ВЫСШЕЕ УЧЕБНОЕ ЗАВЕДЕНИЕ УКООПСОЮЗА
«ПОЛТАВСКИЙ УНИВЕРСИТЕТ ЭКОНОМИКИ И ТОРГОВЛИ»
УЧЕБНО-НАУЧНЫЙ ИНСТИТУТ
МЕЖДУНАРОДНОГО ОБРАЗОВАНИЯ
ФОРМА ОБУЧЕНИЯ ДНЕВНАЯ

КАФЕДРА КОМПЬЮТЕРНЫХ НАУК
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Допускается к защите
Заведующая кафедрой _____ Е. ОЛЬХОВСКАЯ
(подпись)
«____» _____ 2021 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К МАГИСТЕРСКОЙ РАБОТЕ

на тему:

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ТЕМУ «КОДИРОВАНИЕ ТЕКСТОВ ШИФРАМИ ЦЕЗАРЯ И ТРИСЕМИУСА»

по специальности 122 «Компьютерные науки»

Исполнитель работы Рзаев Эмин Фархад оглы

_____ «__» ___ 2021 г.
(подпись)

Научный руководитель к.ф.-м.н., доц. Парфенова Татьяна Александровна

_____ «__» ___ 2021 г.
(подпись)

Полтава – 2021 г.

РЕФЕРАТ

Записка: 50 стр., 33 рис., 3 табл., 2 приложения (на 28 страницах), 8 источников.

Объект разработки – программа шифрования текстов.

Предмет разработки – программа шифрования и расшифровывания русскоязычных текстов, кодированных шифрами Цезаря и Трисемиуса.

Цель работы – создание программы, которая шифрует текстовую информацию, используя шифры Цезаря и Трисемиуса, и расшифровывает криптограммы, кодированные обоими шифрами.

Методы разработки – среда программирования Delphi, язык программирования Object Pascal.

Исследована тема «Кодирование текстовой информации шифрами Цезаря и Трисемиуса».

Осуществлен обзор и анализ программ сходной тематики.

Для двух шифров созданы алгоритмы шифрование и дешифрования, для одного алгоритма – блок-схема.

Созданы две программы. Программы протестированы.

Ключевые слова: ШИФРОВАНИЕ ТЕКСТА, КОДИРОВАНИЕ, ШИФР ЦЕЗАРЯ, ШИФР ТРИСЕМИУСА.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ, СОКРАЩЕНИЙ И ТЕРМИНОВ

Условные обозначения, символы, сокращения, термины	**Объяснение условных обозначений, символов, сокращений, терминов**
DES, стандарт шифрования DES	DES (англ. data encryption standard) – алгоритм для симметричного шифрования, разработанный фирмой IBM и утверждённый правительством США в 1977 г. как официальный стандарт (FIPS 46-3).
Гонсфельд	Граф Гронсфельд – руководитель первой дешифровальной службы Германии в XVII веке. Шифр Гонсфельда назван в его честь.
Трисемиус	Иоганн Тритемий (лат. Трисемиус) – немецкий гуманист, генеалог, монастырский историк и библиофил. Шифр Трисемиуса назван в его честь.
Файстель	Хорст Файстель – учёный-криптограф, который работал над разработкой алгоритмов шифрования в компании IBM, один из основателей современной криптографии как науки. Заложил основы создания алгоритма шифрования DES. Шифр Файстеля назван в его честь.

Фано	Ро́берт Ма́рио Фа́но – итальяно-американский учёный в области информатики, профессор Массачусетского технологического института, член Национальной академии наук США и Национальной инженерной академии США. Шифр Шеннона-Фано назван в его честь.
Цезарь	Гай Юлий Цезарь – древнеримский государственный и политический деятель, полководец, писатель. Шифр Цезаря назван в его честь.
Шеннон	Клод Э́лвуд Ше́ннон – американский инженер, криптоаналитик и математик. Шифр Шеннона-Фано назван в его честь.

ВСТУПЛЕНИЕ

Актуальность. В мире цифровых технологий защита информации стала актуальным вопросом. Любая информация, которую пользователи передают с помощью электронной почты, мессинджеров Viber, Telegram, WhatApps, размещают в базах данных или социальных сетях, должна быть защищена. В первую очередь это касается персональных данных, конфиденциальных сообщений, банковской информации, финансовых транзакций, корпоративной информации. Одним из методов защиты информации является ее шифрование.

Цель магистерской работы – создать программу, которая шифрует текстовую информацию, используя шифры Цезаря и Трисемиуса, и расшифровывает шифровки, кодированные обоими шифрами.

Задачи работы – ознакомиться с некоторыми методами шифрования информации, в частности, с шифрами Цезаря и Трисемиуса. Осуществить обзор программных продуктов, созданных IT-студентами ПУЭТ. Изложить материал по шифрам Цезаря и Трисемиуса. Сделать алгоритмизацию этих методов и создать программы. Описать полученные результаты.

Объект разработки – программа шифрования текстов.

Предмет разработки – программа шифрования и расшифровывания русских текстов, кодированных шифрами Цезаря и Трисемиуса.

Методы разработки – для создания программы использовалась среда программирования `Delphi` и язык программирования `Object Pascal`.

Структура пояснительная записки.

Пояснительная записка состоит из трех частей.

В первой части содержится постановка задачи.

Во второй части осуществлён обзор программ, созданных IT-студентами ПУЭТ, который касаются темы шифрования.

В третьей части описаны принципы шифрования русских текстов, изложено, в чем состоят шифры Цезаря и Трисемиуса, приведены примеры, сформулированы алгоритмы программы, поданы блок-схемы алгоритмов шифрования.

В четвертой части изложена инструкция по работе с программой, описано, как создавался программный продукт, показаны результаты тестирования программы.

1. ПОСТАНОВКА ЗАДАЧИ

Во время дипломного проектирования необходимо ознакомится с методами шифрования текстовой информации.

Обратить пристальное внимание на шифрование русских текстов шифрами Цезаря и Трисемиуса.

Изучить правила шифрования и дешифрования русских текстов шифрами Цезаря и Трисемиуса. Изучить вопрос, насколько сложно взломать такие шифровки.

Изложить правила шифрования этими кодировками в пояснительной записке.

Создать алгоритмы шифрования и дешифрования текстов кодами Цезаря и Трисемиуса.

Нарисовать блок-схемы алгоритмов.

Написать программу, которая реализует кодирование и декодирование сообщений, написанных русским языком, шифрами Цезаря и Трисемиуса. Предусмотреть в программе удобный и понятный интерфейс.

Протестировать программу.

Создать инструкцию по работе с программой. Описать процесс создания программы.

2. ИНФОРМАЦИОННЫЙ ОБЗОР

2.1. Обзор программ, связанных с кодированием и защитой информации

Рассмотрим программные продукты, в которых освещается тема шифрования информации [2-4].

1. В 2018 г. студентом кафедры математического моделирования и социальной информатики ПУЭТ Мусаевым С.Э. создан тренажер по методу Шеннона-Фано (рис. 2.1).

Рисунок 2.1 – Стартовое окно тренажера «Метод Шеннона-Фано»

Программа представляет собой тренажер, т.е. продукт, который обучает учащихся представленной кодировке.

На рисунках 2.1-2.6 поданы несколько шагов тренажера.

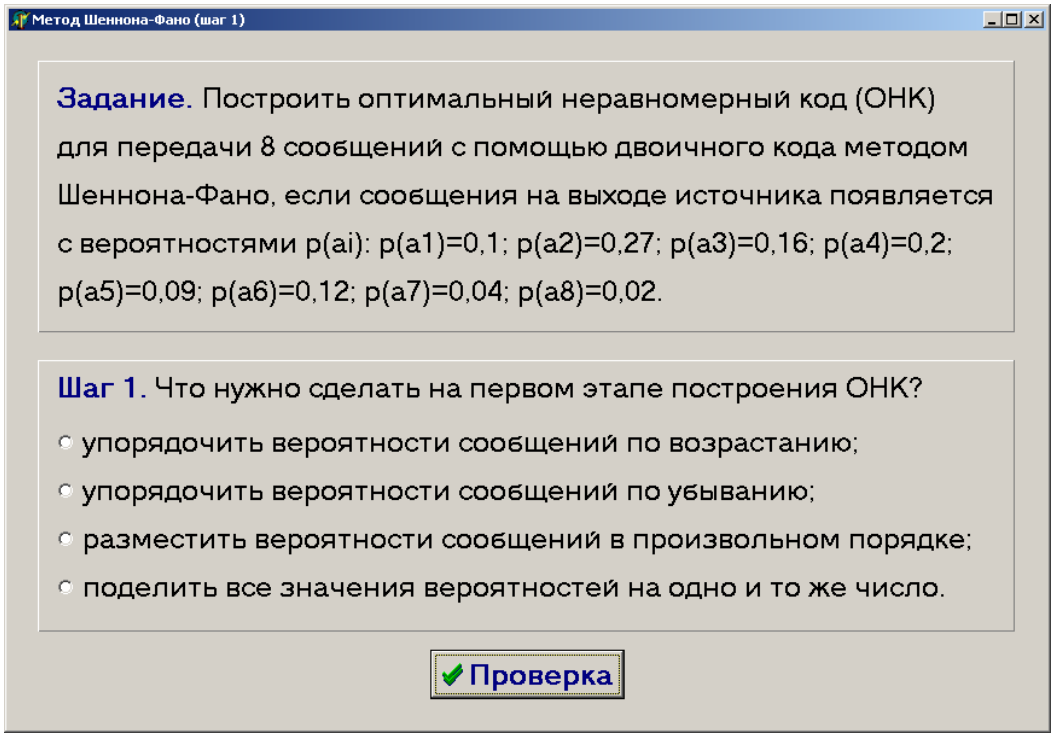

Рисунок 2.2 – Первый шаг

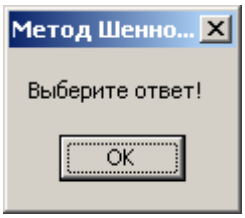

Рисунок 2.3 – Сообщение об ошибке

2. В 2019 г. студенткой той же кафедры Гарбуз И. В. предложен тренажер по шифру Файстеля и стандарту шифрования DES (рис. 2.7).

На рисунках 2.8-2.9 представлено несколько шагов тренажера.

3. В 2020 г. студентом Николаенком А. В. [4] создана программа «Кодирование текстов шифром Гонсфельда».

На рисунках 2.10-2.11 отображено, как программный продукт работает.

2.2. Положительные аспекты рассмотренных программ

Тренажеры проработаны детально и скрупулёзно, и позволяют студентам проработать темы.

