

## Содержание

Введение.....	3
1 Анализ требований и определение спецификаций на программный продукт.....	5
1.1 Функциональное назначение проектируемой программы.....	5
1.2 Функциональные характеристики .....	5
1.3 Требования к аппаратным и программным средствам.....	7
2 Проектирование и разработка программного продукта .....	8
2.1 Проектирование и описание базового алгоритма решения задачи.....	8
2.2 Описание программного продукта.....	11
3 Описание интерфейса.....	13
4 Руководство пользователя .....	16
Заключение .....	17
Список используемых источников.....	18
Приложение А .....	19
Приложение Б.....	22

## Введение

C++ в настоящее время считается господствующим языком, используемым для разработки коммерческих программных продуктов. В последние годы это господство слегка поколебалось вследствие аналогичных претензий со стороны такого языка программирования, как Java, но маятник общественного мнения качнулся в другую сторону, и многие программисты, которые бросили C++ ради Java, в последнее время поспешили вернуться к своей прежней привязанности. В любом случае эти два языка настолько похожи, что, изучив один из них, вы автоматически осваиваете 90% другого.

При разработке языка C++ одним из важнейших критериев выбора была простота. Когда возникал вопрос, что упростить: руководство по языку и другую документацию или транслятор, - то выбор делали в пользу первого. Огромное значение придавалось совместимости с языком C, что помешало удалить его синтаксис.

В C++ нет типов данных и элементарных операций высокого уровня. Например, не существует типа матрица с операцией обращения или типа строка с операцией конкатенации. Если пользователю понадобятся подобные типы, он может определить их в самом языке. Программирование на C++ по сути сводится к определению универсальных или зависящих от области приложения типов. Хорошо продуманный пользовательский тип отличается от встроенного типа только способом определения, но не способом применения.

Из языка исключались возможности, которые могут привести к накладным расходам памяти или времени выполнения, даже если они непосредственно не используются в программе. Например, было отвергнуто предложение хранить в каждом объекте некоторую служебную информацию. Если пользователь описал структуру, содержащую две величины, занимающие по 16 разрядов, то гарантируется, что она поместится в 32-х разрядный регистр.

Язык C++ проектировался для использования в довольно традиционной среде, а именно: в системе программирования C операционной системы UNIX. Но есть вполне обоснованные доводы в пользу использования C++ в более богатой программной среде. Такие возможности, как динамическая загрузка, развитые системы трансляции и базы данных для хранения определений типов, можно успешно использовать без ущерба для языка.

Типы C++ и механизмы упрятывания данных рассчитаны на определенный синтаксический анализ, проводимый транслятором для обнаружения случайной порчи данных. Они не обеспечивают секретности данных и защиты от умышленного нарушения правил доступа к ним. Однако, эти

средства можно свободно использовать, не боясь накладных расходов памяти и времени выполнения программы. Учтено, что конструкция языка активно используется тогда, когда она не только изящно записывается на нем, но и вполне по средствам обычным программам.

В данной курсовой работе будет рассмотрена разработка программы по решению задачи средствами языка программирования C++.

Целью данной курсовой работы является создание программы, задачей которой является изменять порядок двух соседних букв.

Актуальность темы заключается в том, что современное общество невозможно представить без использования компьютера.

Основные задачи при написании курсовой работы:

- Проанализировать условие задачи, составить таблицы данных.
  - Разработать программу.
  - Протестировать и проанализировать программу.
  - Выявить пути оптимизации программы
  - Среда разработки данной программы – C++.

# 1 Анализ требований и определение спецификаций на программный продукт

## 1.1 Функциональное назначение проектируемой программы

Основные задачи программы:

- последовательные вычисления
- ветвления
- повторяющиеся вычисления
- Программа должна работать без сбоев.

## 1.2 Функциональные характеристики

Перед началом работы необходимо определить функции, которые должна выполнять программа, исходные данные и результаты, их форму и характеристики, а также требования к условиям эксплуатации.

К разрабатываемой системе предъявляются следующие функциональные требования:

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС и запуска исполняемого файла системы;
- при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;
- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

В целях надежности системы она должна удовлетворять следующим требованиям:

- разработанная программа должна обладать средствами защиты от ошибочных действий пользователей;
- все ошибки должны отображаться с комментариями или подсказками по их устранению;
- гарантировать сохранность данных при сбоях в работе внешних устройств.

Для повышения надежности необходимо принять следующие меры:

- сконфигурировать аппаратные и программные средства в соответствии с техническими требованиями;
- периодически осуществлять резервное копирование информации;
- регулярно проверять целостность базы данных;
- поддерживать исправность оборудования.

