pictureBox1->TabIndex = 3;
                    this->pictureBox1->TabStop = false;
                    this->pictureBox1->Click += gcnew System::EventHandler(this,
&Form1::pictureBox1_Click);
                    this->startShifrButton->Location = System::Drawing::Point(190, 374);
                    this->startShifrButton->Name = L"startShifrButton";
                    this->startShifrButton->Size = System::Drawing::Size(94, 23);
                    this->startShifrButton->TabIndex = 4;
                    this->startShifrButton->Text = L"Зашифровать";
                    this->startShifrButton->UseVisualStyleBackColor = true;
                    this->startShifrButton->Click += gcnew System::EventHandler(this,
&Form1::startShifrButton_Click);
                    this->startDeshifrButton->Location = System::Drawing::Point(190,
301);
                    this->startDeshifrButton->Name = L"startDeshifrButton";
                    this->startDeshifrButton->Size = System::Drawing::Size(94, 23);
                    this->startDeshifrButton->TabIndex = 5;
                    this->startDeshifrButton->Text = L"Расшифровать";
                    this->startDeshifrButton->UseVisualStyleBackColor = true;
                    this->startDeshifrButton->Click += gcnew System::EventHandler(this,
&Form1::startDeshifrButton_Click);
                    this->keyBox->Location = System::Drawing::Point(186, 335);
                    this->keyBox->Name = L"keyBox";
                    this->keyBox->Size = System::Drawing::Size(100, 20);
                    this->keyBox->TabIndex = 6;
                    this->keyBox->TextChanged += gcnew System::EventHandler(this,
&Form1::keyBox_TextChanged);
                    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                    this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
                    this->ClientSize = System::Drawing::Size(470, 409);
                    this->Controls->Add(this->keyBox);
                    this->Controls->Add(this->startDeshifrButton);
                    this->Controls->Add(this->startShifrButton);
                    this->Controls->Add(this->pictureBox1);
                    this->Controls->Add(this->textBox1);
                    this->Controls->Add(this->imageChooseButton);
                    this->Controls->Add(this->textChooseButton);
                    this->Name = L"Form1";
                    this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
                    this->Text = L"Form1";
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this-
>pictureBox1))->EndInit();
                    this->ResumeLayout(false);
                    this->PerformLayout();

            }
#pragma endregion
      String^ imageFileName;
      String^ textShifr;
      private: System::Void pictureBox1_Click(System::Object^  sender,
System::EventArgs^  e) {
                   }
      private: System::Void textBox1_TextChanged(System::Object^  sender,
System::EventArgs^  e) {
                   }
private: System::Void textChooseButton_Click(System::Object^  sender, System::EventArgs^
e) {

      OpenFileDialog^ openDlg = gcnew OpenFileDialog();
    openDlg->Filter = "Text Files(*.txt;)|*.TXT;";
    if (openDlg->ShowDialog() == System::Windows::Forms::DialogResult::OK){
            String^ TextFileName = openDlg->FileName;
```

```
            textShifr= System::IO::File::ReadAllText(TextFileName);
            textBox1->Text =textShifr;

      }
       }
private: System::Void imageChooseButton_Click(System::Object^  sender, System::EventArgs^
e) {
      OpenFileDialog^ openDlg = gcnew OpenFileDialog();
    openDlg->Filter = "Image Files(*.bmp;)|*.bmp;";
    if (openDlg->ShowDialog() == System::Windows::Forms::DialogResult::OK){
            imageFileName = openDlg->FileName;
            Image^ image1= Image::FromFile(imageFileName);
            pictureBox1->Image=image1;//
            pictureBox1->Refresh();
            this->Refresh();
            delete image1;
      }
             }
private: System::Void startShifrButton_Click(System::Object^  sender, System::EventArgs^
e) {
                  try{
                  textShifr=textBox1->Text;
                  FileStream^ rFile = gcnew FileStream(imageFileName, FileMode::Open);
                  Bitmap^ bImage= gcnew Bitmap(rFile);
                  rFile->Close();
                  int countTextShifr=textShifr->Length;
                  char* textShifrBinary=(char*)malloc(countTextShifr*8);
                  for(int i=0;i<countTextShifr;i++){
                        char c=textShifr[i];
                        for(int j=0;j<8;j++)
                        {
                              char newc=mashinniy(c);
                              (newc==1) ? textShifrBinary[i*8+j]=1:
textShifrBinary[i*8+j]=0;
                              c= c>>1;
                        }
                  }
                  bool endShifrFlag=false;
                  for(int i=0;i<bImage->Width;i++)
                  {
                        for(int j=0;j<bImage->Height;j++)
                        {
                              Color oColor = bImage->GetPixel(i,j);
                              Color nColor;
                              if(i*bImage->Height+j<=countTextShifr*8){
                                    long letter=(oColor.ToArgb());
                                    if(textShifrBinary[i*bImage->Height+j]==0)
                                          letter &= ~(1UL << 0);
                                    else
                                          letter |= 1UL << 0;
                                    nColor = Color::FromArgb(letter);}
                              else
                              {
                                    endShifrFlag=true;
                                    break;
                              }
                              bImage->SetPixel(i,j,nColor);