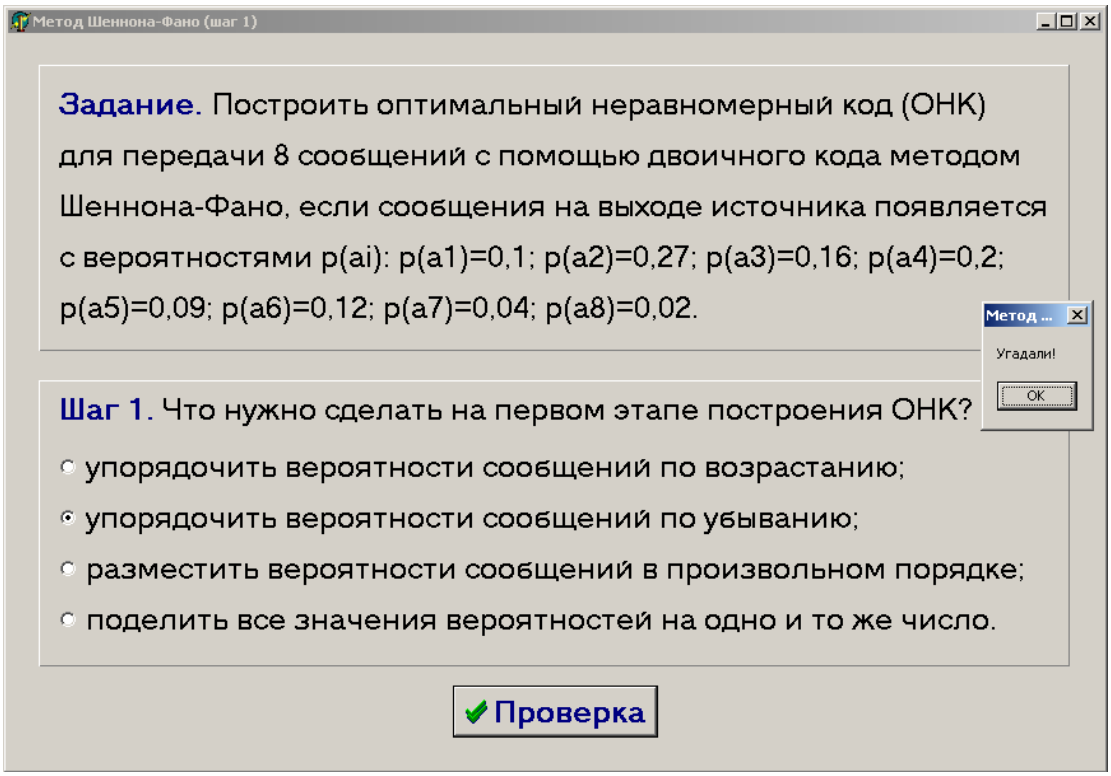

Рисунок 2.4 – Правильный ответ

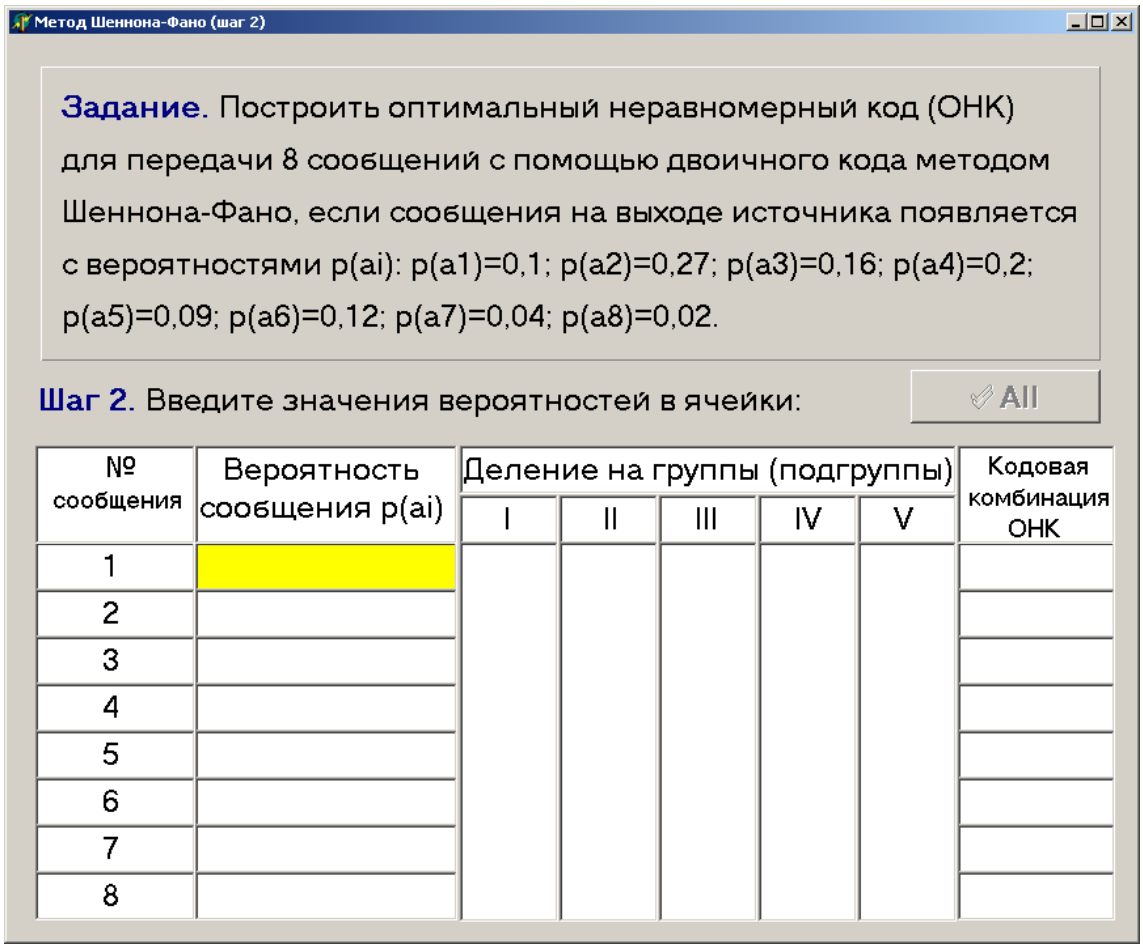

Рисунок 2.5 – Второй шаг тренажера

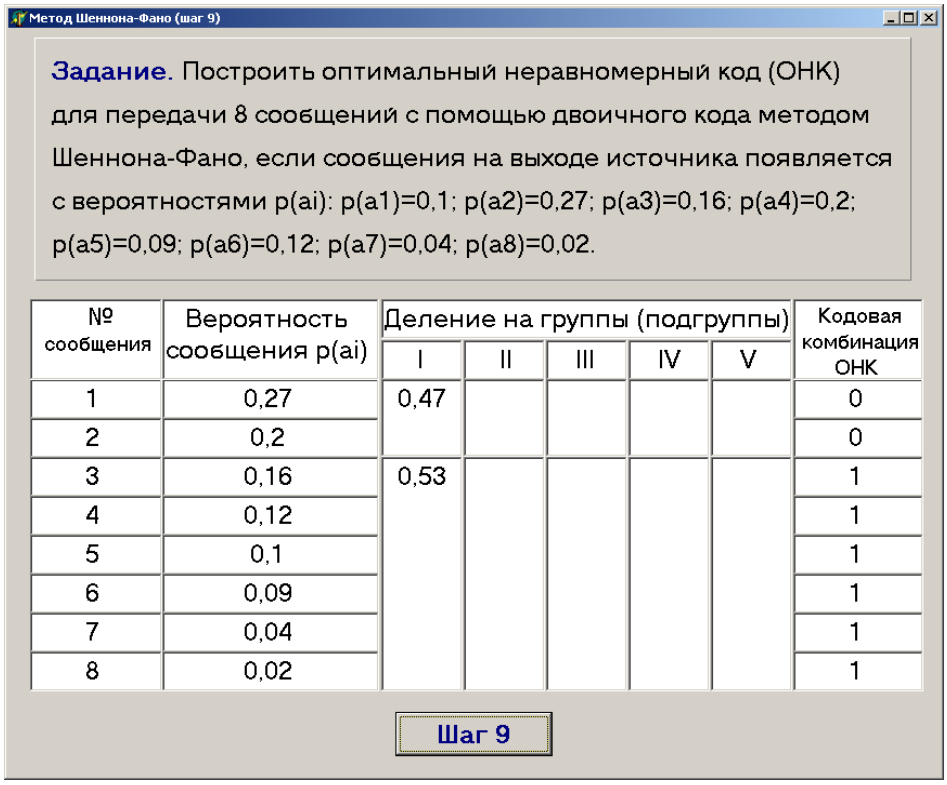

Рисунок 2.6 –Девятый шаг тренажера

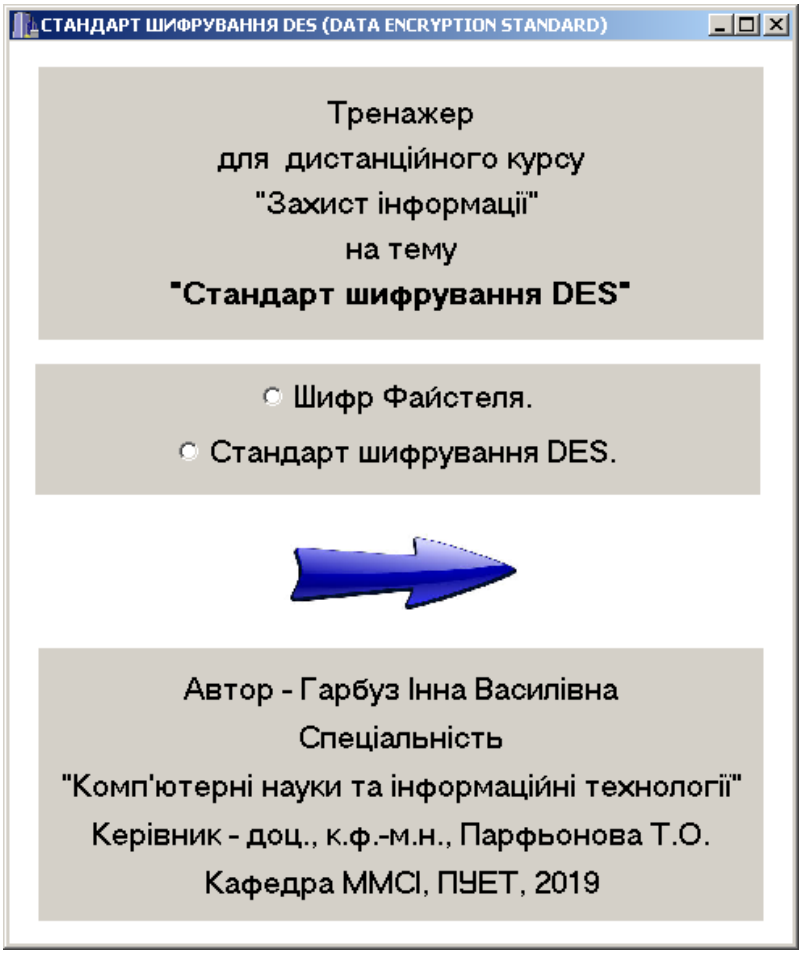

Рисунок 2.7 – Тренажер «Стандарт шифрования DES»

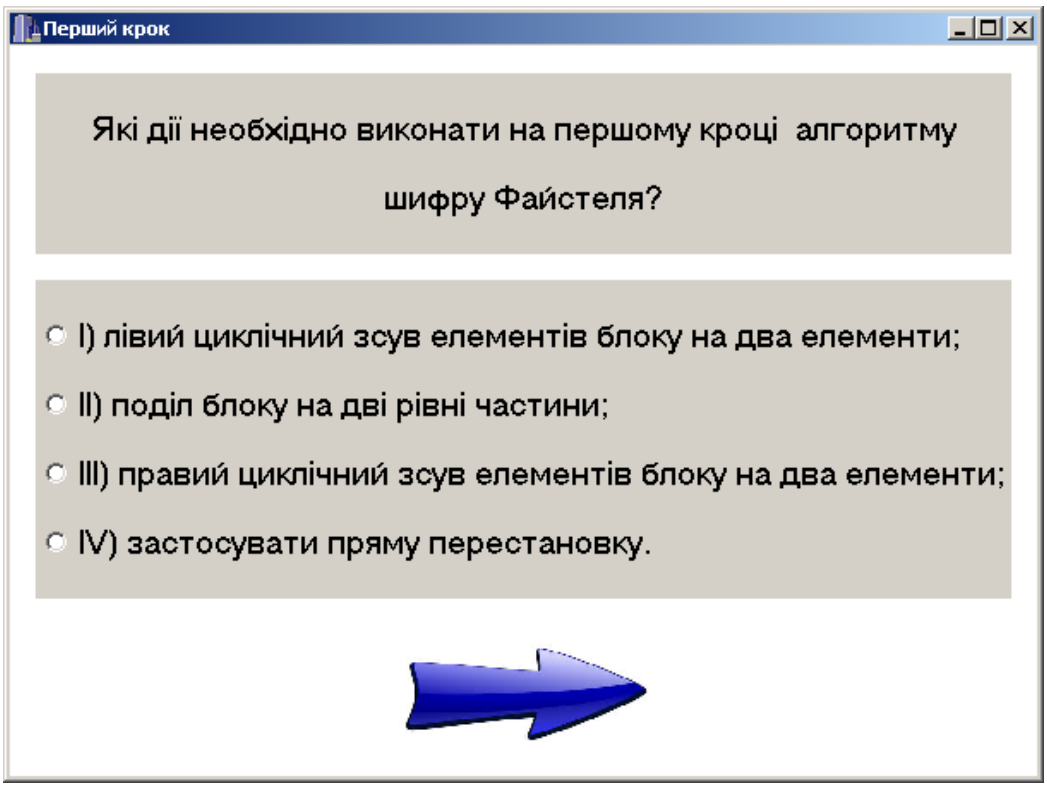

Рисунок 2.8 – Первый вопрос по теме «Шифр Файстеля»

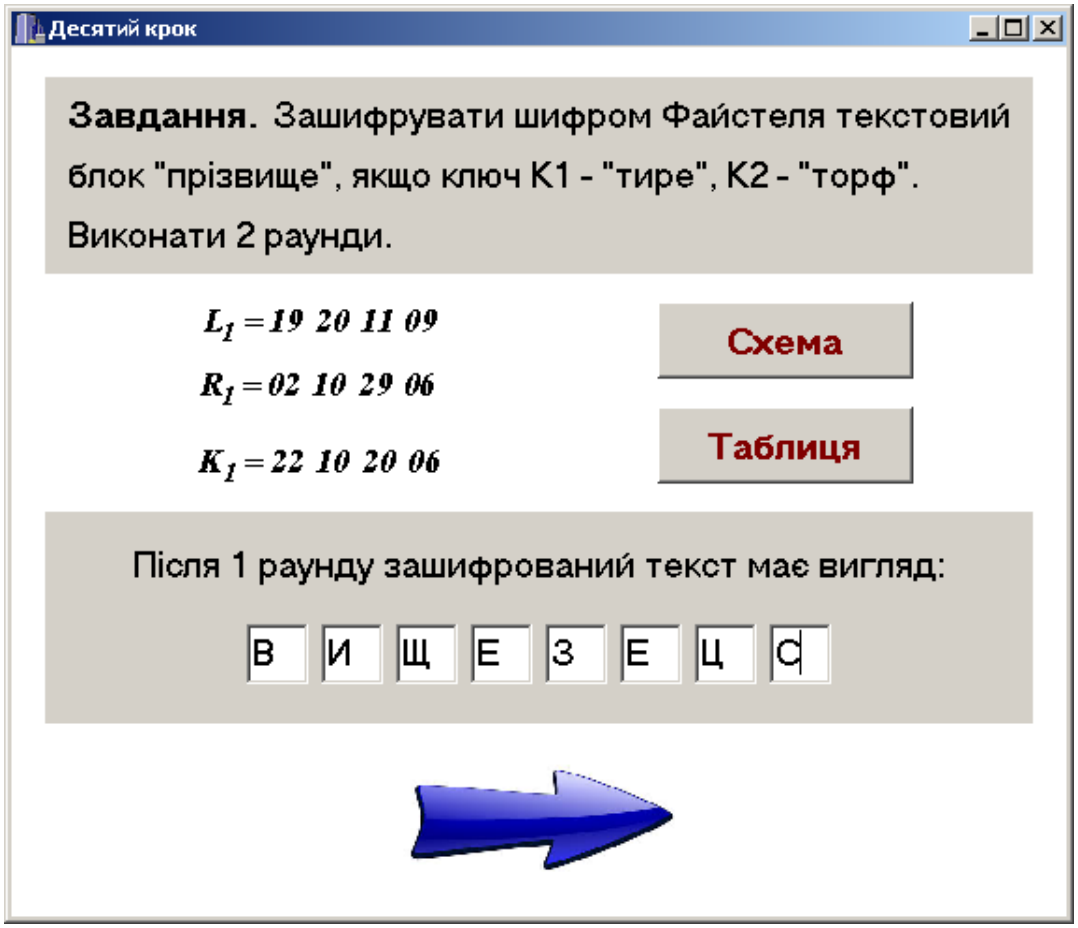

Рисунок 2.9 – Десятый вопрос по теме «Шифр Файстеля»