### 1.3 Требования к аппаратным и программным средствам

Требования к программному обеспечению — совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Создаются в процессе разработки требований к программному обеспечению (ПО), в результате анализа требований.

Минимальные системные требования для использования C++:

- Минимальные требования: Windows 7 или 10, 32 или 64 бит
- 256 Мб оперативной памяти
- Частота процессора 500МГц
- Поддержка OpenGL.

Рекомендуемые системные требования программы:

- Компьютер с четырехъядерным процессором с частотой 2.5 ГГц и выше
- Оперативная память от 4 Гб, в идеале 8 Гб и выше
- Операционная система: Windows 10 64-bit или Mac OS X 10.9.2 и выше
- Минимальная видеокарта NVIDIA GeForce 470 GTX или AMD Radeon 6870 HD. Лучше NVIDIA GeForce GTX 970;

## 2 Проектирование и разработка программного продукта.

### 2.1 Проектирование базового алгоритма решения задачи

В процессе создания курсовой работы «Перестановки двух соседних букв местами с помощью шифра перестановки», был разработан алгоритм решения поставленной задачи. По этому алгоритму на языке C++ составлена и отлажена программа, удачно прошедшая тестирование.

Алгоритм – совокупность чётко определённых правил для решения задачи за конечное число шагов. В работе были учтены все необходимые свойства алгоритма: массовость, понятность, определённость, дискретность, конечность, результативность.

Блок-схема —распространённый тип схем(графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединённых между собой линиями, указывающими направление последовательности.

Рассмотрим алгоритм замены местами двух соседних букв:

К шифрам простой замены относятся многие способы шифрования, возникшие в древности или средневековье, как, например, Атбаш (также читается как этбаш) или шифр Цезаря. Для вскрытия подобных шифров используется частотный криптоанализ.

Является частным случаем шифра подстановки.

Отметим, что шифр простой замены не всегда подразумевает замену буквы на какую-то другую букву. Допускается использовать замену буквы на число. К примеру представим некий шифр-алфавит: А - 33; Б - 17; В - 8; Г - 16; Д - 2; Е - 15; Ё - 14; Ж - 13; З - 73; И - 98; Й - 10; К - 97; Л - 96; М - 24; Н - 0; О - 11; П - 5; Р - 25; С - 7; Т - 3; У - 64; Ф - 26; Х - 66; Ц - 69; Ч - 4; Ш - 6; Щ - 36; Ъ - 21; Ы - 22; Ь - 23; Э - 37; Ю - 39; Я - 18.

Алгоритм преобразования текста в виде представлен на рисунке 1.

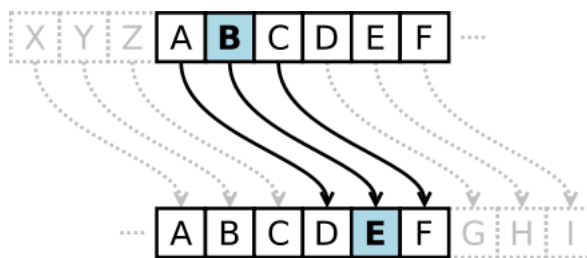


Рисунок 1 - .Перестановка букв местами

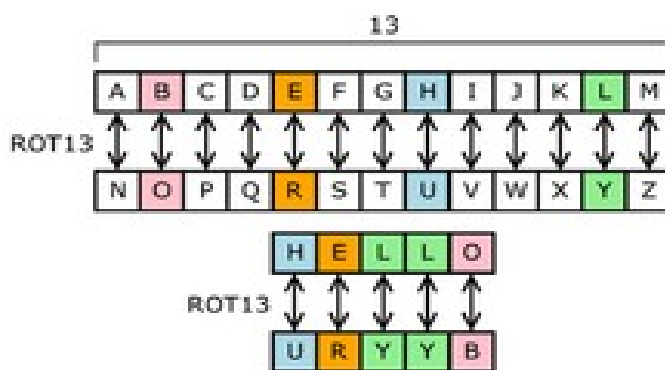


Рисунок 2 - Расшифровка слов



Для работы программы необходимо выполнить:

- Запуск программы.
- Написать значение для добавления размера матрицы.
- Создание Зашифровки через кнопку “Зашифровать ”
- Перестановка данных ввода через кнопку “Расшифровать”.

После того как пользователь нажимает на кнопку “Выполнить”

Данные, введённые в таблицу, изменят две соседние буквы.

## 2.2 Описание программного продукта

Приложение, разработанное в С++, состоит из ряда файлов, назначение которых представлено в таблице 1.

