Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных технологий

Кафедра проектирования информационных компьютерных систем

Дисциплина "Современные технологии проектирования информационных систем"

К защите допустить:
Руководитель курсовой работы
старший преподаватель
кафедры
_____ А.В.Михалькевич
22.12.2024

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

Сервер дешифрования текстов зашифрованных методом замены

БГУИР КР 1-40 05 01-10 № 182 ПЗ

Студент (подпись студента) К.А.Пускова

Курсовая работа
представлена на проверку
22.12.2024

_____
(подпись студента)

2024

Реферат

БГУИР КР 1-40 05 01-10 № 182 ПЗ, гр. 784371

К.А.Пускова, Сервер дешифрования текстов зашифрованных методом замены, Минск: БГУИР - 2024.

Пояснительная записка 54195 с., 3 рис., 4 табл.

Ключевые слова: ООП, Delphi, MVC, Java

Предмет Современные технологии проектирования информационных систем, А.В.Михалькевич

Целью курсового проекта является разработка программного модуля для исследования взаимного положения отрезков.

The aim of the course project is to develop a software module to study the relative position of the segments.

Содержание

Введение
1 ПОСТАНОВКА ЗАДАЧИ
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ
3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ
4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ
Заключение
Список использованных источников
Приложения

Введение

ВВЕДЕНИЕ В настоящее время компьютеры применяются повсеместно. Компью-терные технологии позволяют расширить возможности и быстродействие лю-бой сферы деятельности человека. Современный уровень развития компьютерных технологий позволяет создавать программы, которые обладают неограниченными возможностями, при этом обеспечивают большую достоверность и позволяют перейти на ка-чественно новый уровень проектирования. Эффективное использование ком-пьютеров для решения инженерных и научных задач невозможно без знаний основных методов составления схем алгоритмов, написания эффективного программного обеспечения на языке программирования. Целью курсового проекта является разработка программного модуля для исследования взаимного положения отрезков. Для достижения указанной цели необходимо решение следующих за-дач: - Провести анализ литературных источников для решения задачи, сформулировать требования к программному продукту; - Разработать математическую модель решения задачи; - Разработать необходимые алгоритмы, формы представления вход-ных и выходных данных, написать код программы; - Провести тестирование программного продукта, выявить все до-пущенные ошибки и устранить их. Для реализации курсового проекта будет использована среда про-граммирования Delphi.

1 ПОСТАНОВКА ЗАДАЧИ

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Общая характеристика задачи

Назначение и актуальность разработки. Программный продукт предназначен для шифрования и дешифрования текста. Программа позволит зашифровать текст двумя способами. Также система будет выполнять дешифрацию уже зашифрованного текста.

В отличие от шифрования и дешифрования вручную, автоматизированная система дает ряд преимуществ:

– сокращение ручного труда (сведение к минимуму);

– экономия временного ресурса;

Требования к программе. Программа должна осуществлять распознание и шифрование вводимого текста, также программа должна распознавать уже зашифрованный текст и дешифрировать его.

Входные данные будут вводиться пользователем с клавиатуры, результаты работы – выводится на экран.

1.2 Описание методов шифрования используемых в программе

В данной программе приведены два метода шифрования: “шифр Цезаря” и “шифр Атбаш”.

Шифр Цезаря, также известный как шифр сдвига, код Цезаря или сдвиг Цезаря — один из самых простых и наиболее широко известных методов шифрования. Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом вправо на 3, А была бы заменена на Г, Б станет Д, и так далее. Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами. Шаг шифрования, выполняемый шифром Цезаря, часто включается как часть более сложных схем, таких как шифр Виженера, и все ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет практически никакого применения на практике.

Шифр Атбаш - это шифр простой замены, изначально придуманный для еврейского алфавита древними евреями. Суть шифра: если символ шифруемого текста i-ый в алфавите, если считать с начала алфавита, он заменяется символом, i-ым с конца алфавита. Сейчас шифром Атбаш называется шифр, который работает по такому правилу с каким-либо алфавитом, не важно, каким именно. Допустим, у нас в алфавите А, В, С, запятая, точка, плюс. Тогда получаем вот такую таблицу замены:

А	В	С	,	.	+
+	.	,	С	В	А

Зашифруем текст "А+В". Символ А первый с начала алфавите, символ ПЛЮС первый с конца алфавита. Вместо А пишем плюсик. Нетрудно догадаться, что вместо плюсика, который идёт после А в открытом тексте, ставим А. Символ В второй с начала алфавита, по правилам шифра Атбаш берём символ, второй с конца, а это точка. Итог: "+А."