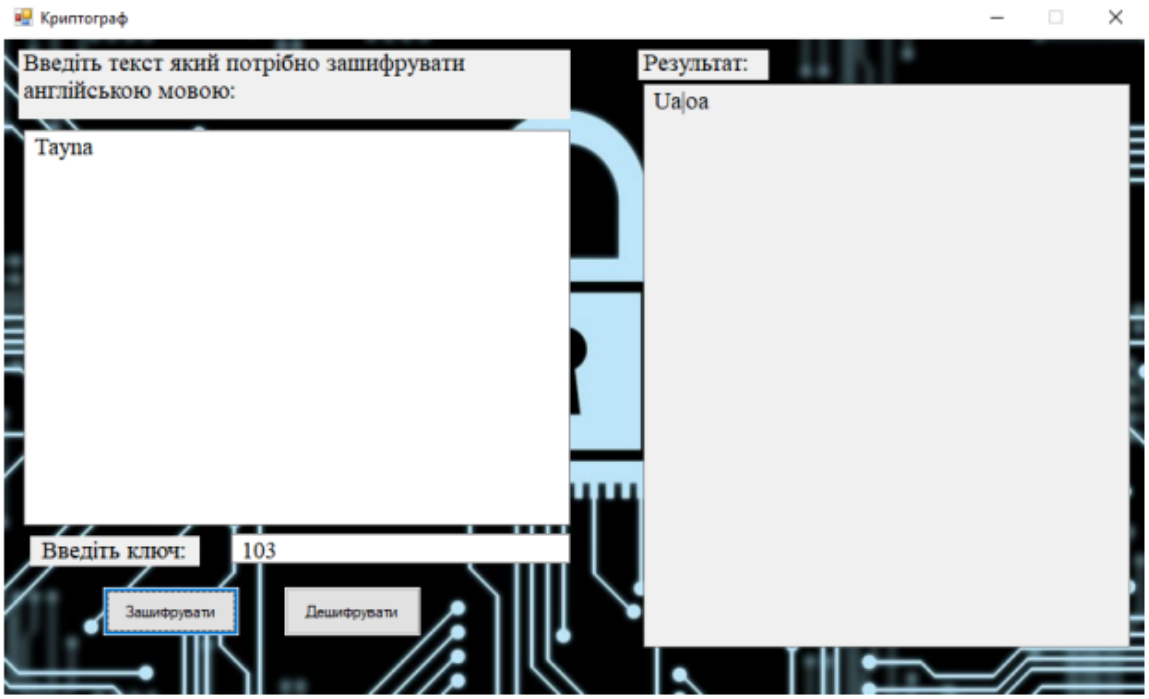

Рисунок 2.10 – Шифрование слова «Tayna»

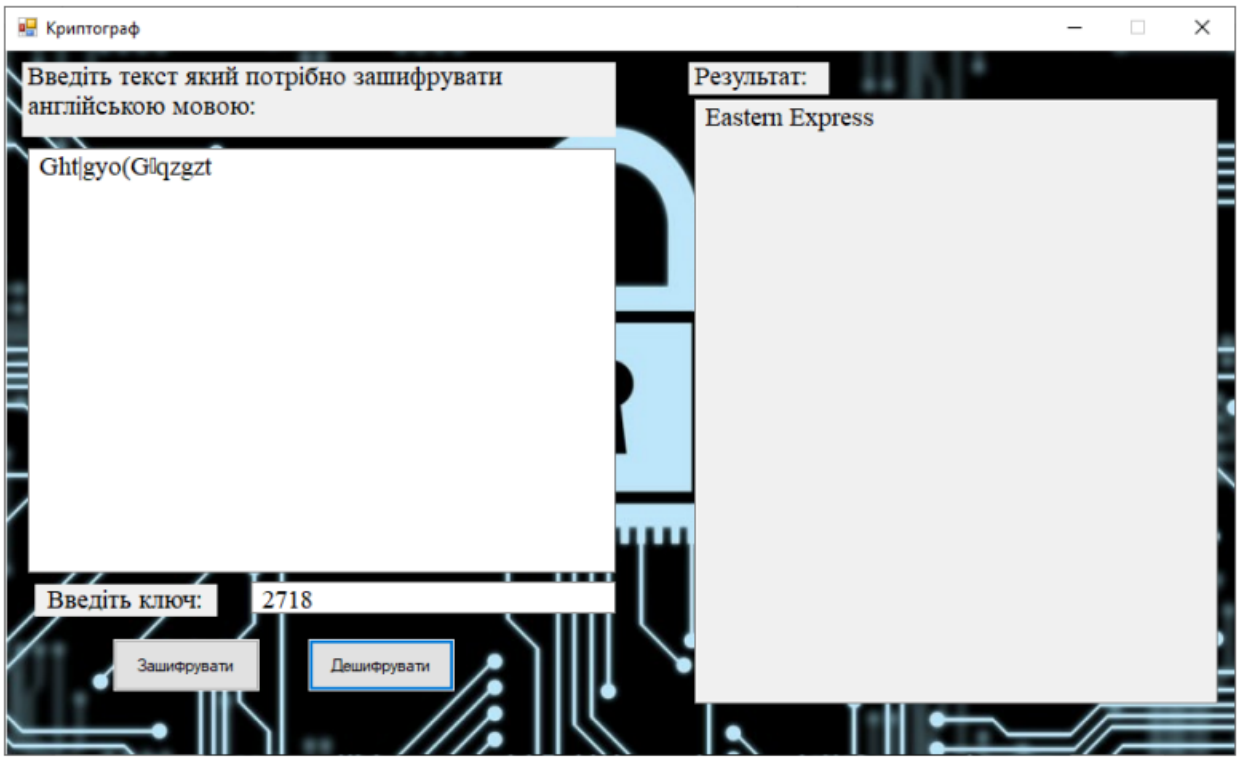

Рисунок 2.11 – Шифрование-дешифрование

Программный продукт Николаенка дает возможность проверить работу метода Генсфольда.

2.3. Негативные аспекты рассмотренных программ

Тренажеры и программа созданы как десктопные продукты, целесообразно адаптировать их под использование смартфоном, планшетом.

Целесообразно, чтобы данные продукты были в свободном для студентов доступе: в сети интернет, или в дистанционном курсе. Сейчас такого доступа нет.

Целесообразно иметь русские версии тренажёров и программ для обучения иностранных студентов.

Удобным для русскоязычных иностранных студентов было бы использование в алгоритмах кодировки русского алфавита, а не украинского или латинского.

2.4. Необходимость и актуальность темы

Как видим, на нашей кафедре уже есть ряд разработок по данной тематике.

Поскольку, любая цифровая информация должна быть зашифрована для обеспечения конфиденциальности, то изучение алгоритмов шифрования и защиты информации, их сферы использования, преимуществ и недостатков является актуальной задачей для ИТ-специалистов.

Так как алгоритмов существует много, то создание и усовершенствование программных продуктов по этой теме продолжает оставаться актуальным.

3. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

3.1. Правила шифрования русских текстов

При ручном шифровании русских текстов принято сокращать русский алфавит до тридцати букв. При этом знаки «ё» и «е», «й» и «и», «ь» и «ъ» шифруются одинаково [2].

Из текста шифровки удаляют все пробелы и знаки пунктуации. Это связано с тем, что такой процесс усложняет взлом кода.

Кроме того, тексты шифровок принято разбивать на блоки длиной в пять букв для облегчения расшифровывания и уменьшения количества ошибок шифровальщиков.

3.2. Шифр Цезаря

3.2.1. Общие сведения

Рассмотрим метод шифрования текстов, названный **шифром Цезаря** [2].

Этот тип шифровки текстовой информации появился во времена Римской империи. Возникновение шифра связано с тем, что главы провинций и их подчиненные обменивались сообщениями, и для того, чтоб информация не попала к несанкционированным лицам, донесения нужно было шифровать.

Шифр Цезаря состоит в том, что буквы в словах меняются. Каждый раз необходимо использовать вместо текущей буквы ту, которая идет в алфавите четвертой по счету. То есть, следует взять букву «Г» вместо буквы «А», букву «Д» вместо «Б» и т. д. То есть, каждая буква в зашифрованном сообщении меняется на другую, стоящую в алфавите позже от нее на четыре позиции.

Таким образом, шифр Цезаря для сокращенного русского алфавита выглядит так, как показано в таблице 3.1.

Табл. 3.1 – Шифр Цезаря для сокращенного русского алфавита

№	Буква русского алфавита	Буква в шифровке
1	А	Г
2	Б	Д
3	В	Е
4	Г	Ж
5	Д	З
6	Е	И
7	Ж	К
8	З	Л
9	И	М
10	К	Н
11	Л	О
12	М	П
13	Н	Р
14	О	С
15	П	Т
16	Р	У
17	С	Ф
18	Т	Х
19	У	Ц
20	Ф	Ч
21	Х	Ш
22	Ц	Щ
23	Ч	Ь
24	Ш	Ы

Продолжение табл. 3.1 – Шифр Цезаря для сокращенного русского алфавита

№	Буква русского алфавита	Буква в шифровке
25	Щ	Э
26	Ь	Ю
27	Ы	Я
28	Э	А
29	Ю	Б
30	Я	В

При расшифровывании информации поступают наоборот: берут букву, стоящую на четыре позиции выше в алфавите от текущей.

Пример 3.2.1.

Используя шифр Цезаря, зашифровать текст «ИНФОРМАЦИЯ».

Разбиваем слово на блоки по пять букв: «ИНФОР МАЦИЯ»

Первый блок: букве «И» в шифровке соответствует буква «М», «Н» – «Р», «Ф» – «Ч», «О» – «С», «Р» – «У».

Второй блок: букве «М» в шифровке соответствует буква «П», «А» – «Г», «Ц» – «Щ», «И» – «М», «Я» – «В».

Получили зашифрованное сообщение «МРЧСУ ПГЩМВ».

Пример 3.2.2.

Используя шифр Цезаря, расшифровать текст «НСПТЮ БХИУ».

Смотрим аналог каждой буквы в таблице 3.1

Первый блок: букве «Н» соответствует «К», «С» – «О», «П» – «М», «Т» – «П», «Ю» – «Ь».

Второй блок: букве «Б» соответствует «Ю», «Х» – «Т», «И» – «Е», «У» – «Р».

Получили дешифрованное сообщение «КОМПЬЮТЕР».

3.2.2. Криптоанализ шифра Цезаря

Негативной характеристикой шифра Цезаря является то, что количество возможных вариантов на единицу меньше, чем букв в алфавите, то есть 29. То есть, шифр относительно легко взломать [2].

3.2.3. Алгоритм шифра Цезаря

Составим алгоритм программы, которая будет кодировать сообщения и раскодировать их, используя шифр Цезаря.

Алгоритм шифрования

1. Берем первые пять букв сообщения исходного сообщения.
2. Каждый символ в пятерке заменяем согласно таблице 3.1.
3. Ставим пробел в шифровке.
4. Повторяем пункты 1-2 до тех пор, пока не достигнем конца текста.

Алгоритм дешифрования

1. Берем первый блок з пяти букв шифровки.
2. Каждый символ в блоке шифровки заменяем согласно таблице 3.1.
3. Повторяем пункты 1-2 до тех пор, пока не достигнем конца шифровки.

3.2.4. Блок-схема алгоритма шифра Цезаря

На рисунке 3.1 показана блок-схема алгоритма зашифровывания текста.