Таблица 1 – Состав файлов программы

Расширение	Назначение файла
Project1.cfg	Файл конфигурации проекта, где хранятся данные о состоянии рабочего пространства среды (установки проекта), в частности, используемые директивы компилятора.
Project1.dof	файл, содержащий опции проекта
CPP	это временный кеш С++ . Файл расширения
Project1.res	файл графических ресурсов - хранит курсоры, иконки и рисунки. Для редактирования графических ресурсов Borland С++ Builder предусмотрен редактор Image Editor.
Unit1.dcu	Это означает, что, если у вас есть чей-то, например, компонентный скомпилированный исходный код, все, что вам нужно сделать, это перевернуть его и получить код.
Unit1.ddp	файл, содержащий визуальную информацию о форме.

Таблица 2- Использование модулей в С++

Имя процедуры	Применение
Button1Click	Выполняет расчет скорости и выводит результат расчета в поле Label4.
Button2Click	
Edit1KeyPress	
TForm1.Button1Click	Событие, которое произойдет при нажатии на кнопку Button1
TForm1.Button2Click	Отвечает программы за завершение.
TForm1.Edit1KeyPress	проверяет, какая клавиша нажата при вводе исходных данных в поле редактирования Edit1.

### 3 Описание интерфейса

Когда вы разрабатываете класс в C++, первое, что вы должны продумать — открытый интерфейс для класса. Открытый интерфейс определяет, как ваш класс будет использоваться другими программистами (или вами), и после разработки и реализации он должен, как правило, оставаться постоянным. Вы можете добавлять что-то в интерфейс, но как только вы начали использовать класс, будет трудно удалить функции из общего интерфейса (если только они не используются и не были необходимы в первую очередь).

Но это не означает, что вы должны включать больше функциональности в своем классе, только чтобы позже решить, что удалить из интерфейса. Если вы так делаете, вы просто усложняете использование класса. Люди будут задавать вопросы вроде: «Почему есть четыре способа сделать это? Какой из них лучше? Какой способ выбрать мне?» Как правило, легче делать все просто и обеспечить один способ, если нет веских причин предложить вашему классу несколько методов с одинаковой функциональностью.

В то же время, только потому, что добавление методов для открытого интерфейса (вероятно) ничего не испортит, это не означает, что вы должны начинать с крошечного интерфейса. Прежде всего, если кто-нибудь решит унаследоваться от вашего класса, а в последствии вы выберете функцию с тем же именем, то возникнет путаница. Во-первых, если вы не объявите функцию виртуальной, то объект подкласса выберет функцию в зависимости от статического типа указателя. Это может создать проблемы. Кроме того, если вы объявите ее виртуальной, то у вас, возможно, возникнет проблема с тем, что она может предоставлять другую функциональность, отличную от оригинальной реализации этой функции. И, наконец, вы просто не можете добавлять чисто виртуальные функции классу, который уже используется, потому что никто из унаследовавших от него не будет реализовывать эту функцию.

Открытый интерфейс должен оставаться постоянным как можно дольше. В самом деле, хороший подход к проектированию классов — написать интерфейс до реализации, потому что это то, что определяет, как ваш класс взаимодействует с остальным миром (что более важно для программы в целом, чем фактическая реализация класса). Кроме того, если вы сначала пишете интерфейс, вы можете увидеть, как класс будет работать с другими классами, прежде чем вы на самом деле погрузитесь в детали реализации.

Таблица 3 – Табличные компоненты программы

Компонент	Описание
Form	Формы в С++ используются для размещения графических объектов, которые делают программу такой, какой она должна получиться при выполнении.
TableEditor1	для редактирования таблицы и сохранения её в формате XML
DataGrid	решение для показа связанных таблиц в одной компоненте, в DataGridView такой возможности нет. Не все компоненты доступны по умолчанию, в том числе, и DataGrid. Щёлкаем правой кнопкой мыши на панели элементов управления, Выбрать элементы, на вкладке "Компоненты .NET Framework" устанавливаем флажок возле элемента DataGrid пространства имен System.Windows.Forms, нажимаем ОК.
Edit	Строка ввода. Предназначена для ввода пользователем текстовой информации в виде одной строки. Имеет возможности по управлению вводимой информацией, например, переопределению символов до их появления в поле ввода, что используется в формах для ввода пароля.
StringGrid	Свойства RowCount и ColCount задают число строк и столбцов таблицы. Для доступа к ячейке таблицы используется свойство Cells, которое является двухмерным массивом всех ячеек таблицы (нумерация начинается с нуля).

Отладка и тестирование программного продукта это один из самых важных этапов разработки, без которого не представляется возможным написание законченного, правильно работающего приложения.