Для русского алфавита шифр Атбаш работает по такой таблице:

А Б В … Ю Я

Я Ю Э … Б А

Заметим, что если шифр Атбаш используем для алфавита с нечётным числом символов, символ, которые посередине алфавита, не заменяется. Ну это так, просто примечательное свойство… Шифр Атбаш является шифром без ключа, то есть ведёт замену символов всегда одинаково при заданном алфавите. Это значит, что требуется исключить знание противником, что используется именно этот шифр, иначе даже вручную зашифрованный текст быстро прочитывается.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

2.1 Описание данных, используемых в программе

Исходными (входными) данными, которые будут вводиться с клавиатуры, является какой-нибудь текст, который нужно зашифровать (дешифровать). Результатами работы программы будет являться зашифрованный текст (дешифрованный). Описание переменных, используемых в программе, приведено в таблице 1.

Таблица 1 – Описание переменных

Имя переменной	Описание	Тип данных
i, j	Счётчик	Целочисленный
s, s2, v	Исходные и зашифрованные строки	Строковый
s1	Отдельный символ из шифруемого слова	Строковый
k	Количество строк	Целочисленный
a[i], s[i]	Отдельная строчка, которую нужно зашифровать	Массив
z	Номер символа в таблице ASCII	Целочисленный
n	Количество символов в строке	Целочисленный

2.2 Описание схемы программы

Схема алгоритма работы программы представлена на рисунке 1.

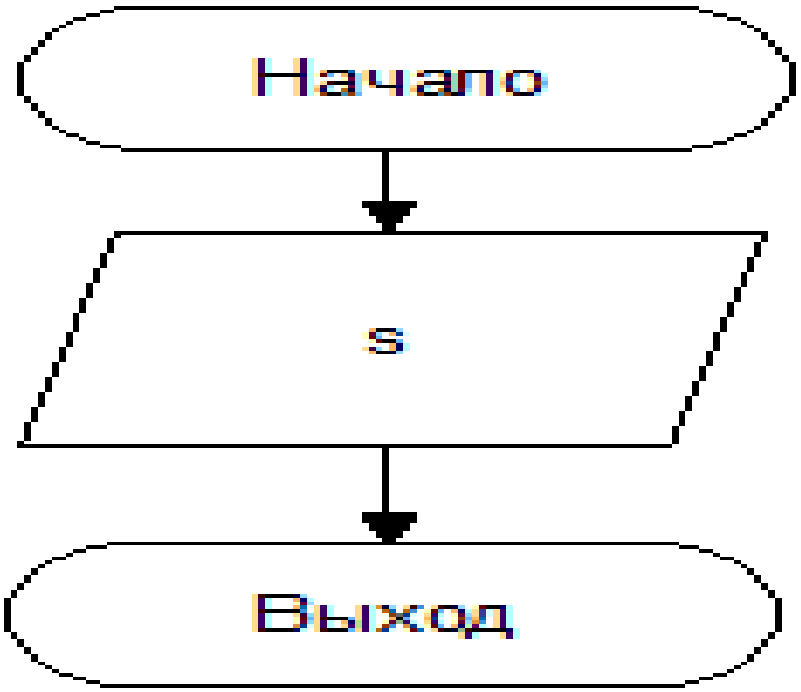

Рисунок 1 Лист 1 – Схема алгоритма работы программы

Схема алгоритма ввода исходных данных представлена на рисунке 2.

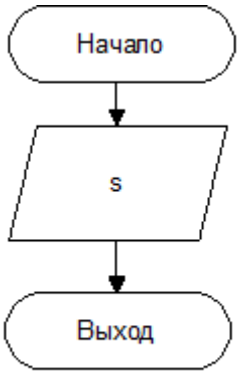

Рисунок 2 Лист2 – Схема алгоритма ввода данных

Схема алгоритма шифрования данных методом “Цезая” и вывода результатов, представлена на рисунке 3.