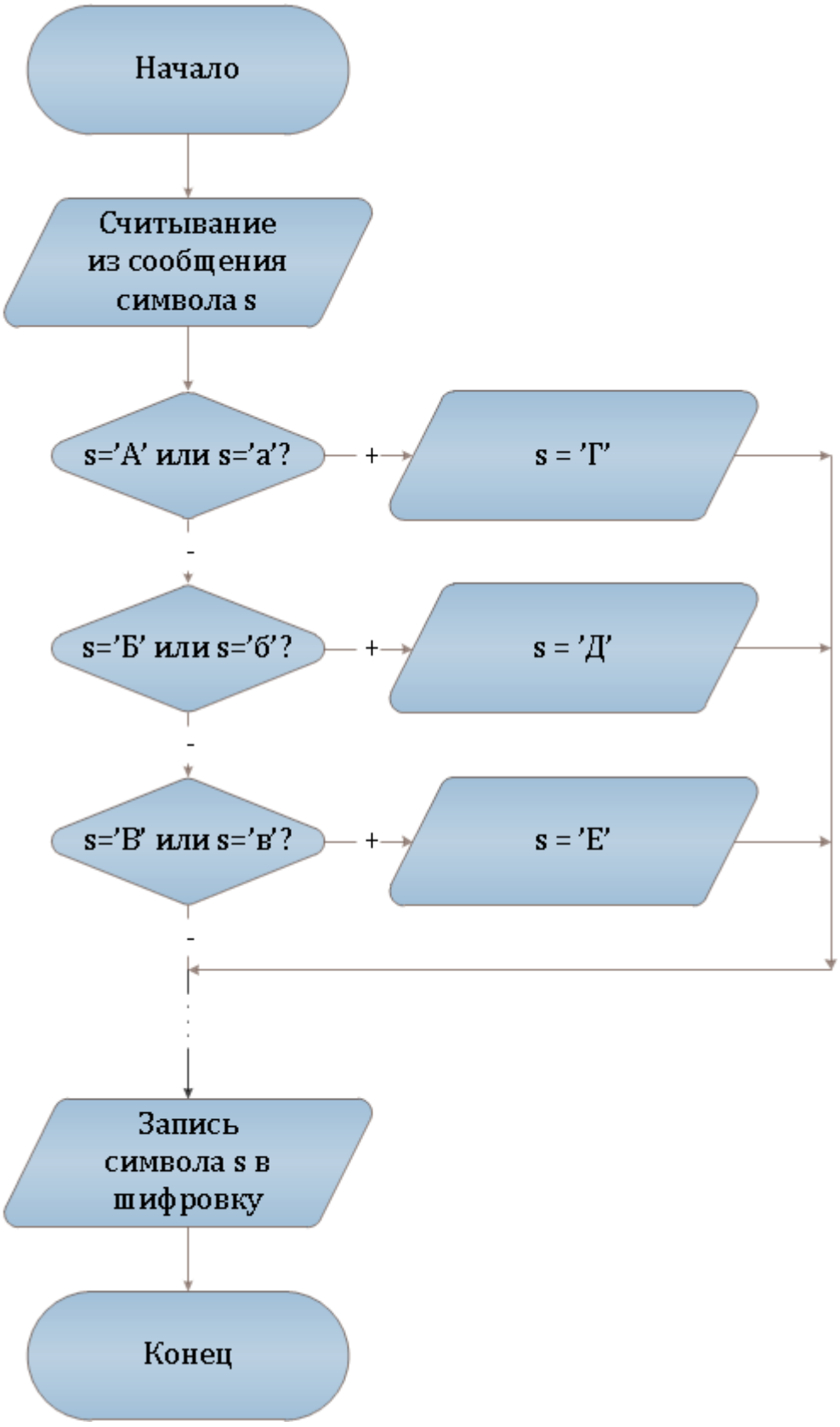

Рисунок 3.1 – Блок-схема шифра Цезаря

3.3. Шифр Трисемиуса

3.3.1. Общие сведения

Немецкий исследователь Иоганн Трисемиус модернизировал изыскания греческих исследователей и создал кодировку под названием «шифр Трисемиуса» [2]. Создавалась шифрующая таблица, заполненная буквами алфавита случайным образом. Для получения кодировки использовали, как правило, специальное (ключевое) слово или фразу и таблицу. Для русского языка таблица может, например, состоять из шести строк и пяти столбцов. Специальное (ключевое) слово записывают в таблицу строками, откидывая повторяющиеся буквы, а потом дописывают в алфавитном порядке те буквы алфавита, которые отсутствовали в специальном слове или фразе.

Такой поход позволяет быстро и легко запомнить шифрующую таблицу. Например, для специального слова «РЕСПУБЛИКА» шифрующая таблица принимает вид (рис. 3.2):

Р	Е	С	П	У
Б	Л	И	К	А
В	Г	Д	Ж	З
М	Н	О	Т	Ф
Х	Ц	Ч	Ш	Щ
Ь	Ы	Э	Ю	Я

Рисунок 3.2 – Шифровальная таблица для ключа «РЕСПУБЛИКА»

Для зашифровывания текста в таблице находят букву из текста и меняют ее на букву из того же столбца, но из рядка, который идет ниже. Если буква находится в последней строке, то для зашифровывания берут букву из первой строки и того же самого столбца.

При расшифровывании текста поступают противоположным образом: ищут букву на строку выше, чем буква в зашифрованном сообщении.

Пример 3.3.1.

Используя шифр Трисемиуса, зашифровать текст «ИНФОРМАЦИЯ», используя шифрующую таблицу, поданную на рис. 3.2.

Разбиваем слово на блоки по пять букв: «ИНФОР МАЦИЯ»

Первый блок:

Букву «И» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Д».

Букву «Н» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Ц».

Букву «Ф» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Щ».

Букву «О» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Ч».

Букву «Р» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Б».

Второй блок:

Букву «М» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Х».

Букву «А» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «З».

Букву «Ц» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Ы».

Букву «И» меняем на букву из того же столбца, но размещённую на строку ниже, получаем букву «Д».

Букву «Я» меняем на букву «У», так как буква «Я» стоит в последней строке, а для последней строки берем ее заменитель из первой строки того же столбца.

Получили зашифрованный текст «ДЦЩЧБ ХЗЫДУ».

Пример 3.3.2.

Используя шифр Трисемиуса, расшифровать текст «КБЧНБ ЗХХДБ ЧМЗЦД Л», используя шифрующую таблицу, поданную на рис. 3.2.

Первый блок:

Букву «К» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «П».

Букву «Б» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «Р».

Букву «Ч» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «О».

Букву «Н» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «Г».

Букву «Б» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «Р».

Второй блок:

Букву «З» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «А».

Букву «Х» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «М».

Букву «Х» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «М».

Букву «Д» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «И».

Букву «Б» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «Р».

Третий блок:

Букву «Ч» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «О».

Букву «М» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «В».

Букву «З» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «А».

Букву «Ц» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «Н».

Букву «Д» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «И».

Четвертый блок:

Букву «Л» меняем на букву из того же столбца, но размещенную на строку выше. Получаем букву «Е».

Получили дешифрованный текст «ПРОГРАММИРОВАНИЕ».

3.3.2. Криптоанализ шифра Трисемиуса

Такие шифры называют монограммными, поскольку шифрование осуществляется по одной букве [2].

Если алфавит содержит тридцать букв, то существует $30! \approx 2.65 \times 10^{32}$ способов построение шифрующей таблицы. Но любую кодировку можно взломать, если объем зашифрованного текста от двадцати-тридцати букв и выше. Это зависит от особенностей языка. А при объеме криптограммы большего, чем сто букв, взлом не представляет сложности. Объяснение этого состоит в том, что вероятность появления букв можно высчитать.

В таблицах 3.1 и 3.2 представлена вероятность появления буквы в русскоязычном тексте без учета символ пробела и с его учетом [2].

Табл. 3.2 – Вероятность появления буквы в русскоязычном тексте (без учета символа пробела)

№¤	Буква¤	Вероятность появления¤
1¤	А¤	0,07922¤
2¤	Б¤	0,01651¤
3¤	В¤	0,04519¤
4¤	Г¤	0,01799¤
5¤	Д¤	0,02965¤
6¤	Е¤	0,08363¤
7¤	Ж¤	0,00894¤
8¤	З¤	0,01718¤
9¤	И¤	0,06789¤
10¤	Й¤	0,01297¤
11¤	К¤	0,03458¤
12¤	Л¤	0,05028¤
13¤	М¤	0,03147¤
14¤	Н¤	0,06700¤
15¤	О¤	0,10835¤
16¤	П¤	0,02852¤
17¤	Р¤	0,04834¤
18¤	С¤	0,05569¤
19¤	Т¤	0,05527¤
20¤	У¤	0,02909¤
21¤	Ф¤	0,00189¤
22¤	Х¤	0,01060¤
23¤	Ц¤	0,00330¤
24¤	Ч¤	0,01367¤
25¤	Ш¤	0,00971¤
26¤	Щ¤	0,00406¤

Продолжение табл. 3.2 – Вероятность появления буквы в русскоязычном тексте (без учета символа пробела)

№	Буква	Вероятность появления
27	Ъ	0,00026
28	Ы	0,02200
29	Ь	0,01770
30	Э	0,00245
31	Ю	0,00569
32	Я	0,02091

Табл. 3.3 – Вероятность появления буквы в русскоязычном тексте (с учетом символа пробела)

№	Буква	Вероятность появления
1	А	0,063522
2	Б	0,013242
3	В	0,036238
4	Г	0,014428
5	Д	0,023775
6	Е	0,067062
7	Ж	0,007168
8	З	0,013775
9	И	0,054435
10	Й	0,010401
11	К	0,027731
12	Л	0,040318
13	М	0,025238
14	Н	0,053725

Продолжение табл. 3.3 – Вероятность появления буквы в русскоязычном тексте (с учетом символа пробела)

15	О	0,086881
16	П	0,02287
17	Р	0,038758
18	С	0,04432
19	Т	0,04432
20	У	0,023329
21	Ф	0,001519
22	Х	0,008500
23	Ц	0,002647
24	Ч	0,010962
25	Ш	0,00790
26	Щ	0,003257
27	Ъ	0,000206
28	Ы	0,017638
29	Ь	0,014189
30	Э	0,001965
31	Ю	0,004561
32	Я	0,004561
Пробел		0,198128

3.3.3. Алгоритм шифра Трисемиуса

Составим алгоритм программы, которая будет кодировать сообщения и раскодировать их, используя шифр Трисемиуса.

Алгоритм шифрования

1. Берем первые пять букв сообщения исходного сообщения.

2. Каждый символ в пятерке заменяем, используя рисунок 3.2, а именно: смотрим на искомую букву, ищем ее в таблице, заменяем ее на букву, стоящую в том же столбце, но строкой ниже (для букв, стоящих в последней строке, берем буквы из первой строки).

3. Ставим пробел в шифровке.

4. Повторяем пункты 1-2 до тех пор, пока не достигнем конца текста.

Алгоритм дешифрования

1. Берем первый блок з пяти букв шифровки.

2. Каждый символ в блоке шифровки заменяем, используя рисунок 3.2, а именно: смотрим на искомую букву, ищем ее в таблице, заменяем ее на букву, стоящую в том же столбце, но строкой выше (для букв, стоящих в первой строке, берем буквы из последней строки).

3. Повторяем пункты 1-2 до тех пор, пока не достигнем конца шифровки.

4. ПРАКТИЧЕСКАЯ ЧАСТЬ

4.1. Инструкция по работе с программами

Для реализации двух методов было создано две программы.

При запуске программы первой программы – «Шифр Цезаря» – открывается окно как на рис. 4.1. На стартовой форме показано изображения римского императора Цезаря.

Рисунок 4.1 – Стартовое окно для программы «Шифр Цезаря»

По нажатию на кнопку «Ок» появляется следующее окно программы (рис. 4.2). Здесь видно, как заменяются буквы при шифровании и расшифровывании (в полях «Алфавит сообщения» и «Алфавит шифровки»).

Поле «Текст оригинального сообщения» предназначено для ввода информации, которую следует зашифровать шифром Цезаря.

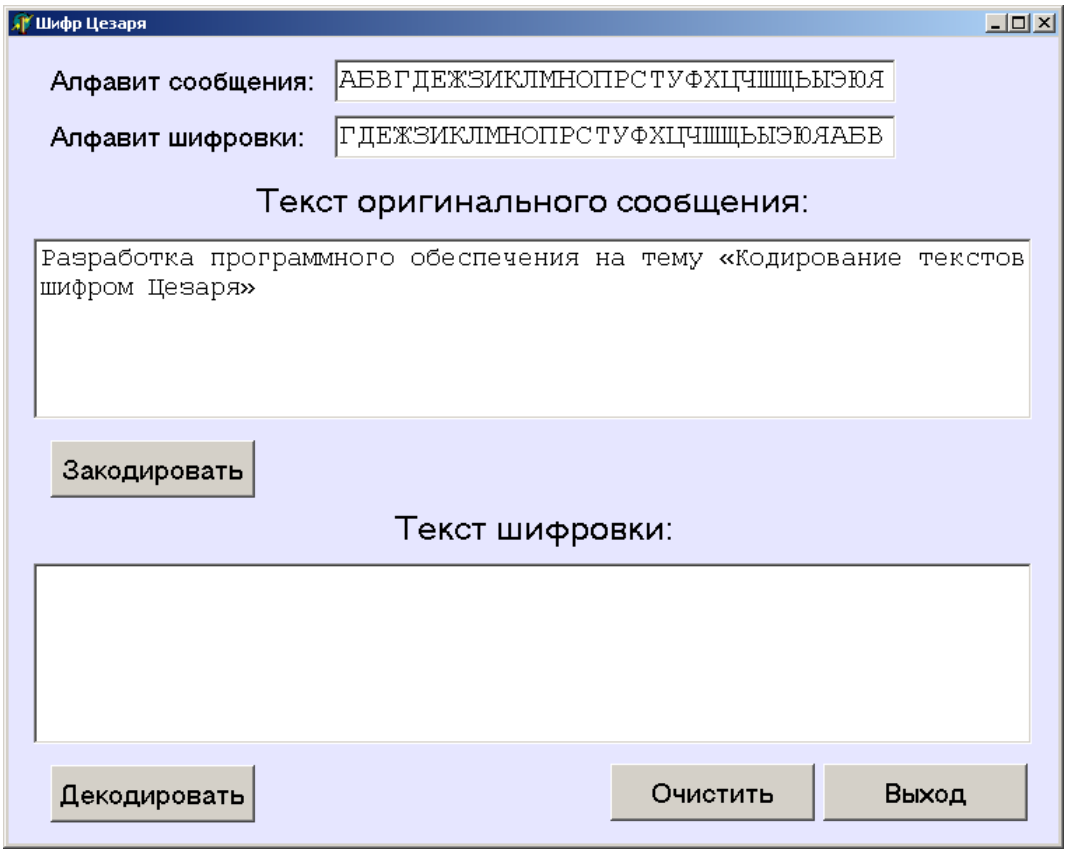

Рисунок 4.2 – Программа «Шифр Цезаря» (второе окно)

После ввода туда информации следует нажать кнопку «Закодировать». Результат отображен на рис. 4.3.