Под отладкой программы понимается процесс испытания работы программы и исправления обнаруженных при этом ошибок.

Проверка на компьютере правильности алгоритма производится с помощью тестов.

Тест – это конкретный вариант значений исходных данных, для которого известен ожидаемый результат.

Отладка программы – это процесс поиска и устранения ошибок в программе, который производится после её прогона на компьютере.

Тестирование – это процесс проверки правильности работы всей программы или ее составных частей.

Цель тестирования программы – поиск ошибок, которые могут препятствовать работе программы.

## 4 Руководство пользователя

Руководство пользователя — документ, назначение которого — предоставить людям помощь в использовании некоторой системы.

### 1) Запуск программы

Для того чтобы открыть программу нужно запустить файл "project1.cpp;"

### 2) Добавление и изменение букв местами

- Далее выбираем пункт "Создать таблицу" чтобы построить таблицу.

- Вводим значения в появившейся таблице и выбираем пункт "Расшифровка, Зашифровка"

- Готово в правой части интерфейса у вас появилась готовая таблица.

## Заключение

В своей курсовой работе я показал некоторые возможности языка C++ на примере написания таблицы позволяющей менять соседние буквы местами.

Это лишь малая доля тех возможностей, какие предоставляет этот язык. Различия с Паскалем очевидны и довольно существенны, причем эти отличия только усложняют работу программиста. Но функциональность C++ куда больше, и вариативность написания многих частей программы сводится к нескольким способам. C++ позволяет создать сложные приложения, не только работая в консольном режиме, но и в графическом. Многие приложения в Windows написаны именно на этом языке, поэтому о нем можно говорить как об очень мощном языке программирования.

В ходе написания курсового проекта были подробно рассмотрены профессиональные задачи.

В результате выполнения курсовой работы была разработана программа для «Перестановки двух соседних букв местами с помощью шифра»

Данная программа может пригодиться в образовательных учреждениях.

Программа была разработана на языке C++ в объектно-ориентированной среде программирования Visual Studio Code.



#### Список использованных источников:

1. Ростовцев А.Г., Маховенко Е.Б. Введение в теорию итерированных шифров.- СПб.:Мир и семья,2003. - 302 с.
2. Уэзерелл Ч. Этюды для программистов. - М.: Мир, 1982. - 288 с.
3. Шнайер Б. Подстановочные шифры // Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке С++ = Applied Cryptography. Protocols, Algorithms and Source Code in C. — М.: Триумф, 2002. — С. 25—26. — 816 с. — 3000 экз. — ISBN 5-89392-055-
4. статья: Чехова М.Р.: Кодирование информации. <http://festival.1september.ru/articles/213248/>
5. Microsoft Corporation. Разработка приложений на Microsoft Visual С++ 6.0. Учебный курс: Официальное пособие Microsoft для самостоятельной подготовки / Пер. с англ.- М.: Издательско-торговый дом «Русская Редакция», 2000. – 576 стр.: илл.
6. С/С++. Программирование на языке высокого уровня / Т.А. Павловская — СПб.: Питер, 2011 .
7. Абельсон Х., Сассман Д. Структура и интерпретация компьютерных программ. — М.: Добросвет, 2010.
8. Архангельский А.Я. Программирование в С++ Builder 5 (или 6). – М.: ЗАО «Издательство БИНОМ», 2002 г. – 1152 с.
9. Архангельский А.Я., Тагин М.А. Приемы программирования в С++ Builder. Механизмы Windows, сети. – М.: ООО «Бином-Пресс», 2004 г. – 656 с.

Приложение А –  
Листинг программы  
(Обязательное)

```

class Transposition
{
    private int[] key = null;

    public void SetKey(int[] _key)
    {
        key = new int[_key.Length];

        for (int i = 0; i < _key.Length; i++)
            key[i] = _key[i];
    }

    public void SetKey(string[] _key)
    {
        key = new int[_key.Length];

        for (int i = 0; i < _key.Length; i++)
            key[i] = Convert.ToInt32(_key[i]);
    }

    public void SetKey(string _key)
    {
        SetKey(_key.Split(' '));
    }

    public string Encrypt(string input)
    {
        for (int i = 0; i < input.Length % key.Length; i++)
            input += input[i];

        string result = "";

        for (int i = 0; i < input.Length; i += key.Length)
        {
            char[] transposition = new char[key.Length];

            for (int j = 0; j < key.Length; j++)
                transposition[key[j] - 1] = input[i + j];

            for (int j = 0; j < key.Length; j++)
                result += transposition[j];
        }

        return result;
    }
}