                        }
                        if(endShifrFlag)
                              break;
                  }
                  String^ saveImageFileName;
                  SaveFileDialog^ saveDlg = gcnew SaveFileDialog();
```

```
                    saveDlg->Filter = "Файлы изображений (*.bmp)|*.bmp|Все файлы
(*.*)|*.*";
                    if (saveDlg->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
                            saveImageFileName=saveDlg->FileName;
                            FileStream^ wImageFile;
                            wImageFile = gcnew FileStream(saveImageFileName,
FileMode::Create);
                            bImage-
>Save(wImageFile,System::Drawing::Imaging::ImageFormat::Bmp);
                            wImageFile->Close();

                    pictureBox1->Image= bImage;
                    keyBox->Text=countTextShifr+"";
                    textBox1->Text="Для расшифровки файла используйте ключ из поля ниже";
                    }
                    }
                    catch(...)
                    {return;}
                }
private: System::Void startDeshifrButton_Click(System::Object^  sender,
System::EventArgs^  e) {
                    try{
                    String^ shifr="";
                    String^ deshifrImageFileName;
                    OpenFileDialog^ openDlg = gcnew OpenFileDialog();
                    openDlg->Filter = "Файлы изображений (*.bmp)|*.bmp|Все файлы
(*.*)|*.*";
                    if (openDlg->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
                    {
                            deshifrImageFileName=openDlg->FileName;
                            FileStream^ deshifrImageFile;
                            deshifrImageFile = gcnew FileStream(deshifrImageFileName,
FileMode::Open);
                            Bitmap^ bImage= gcnew Bitmap(deshifrImageFile);
                            deshifrImageFile->Close();
                            int countTextShifr=Int32::Parse(keyBox->Text);
                            char* textShifrBinary=(char*)malloc(countTextShifr*8+1);
                            bool endShifrFlag=false;
                            for(int i=0;i<bImage->Width;i++)
                            {
                                    for(int j=0;j<bImage->Height;j++)
                                    {
                                            if(i*bImage->Height+j<=countTextShifr*8)
                                            {
                                            Color oColor = bImage->GetPixel(i,j);
                                            long letter = oColor.ToArgb();
                                            if(abs(letter%2)==1)
                                                    textShifrBinary[i*bImage->Height+j]=1;
                                            else
                                                    textShifrBinary[i*bImage->Height+j]=0;
                                            }
                                            else
                                            {
                                                    endShifrFlag=true;
                                                    break;
                                            }

                                    }
                                    if(endShifrFlag)
                                                    break;
                            }
                            array<Char>^ array1=gcnew array<Char>(countTextShifr);
                            for(int i=0;i<countTextShifr;i++)
```

```
                        {
                                unsigned char c=0;
                                for(int j=0;j<7;j++)
                                {
                                        c+=textShifrBinary[i*8+j]*128;
                                        c=c>>1;
                                };
                                array1[i]=c;
                        }
                    textBox1->Text=gcnew String(array1);
                 }
            }
                  catch(...){
                        return;}
            }
private: System::Void keyBox_TextChanged(System::Object^  sender, System::EventArgs^  e)
{
            }
};
}
```

Файл mashinniy.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include "StdAfx.h"
#include <cmath>
char mashinniy(char c) {
       char newc;
    _asm{
              mov al,c;
              mov ah,1;
              and al,ah;
              mov newc,al;
    }
    return newc;
}
```

Приложение В

Информационная заставка

Приложение В.1 – Информационная заставка

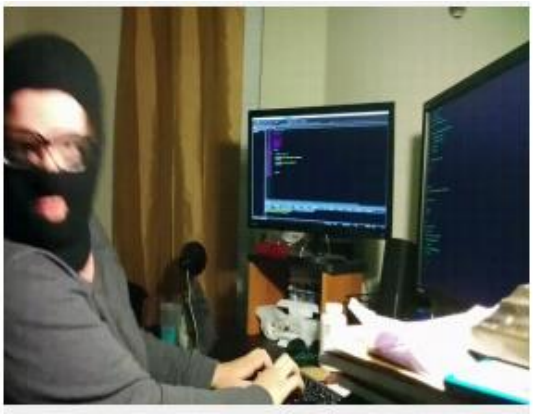



Рисунок 3