Для обратного процесса, т.е. для расшифровывания шифровки следует ввести шифровку в поле «Текст шифровки». Как это показано на рис. 4.4. Шифровка вводится блоками из 5 символов. Блоки разделены одним пробелом.

Далее следует нажать кнопку «Декодировать». Результат видно на рис. 4.5.

Кнопка «Очистить» очищает оба поля. Результат показан на рис 4.6. Кнопка «Выход» (рис. 4.6) закрывает программу.

При запуске второй программы – «Шифр Трисемиуса» – видим окно как на рис. 4.7. На стартовой форме показано снимок книги Трисемиуса «Полиграфия». Принцип работы этой программы аналогичный: нажимаем «Ок» и появляется основное окно программы (рис. 4.8); используем «Закодировать» и видим результат на рис. 4.9.

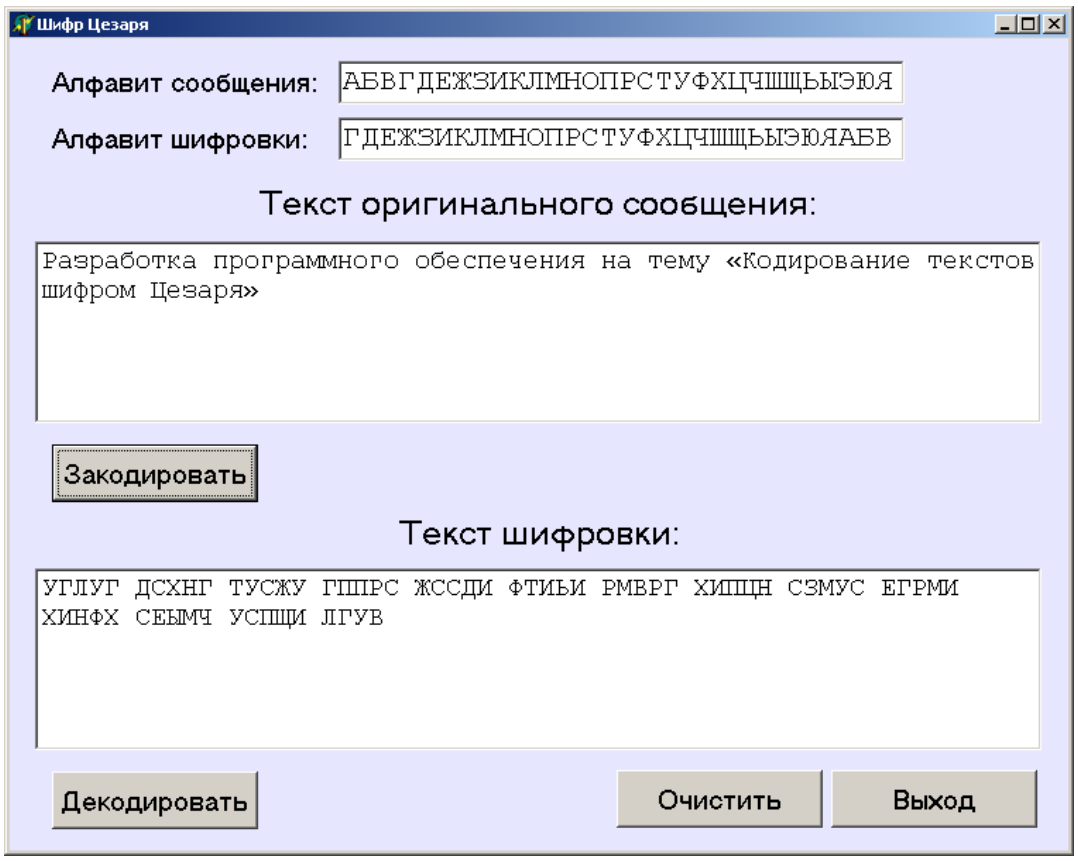

Рисунок 4.3 – Кодирование сообщения шифром Цезаря

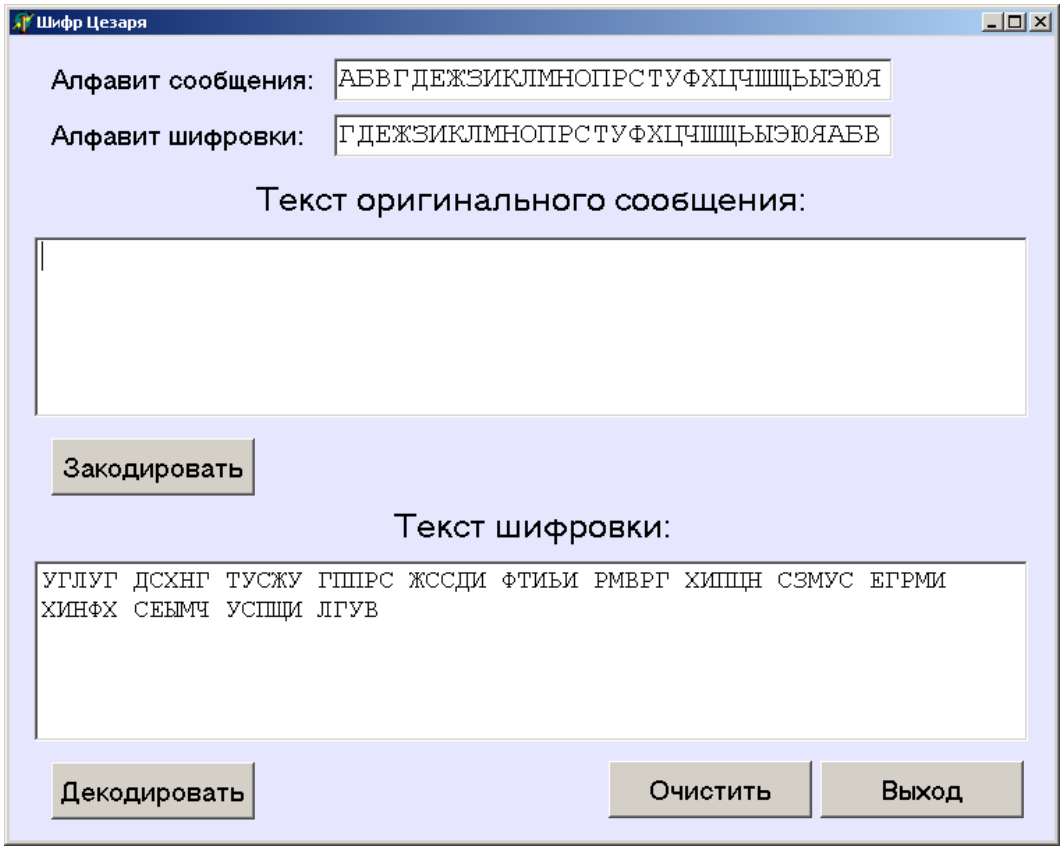

Рисунок 4.4 – Расшифровывания сообщения (шифр Цезаря)

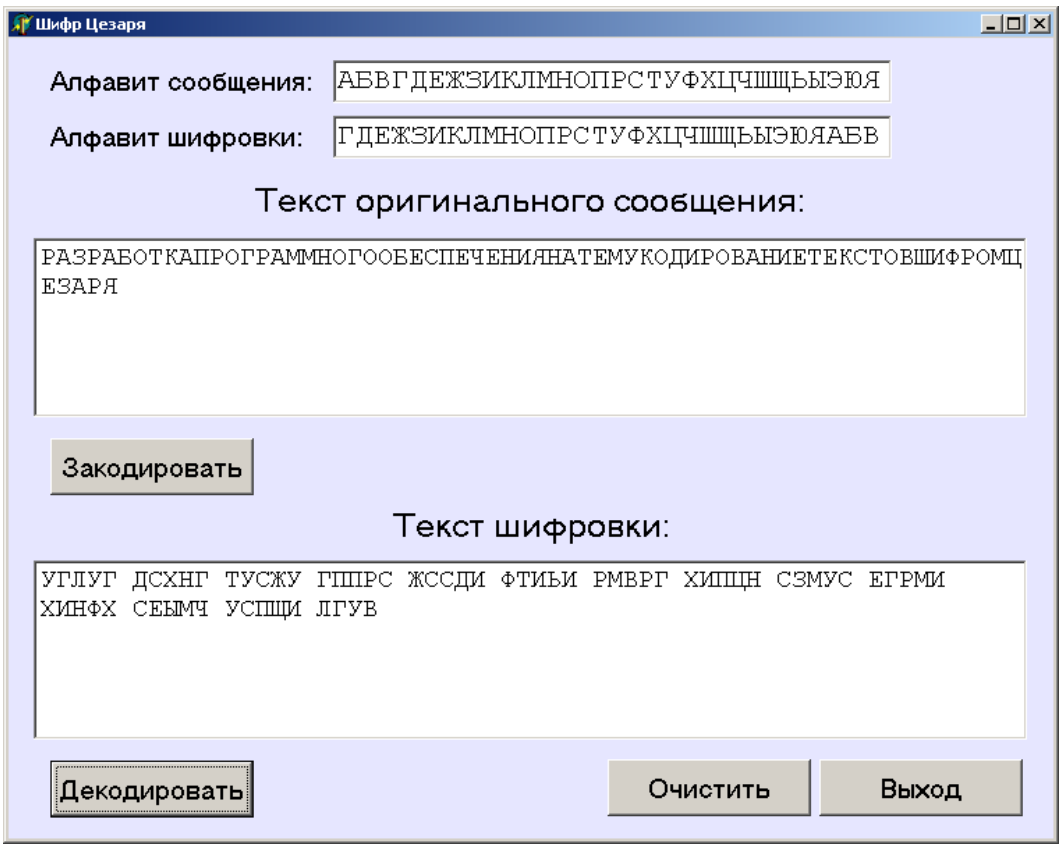

Рисунок 4.5 – Расшифровывания сообщения (шифр Цезаря)

Рисунок 4.6 – Очищения полей (шифр Цезаря)

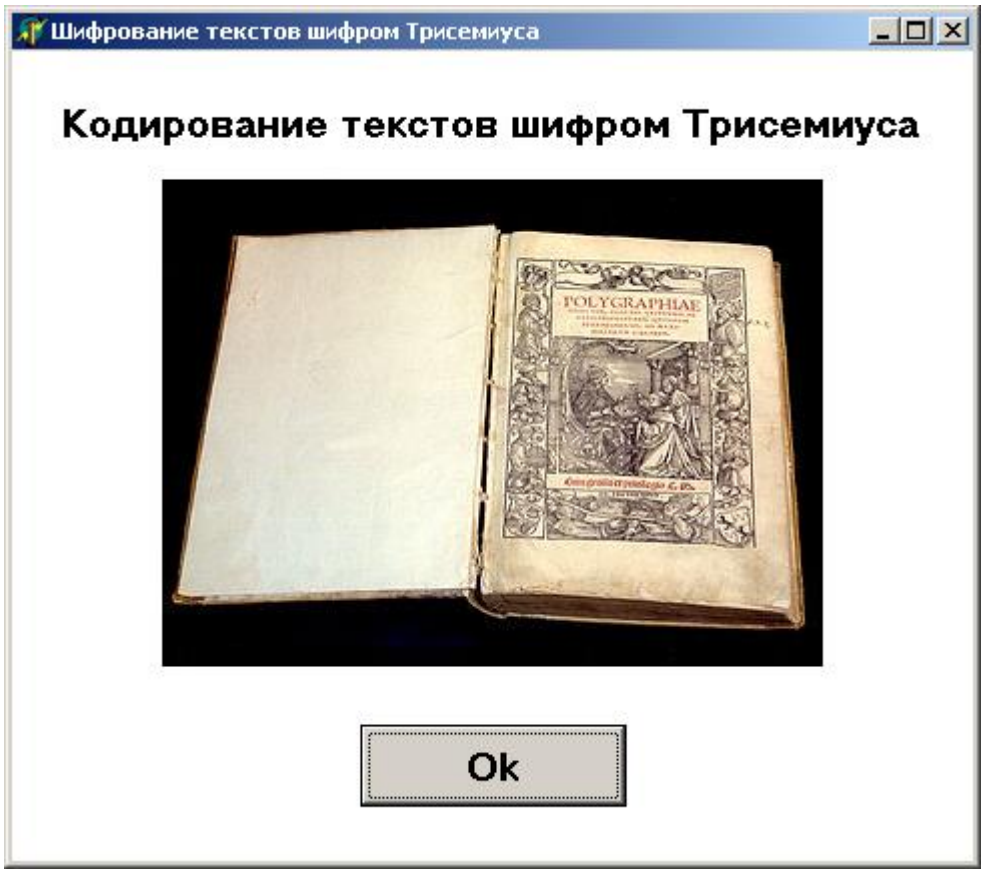

Рисунок 4.7 — Стартовое окно для программы «Шифр Трисемиуса»