```

```
}  
  
public string Decrypt(string input)  
{  
    string result = "";  
  
    for (int i = 0; i < input.Length; i += key.Length)  
    {  
        char[] transposition = new char[key.Length];  
  
        for (int j = 0; j < key.Length; j++)  
            transposition[j] = input[i + key[j] - 1];  
  
        for (int j = 0; j < key.Length; j++)  
            result += transposition[j];  
    }  
  
    return result;  
}
```

Приложение Б  
Интерфейс программы  
(Обязательное)

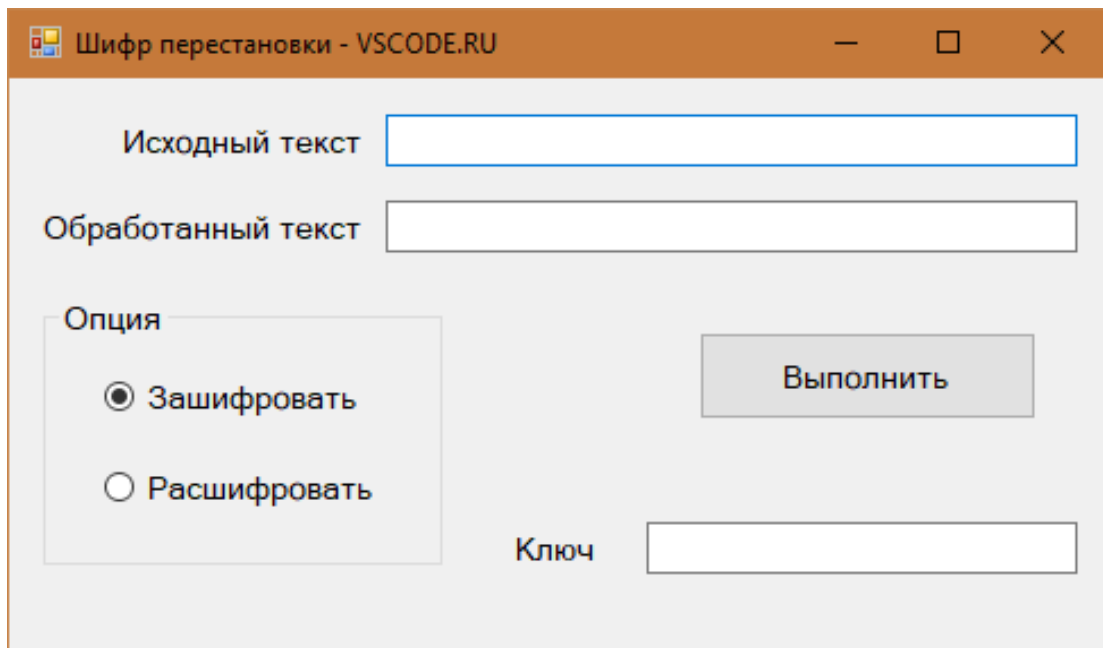


Рисунок Б1 - Интерфейс программы

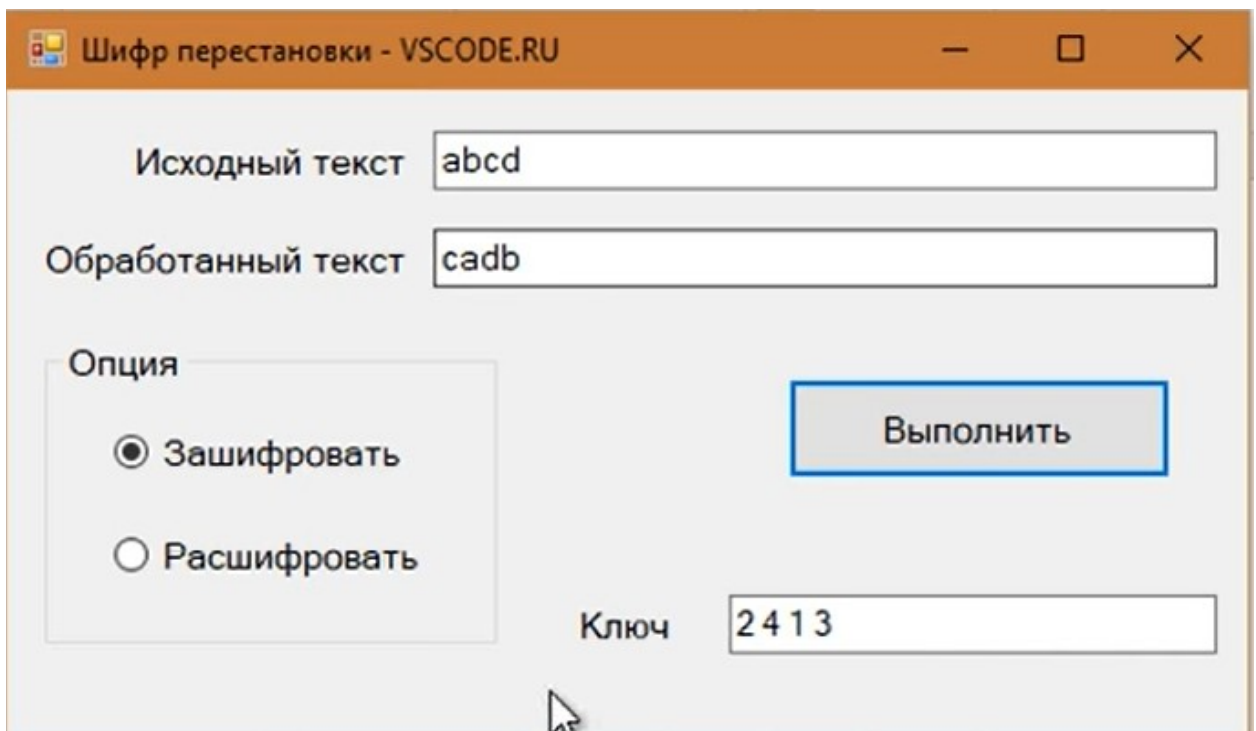


Рисунок Б2- Пример решения

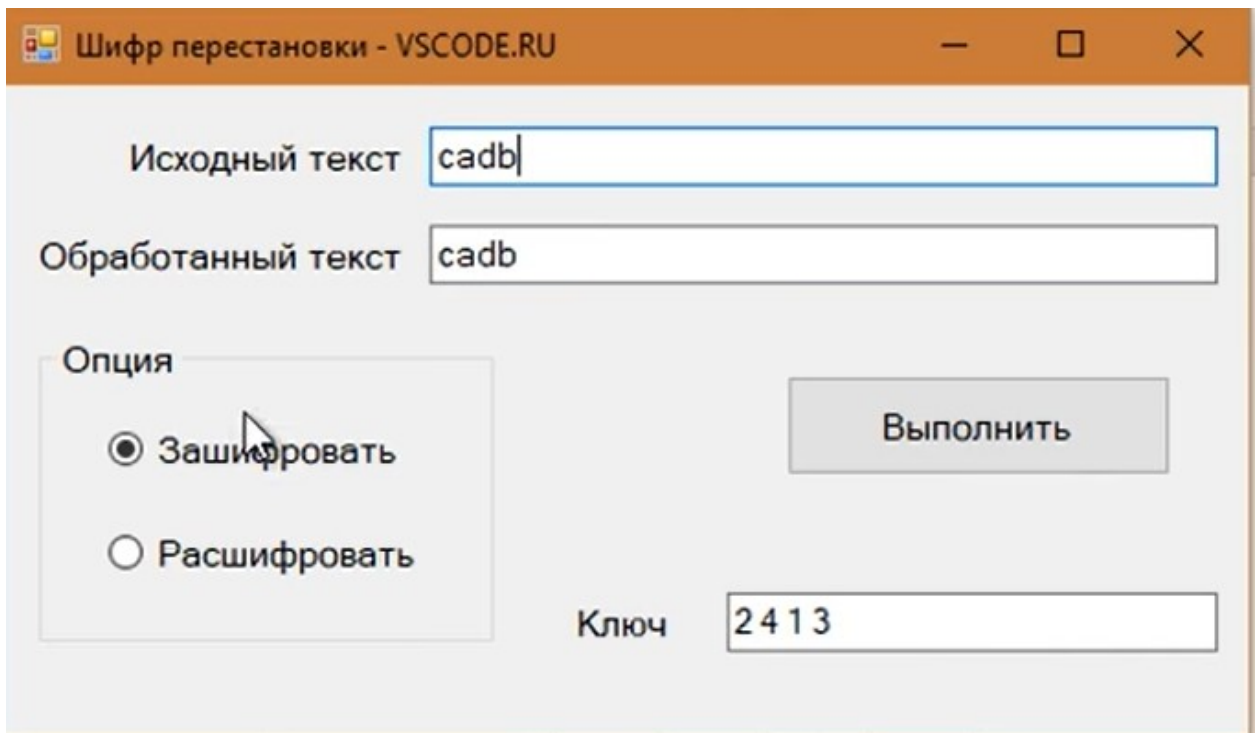