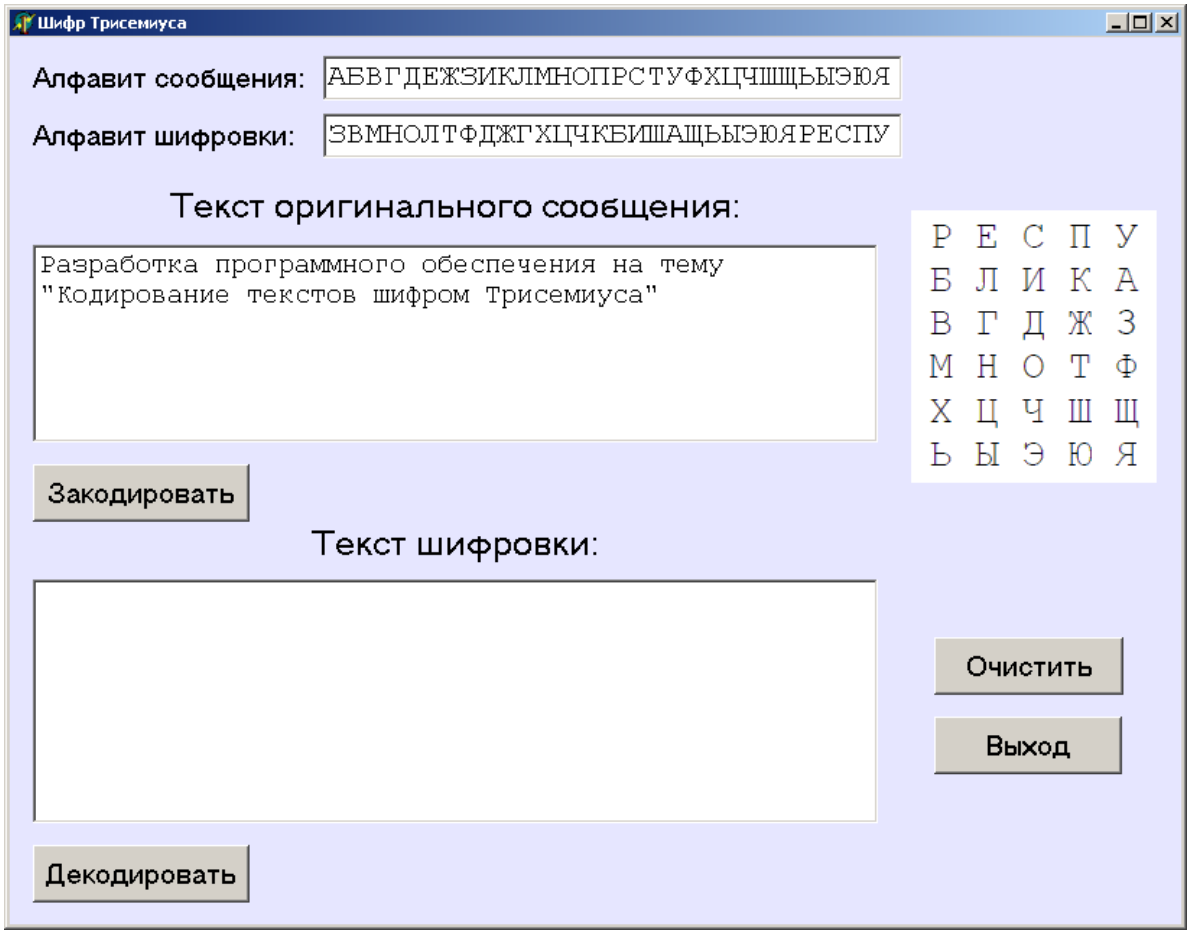

Рисунок 4.8 – Программа «Шифр Трисемиуса» (второе окно)

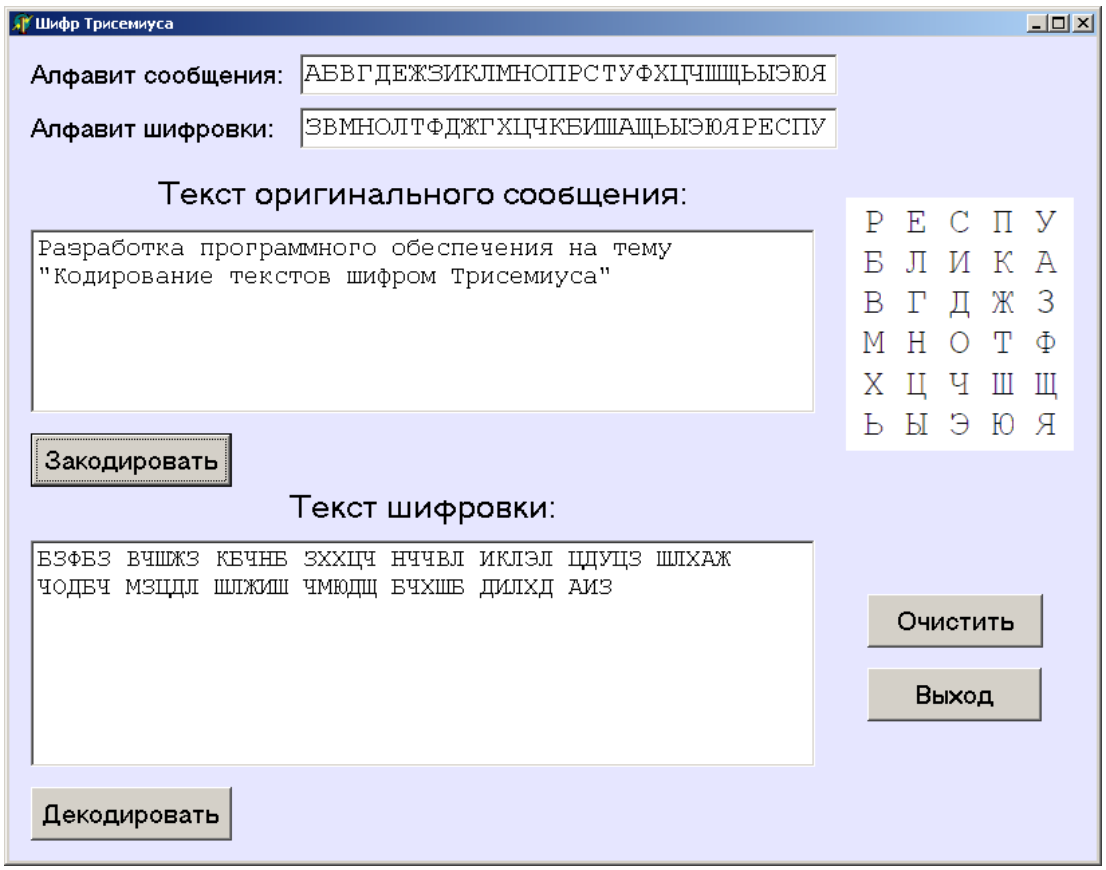

Рисунок 4.9 – Кодирование сообщения шифром Трисемиуса

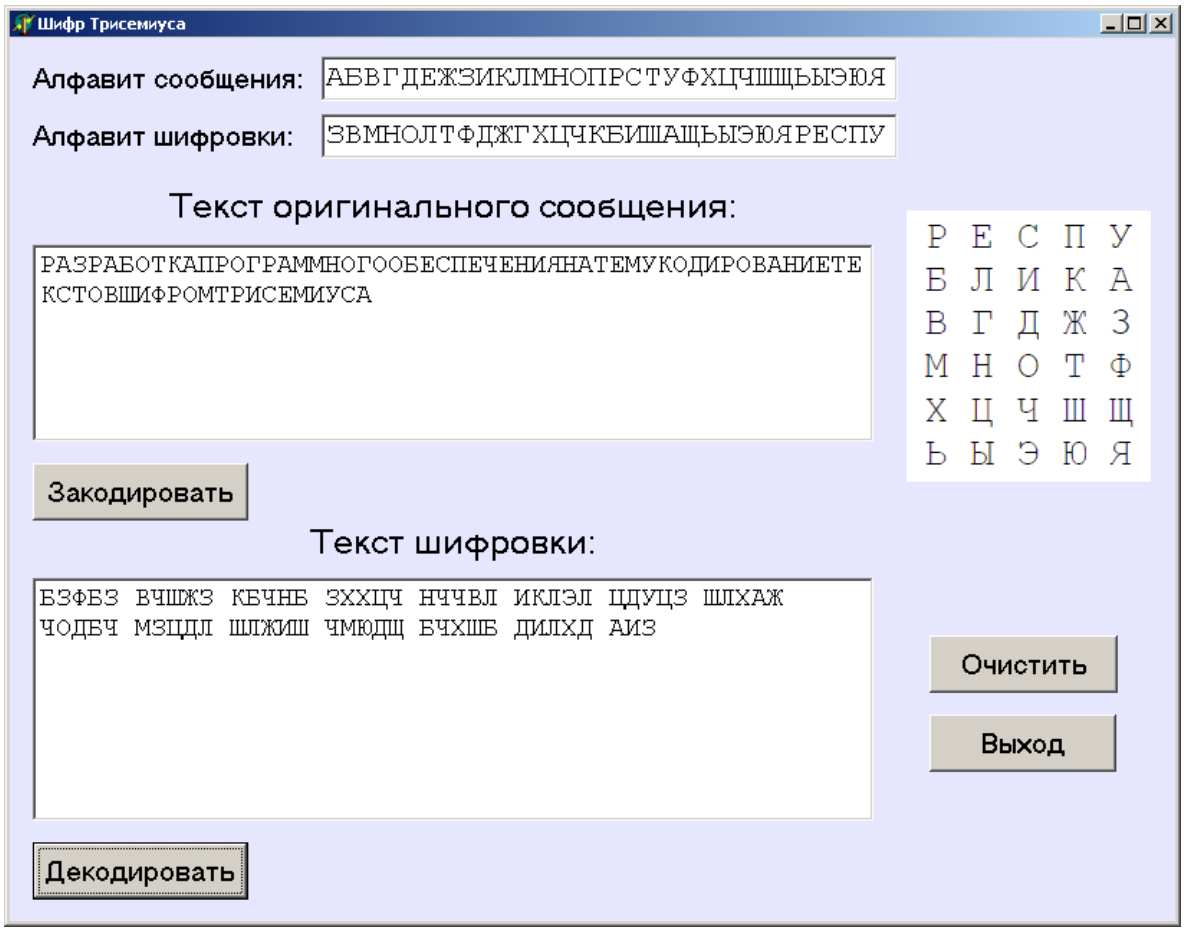

Рисунок 4.10 – Расшифровывания сообщения (шифр Трисемиуса)

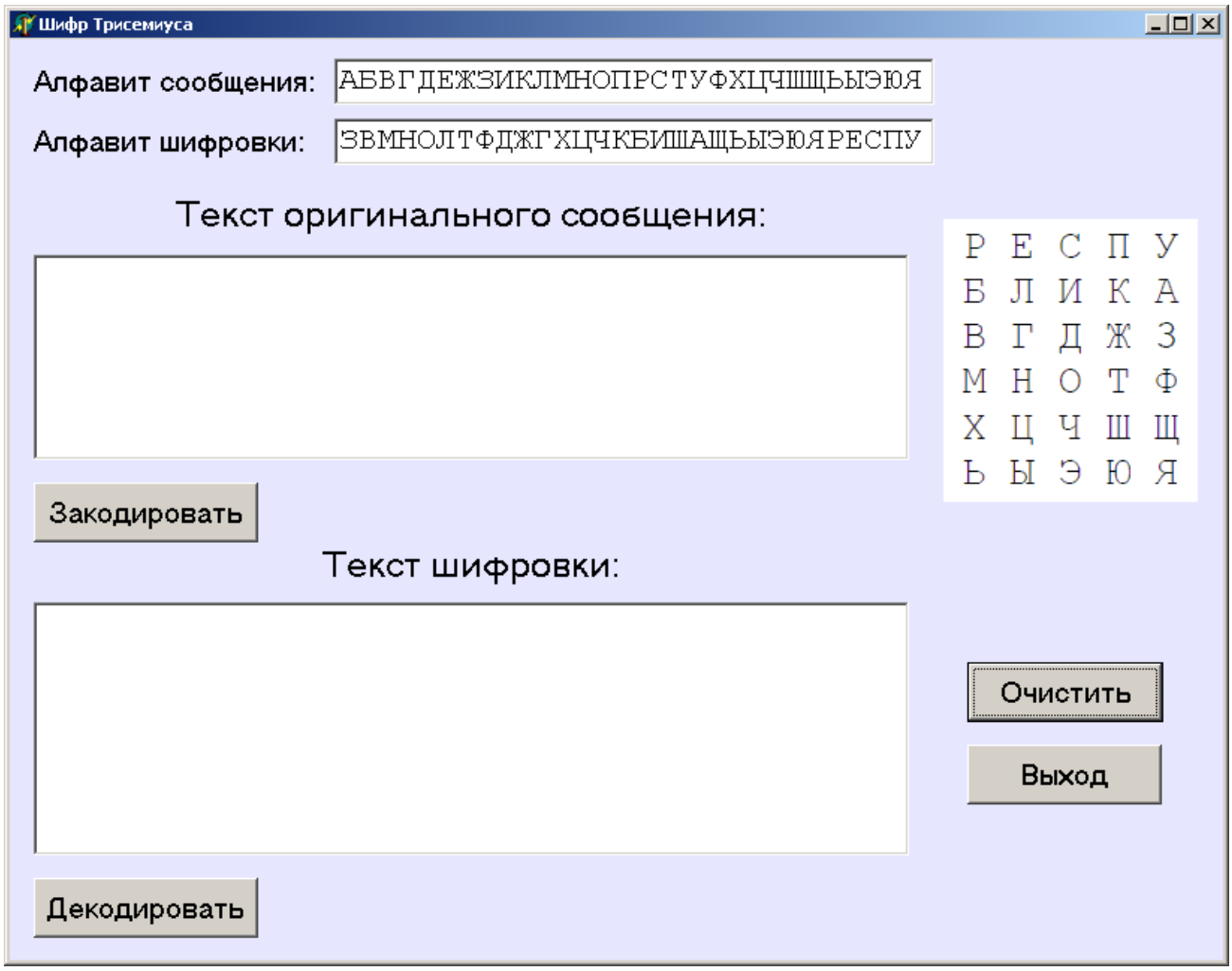

Рисунок 4.11 – Очищения полей (шифр Трисемиуса)

4.2. Тестирование программ

Проверим работу программ на примерах из раздела 3.

4.2.1. Тестирование программы «Шифр Цезаря»

В примере 3.2.1 следует закодировать сообщение «Информация». Вводим текст в первое поле. Результат виден на рис. 4.12. Ответ, полученный в примере, совпадает с ответом, выданным программой.

В примере 3.2.2 расшифровывается фраза «НСПТЮ БХИУ». Вводим текст во второе поле. Результат виден на рис. 4.13. Ответы совпали.

То есть, видим, что программа работает корректно

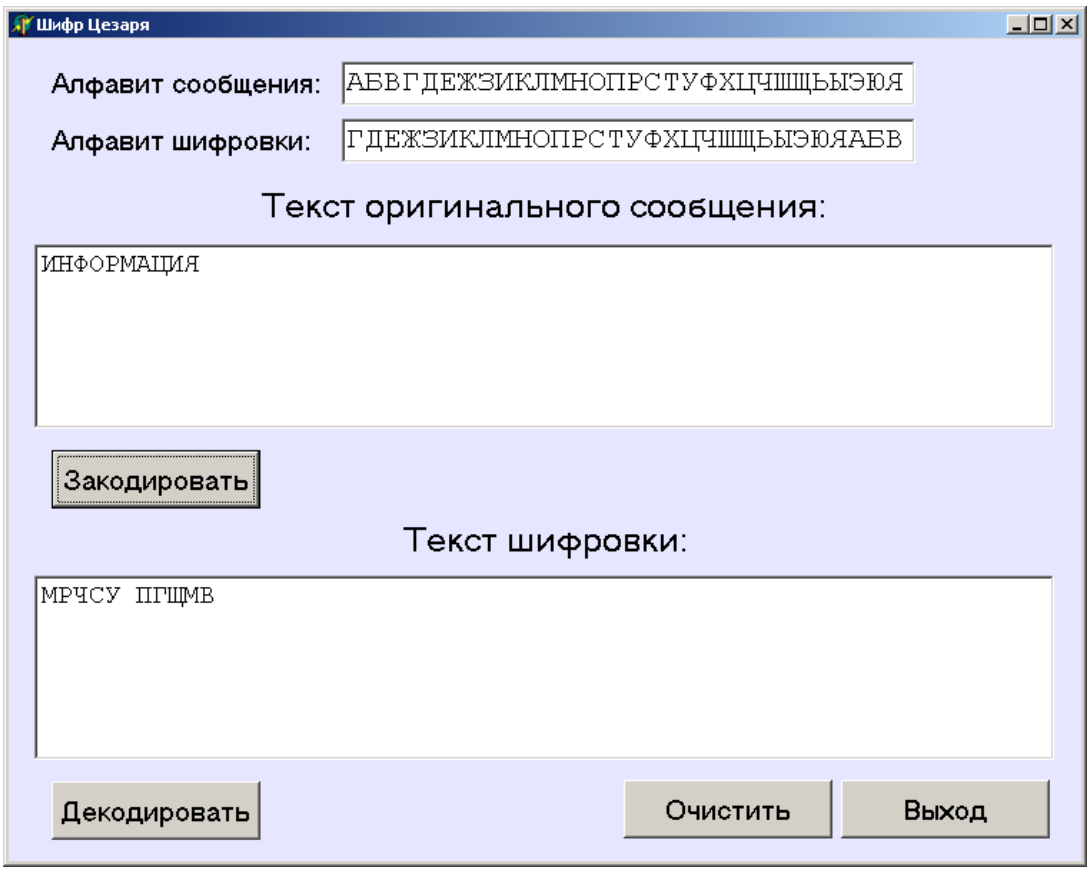

Рисунок 4.12 – Кодирования слова «Информация» (пример 3.2.1)

Рисунок 4.13 – Расшифровка сообщения «НСПТЮ БХИУ» (пример 3.2.2)

4.2.2. Тестирование программы «Шифр Трисемиуса»

В примере 3.3.1 следует закодировать сообщение «Информация». Вводим текст в первое поле. Результат виден на рис. 4.14. Ответ, полученный в примере, совпадает с ответом, выданным программой.

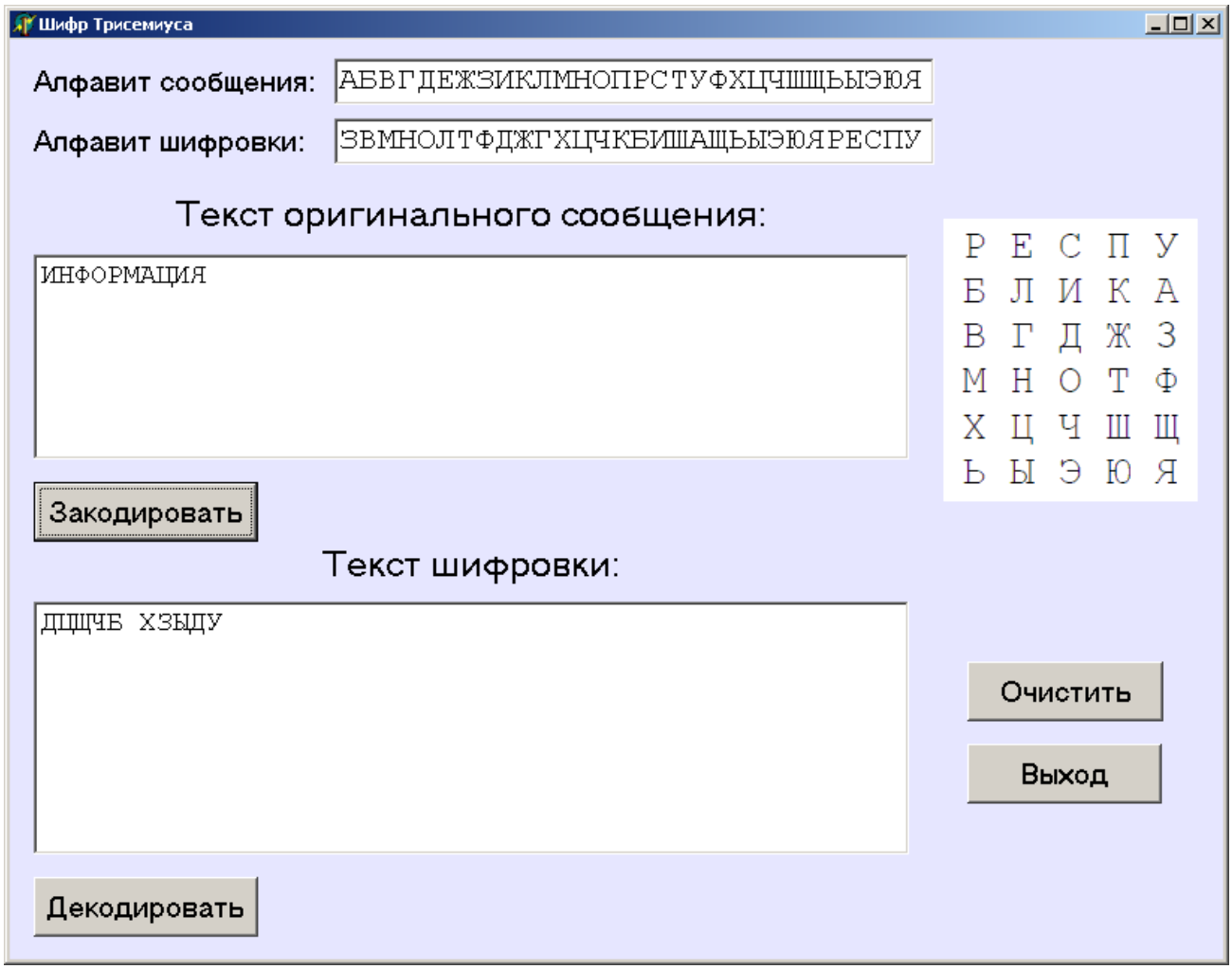

Рисунок 4.14 – Кодирования слова «Информация» (пример 3.3.1)

В примере 3.3.2 расшифровывается текст «КБЧНБ ЗХХДБ ЧМЗЦД Л». Вводим текст во второе поле. Результат виден на рис. 4.15. Ответы совпали.

Вторая программа тоже работает корректно

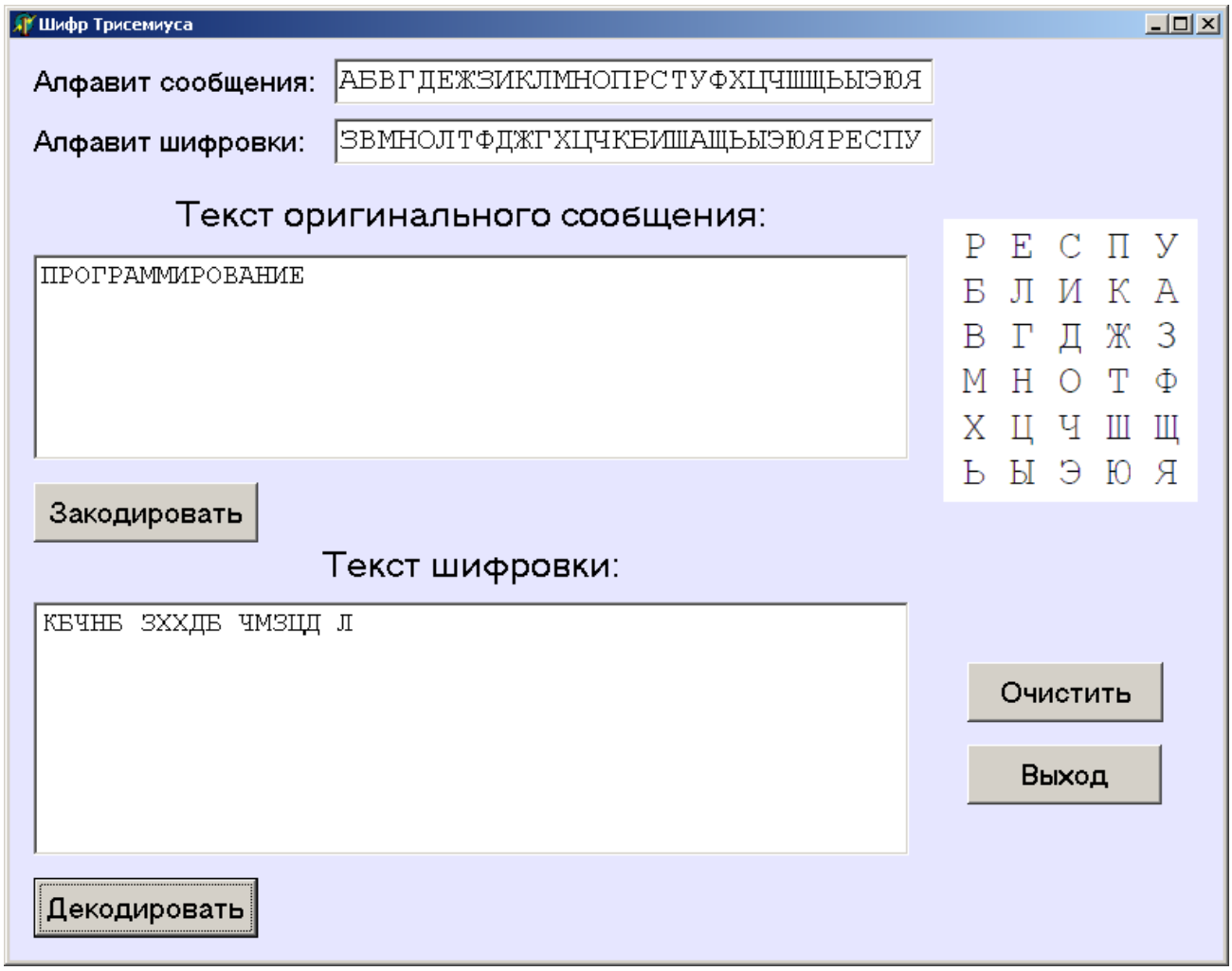

Рисунок 4.15 – Расшифровка сообщения «КБЧНБ ЗХХДБ ЧМЗЦД Л» (пример 3.3.2)

4.3. Описание создания дизайна программ

Обе программы были созданы с использование среды Delphi и языка программирования Object Pascal.

При создании програм образец интерфейса был взят из [2].

Рассмотрим эту работу на примере первой программы: «Шифр Цезаря».

Сперва была создана первая форма (рис. 4.16), которая предназначена для стартового окна (рис. 4.1).

Дизайн и структура формы показаны на рис. 4.16.

Рисунок 4.16 – Структура первого окна программы

Для отображения текста в заголовке формы было выбрано свойство `Caption`.

Для изменения цвета фона формы было взято свойство `Color=clWhite`.