Рисунок Б3 Пример решения

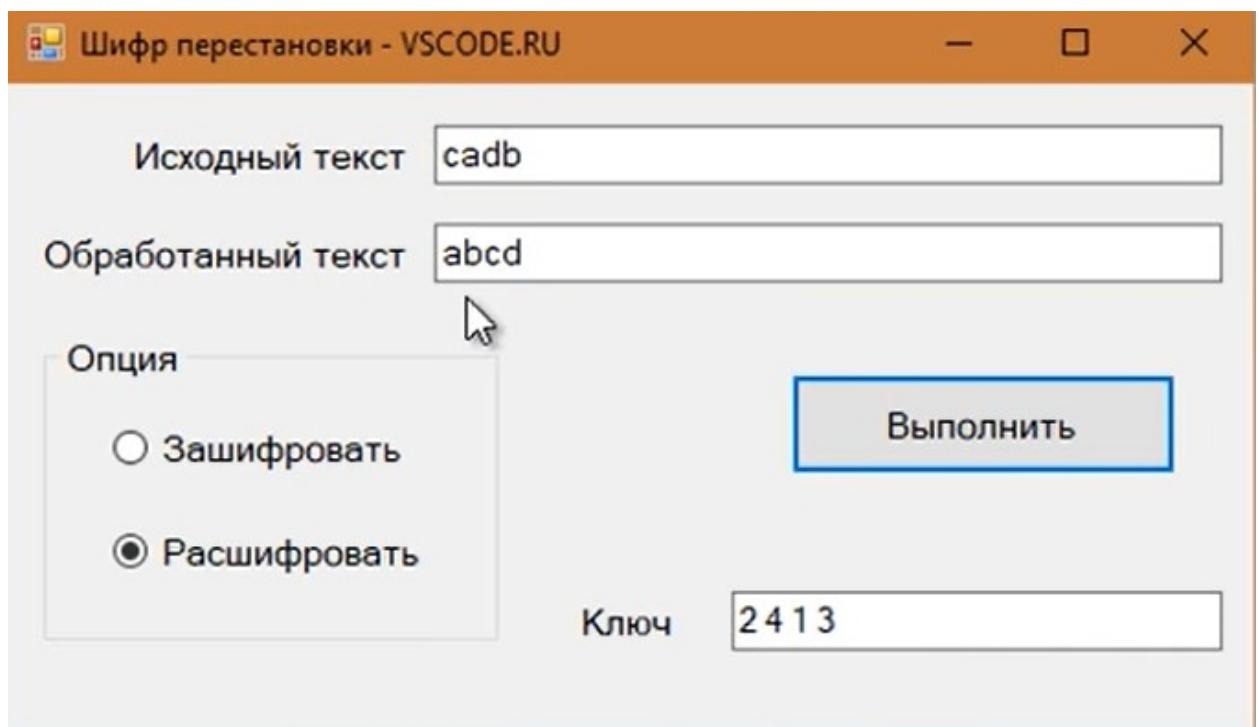


Рисунок Б4 Пример решения

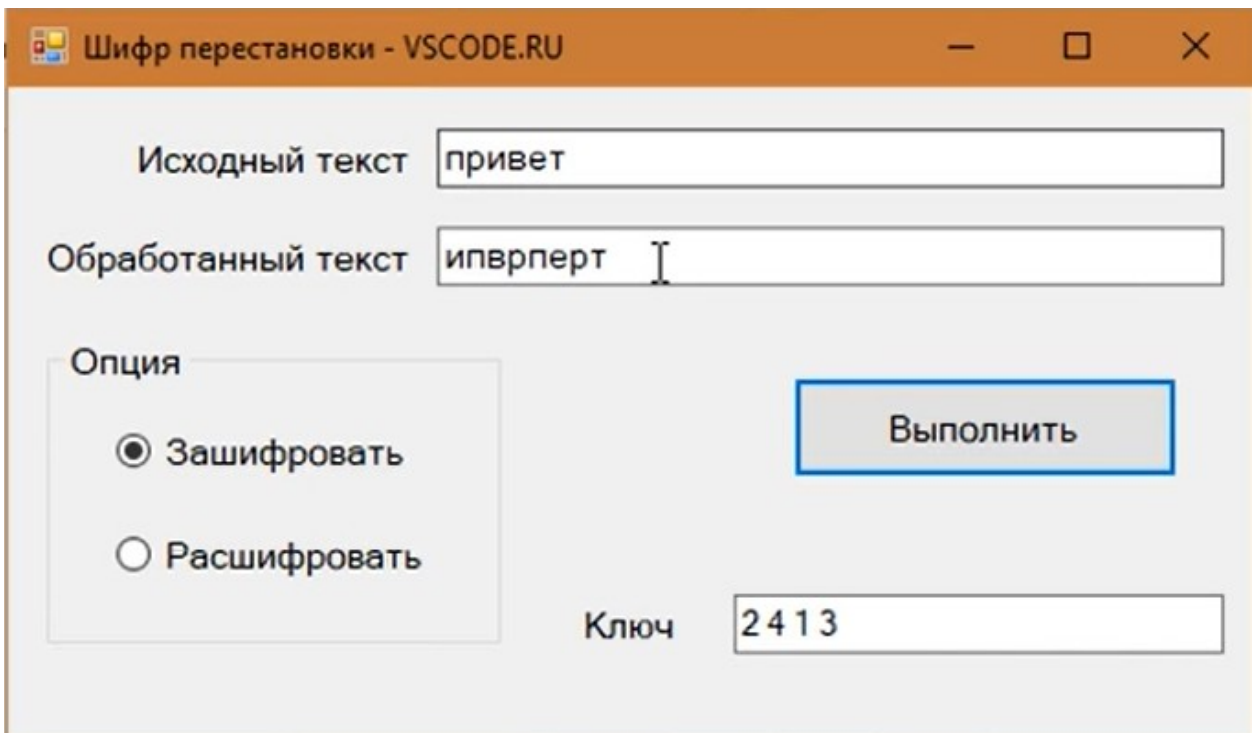


Рисунок Б5 Пример решения

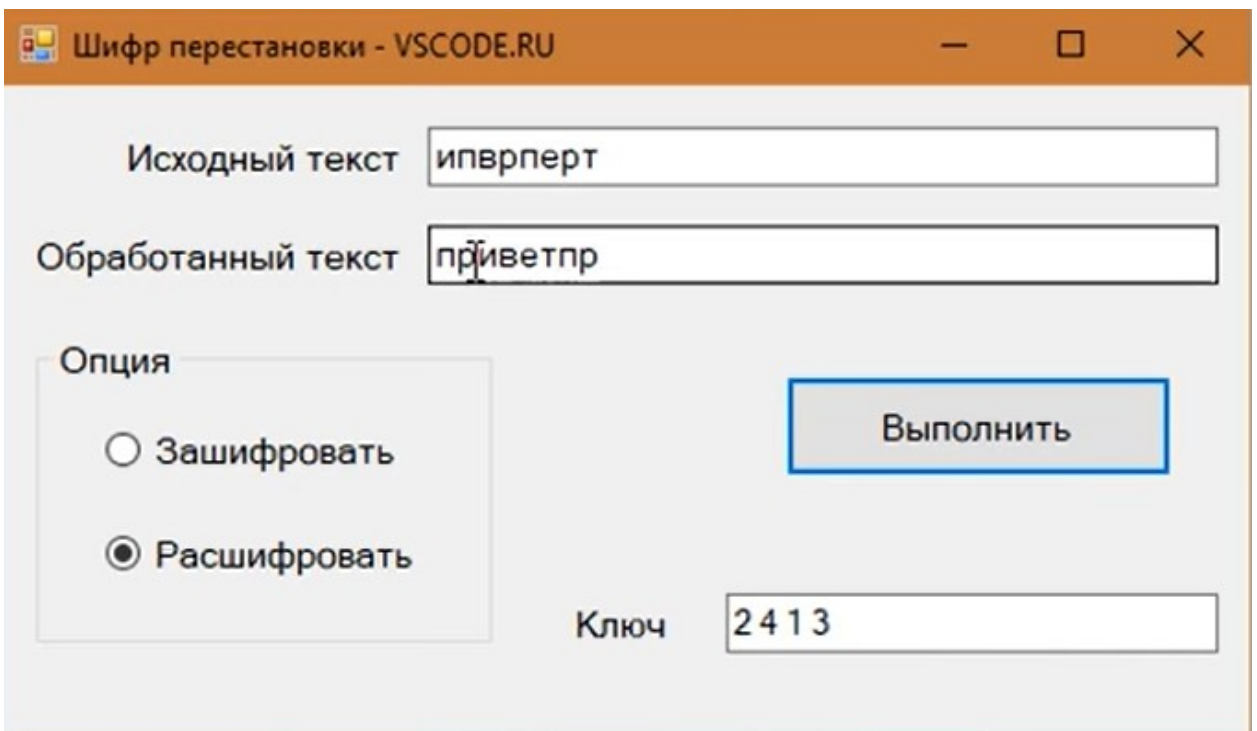


Рисунок Б6 Пример решения