Для вывода информации был использован компонент `Label1`. Для ввода текста в этот компонент взято свойство `Caption`.

Для отображения картинки был размещен компонент `Image1`. Изображение императора Цезаря было взято из интернета и с помощью свойства `Picture` загружено в компонент.

Поскольку изображение большое, то для компонента `Image1` были настроены свойства: `Stretch=true`, `Proportional=True`, `AutoSize=false`.

Первой свойство дает возможность растягивать картинку, второе сохраняет пропорции картинки при этом, третье – не отображает оригинальный размер картинки.

Для диалога с пользователем была взята кнопка `BitBtn1`. В отличие от кнопок класса `Button`, кнопки `BitBtn` имеют больше функциональных возможностей.

Надпись на кнопке была сделана с помощью свойства `Caption`.

Вторая форма программы (рис. 4.17) предназначена для шифрования текста (рис. 4.2). Ее структура показана на рисунке 4.17.

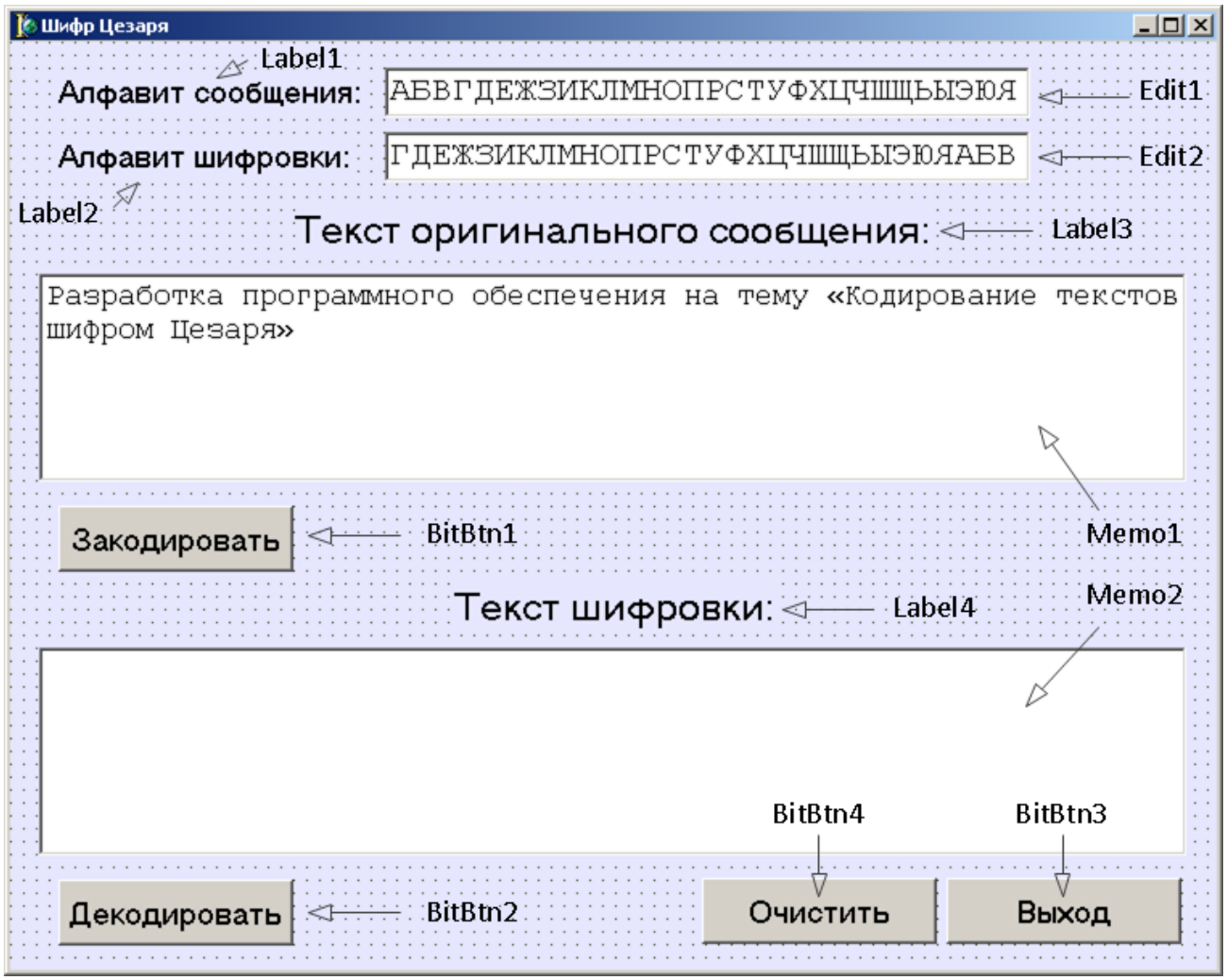

Рисунок 4.17 – Структура второго окна программы

Для подписей использованы компоненты `Label1-Label4`.

Как кнопки использованы компоненты `BitBtn1-BitBtn4`.

Для отображения алфавита сообщений и шифровки использованы компоненты `Edi1`, `Edit2`. Информация в них размещена с помощью свойства `Text`.

Для ввода текста для шифрования и отображения шифровки (и наоборот) были взяты текстовые поля `Memo1`, `Memo2`. В первое поле с помощью свойства `Lines` введен текст, показанный на рис. 4.17. Второе поле с помощью этого же свойства было очищено от тестовой информации.

Цвет фона формы было изменен с помощью свойства `Color`. В цветовой палитре был выбран светло-фиолетовый цвет (рис. 4.18). Этому цвету соответствует значение `$00FFE6E6` (рис. 4.18-4.19).

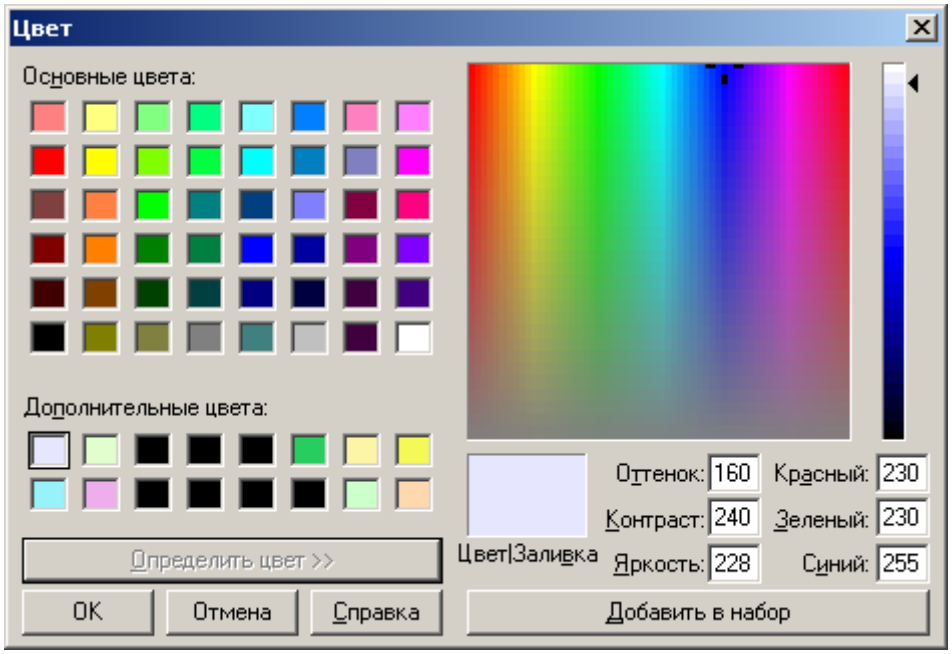

Рисунок 4.18 – Подбор цветовой гаммы для фона формы

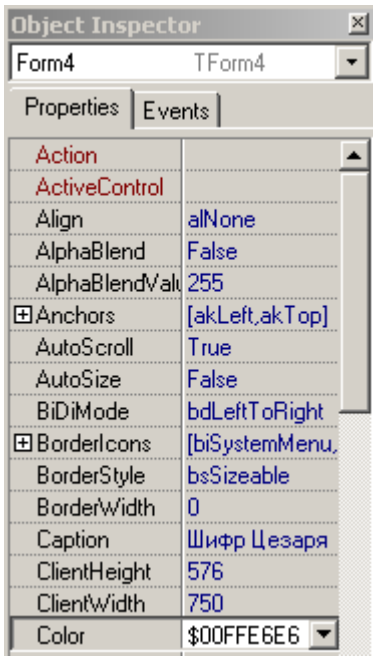

Рисунок 4.19 – Установка цвета

4.4. Описание создания кода программ

Рассмотри написание кода на примере первой программы – «Шифр Цезаря».

Для каждой кнопки был написан код программы.

Кроме того были созданы две функции: `kod` и `dekod`, которые отвечают за шифрование текста кодом Цезаря и расшифровывание. Функции принимают один символ, и возвращают аналог шифровки (функция `kod`) или дешифровки (функция `dekod`).

Фрагмент функций показан на рис. 4.20-4.21

```
// шифр Цезаря
// зашифровывание: оригинальный текст в шифровку
function kod(c:char):char;
begin
  if ((c='A') or (c='a')) then
  begin
     result:='Г';
     exit;
  end;
```

```
  if ((c='Б') or (c='б')) then
  begin
     result:='Д';
     exit;
  end;

  …

end;
```

Рис. 4.20 – Фрагмент функции kod

```
// шифр Цезаря
// расшифровывание: шифровка в расшифрованный текст
function dekod(c:char):char;
begin
  if ((c='А') or (c='а')) then
  begin
     result:='Э';
     exit;
  end;

  if ((c='Б') or (c='б')) then
  begin
     result:='Ю';
     exit;
  end;

  …

end;
```

Рис. 4.21 – Фрагмент функции dekod

При нажатии на кнопку «Закодировать» происходит следующее:

1. Создается текстовый файл «message.txt».

2. Считывается количество строк в компоненте Memo1.

3. Каждая строка запоминается в строчной переменной s2, которая затем записывается в файл. (Это связано с тем, что размер строчной

переменной не может содержать больше 255 символов, а в компоненте Memo1 может размещаться более длинный текст).

5. Создается текстовый файл «encryption.txt».

6. После создания файла «message.txt» он снова открывается и посимвольно анализируется.

7. Если символ из таблицы 3.1 (большая или маленькая буква сокращенного русского алфавита), то вызывается функция kod, которая дает шифрует символ. Зашифрованный символ записывается в файл «message.txt». Если символ не из таблицы 3.1, то он игнорируется. Через каждые пять символов шифровки ставится пробел. Символы шифровки отображаются большими буквами.

8. Файл «encryption.txt» снова открывается и его содержимое выводится в компоненте Memo2.

Код, написанный для кнопки «Декодировать», аналогичный. Только при этом используются текстовые файлы «message2.txt», «encryption2.txt» и функция dekod.

Полный листинг программ размещен в приложениях А, Б.

ВЫВОДЫ

При выполнении магистерской работы были изучены различные методы шифрование текстовой информации [2-4].

Детально был изучены шифры Цезаря и Трисемиуса и их сложность для взлома.

В пояснительной записке изложены правила шифрования и дешифрования сообщений на русском языке кодировками Цезаря и Трисемиуса.

Сформулированы алгоритмы шифрования и дешифрования обоих методов шифровок. Нарисована блок-схема одного из алгоритмов.

На языке Object Pascal в среде программирования Dephi написаны две программы, которые кодируют и раскодируют текстовую информацию на русском языке с помощью шифров Цезаря и Трисемиуса.

Программы протестированы и показали корректную работу.

Была создана инструкция по работе с программами. Процесс создания программ задокументирован. Код обоих программ (полностью) изложен в приложениях к пояснительной записке.

В дальнейшем целесообразно детально ознакомиться с другими методами шифрования текстовой информации и реализовать эти методы программно.

ЛИТЕРАТУРА

1. Емец О. А. Методические рекомендации по оформлению пояснительных записок курсовых проектов (работ) [Электронный ресурс]: для студентов направления подготовки 6.040302 «Информатика» специальности 7.04030203, 8.04030203 «Социальная информатика» ПУЭТ / О. А. Емец, А. О. Емец. – Полтава: ПУЭТ, 2015. – [Электронный ресурс].

2. Басов В. Е. Методичний посібник до лабораторних робіт по курсу «Захист інформації» / В. Е. Басов. – Одеса: Одеська національна академія зв'язку ім. О. С. Попова, 2003. – 26 с.

3. Колінько Є.А. Пояснювальна записка до бакалаврської роботи на тему «Розробка програмного забезпечення з теми «Кодування текстів шифром play fair». – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/9004

4. Ніколаєнко О.В. Пояснювальна записка до бакалаврської роботи на тему «Розробка програмного забезпечення з теми «Кодування текстів шифром Гонсфельда». – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/9019

5. Культин Н. Основы программирования в Delphi 2010. / Н. Культин. – СПб.: БХВ-Петербург, 2010. – 438 с.

6. Осипов Д. Л. Delphi. Программирование для Windows, OS X, iOS и Android / Л. Д. Осипов. – СПб.: БХВ-Петербург, 2014. – 464 с.

7. Парфьонова Т.О. Дистанційний курс ПУЕТ «Захист інформації» для студентів спеціальності «Комп'ютерні науки» / Т.О. Парфьонова. – [Електронний ресурс].

8. Парфьонова Т.О. Дистанційний курс ПУЕТ «Теорія інформації та кодування» для студентів спеціальності «Комп'ютерні науки» / Т.О. Парфьонова. – [Електронний ресурс].