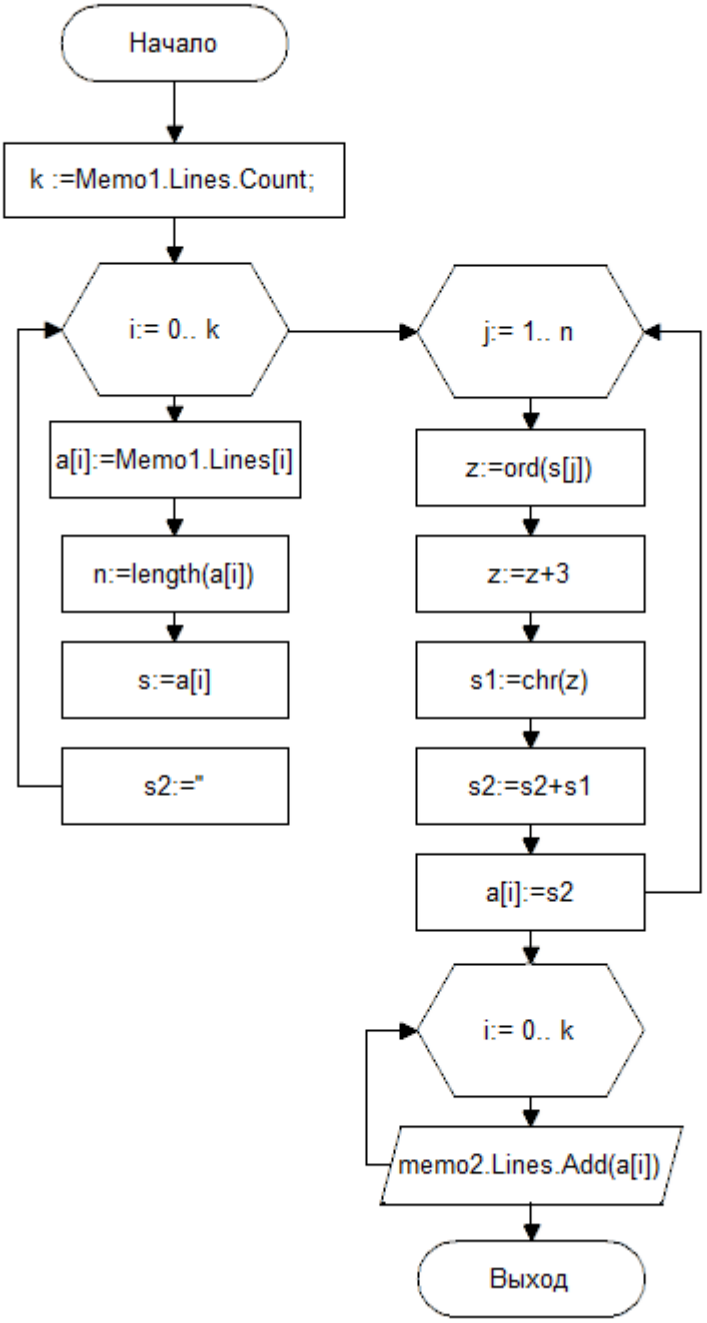

Рисунок 3 Лист 2 – Схема алгоритма шифрования данных методом “Цезая”

Схема алгоритма шифрования данных методом “Атбаш” и вывода результатов, представлена на рисунке 4.

Рисунок 4 Лист 3 – Схема алгоритма шифрации данных методом “Атбаш”

Схема алгоритма дешифрования данных методом “Цезая” и вывода результатов, представлена на рисунке 5.

Рисунок 5 Лист 4 – Схема алгоритма дешифрования данных методом “Цезая”

Схема алгоритма дешифрования данных методом “Атбаш” и вывода результатов, представлена на рисунке 6.

Рисунок 6 Лист 5 – Схема алгоритма дешифрования данных методом “Атбаш”

3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ

3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ

3.1 Описание структуры разрабатываемого проекта

Разрабатываемый программный продукт будет содержать 2 формы: главную форму приложения и основную форму (в которой происходит ввод данных, само шифрование/дешифрование данных и вывод конечных результатов). Основное окно приложения будет содержать главное меню. Схема главного меню представлена рисунке 7.

Рисунок 7 Лист 6 – Схема главного меню

3.2 Создание форм программы

Главная форма программы раскрывается при запуске программы и содержит следующие элементы:

Заголовок приложения с изображением собственной пиктограммы;
Строку пояснения, где отображаются для чего разработана данная программа;
Кнопку “Далее”, при нажатии которой выполняется переход на основную форму;

Форма главного окна представлена на рисунке 7.

Рисунок 7 – Форма главного окна

Основная форма программы раскрывается при нажатии кнопки и содержит следующие элементы:

Заголовок приложения с изображением собственной пиктограммы;
Строку главного меню в верхней части формы;
Поля для исходного/зашифрованного текста;
Кнопку “Зашифровать”, при нажатии которой выполняется шифрование текста введенного в поле для исходного текста и вывод зашифрованного текста в поле для зашифрованного текста (шифрование выполняется методом “Цезаря”);
Кнопку “Зашифровать”, при нажатии которой выполняется шифрование текста введенного в поле для исходного текста и вывод зашифрованного текста в поле для зашифрованного текста (шифрование выполняется методом “Атбаш”);

Кнопку “Расшифровать”, при нажатии которой выполняется дешифрование текста введенного в поле для зашифрованного текста и вывод расшифрованного текста в поле для исходного текста (дешифрование выполняется методом “Цезаря”);
Кнопку “Расшифровать”, при нажатии которой выполняется дешифрование текста введенного в поле для зашифрованного текста и вывод расшифрованного текста в поле для исходного текста (дешифрование выполняется методом “Атбаш”);
Кнопку “Cancel”, при нажатии которой выполняется выход из программы;

Форма основного окна представлена на рисунке 8.

Рисунок 8 – Форма основного окна

При разработке основной формы программы были использованы следующие компоненты: BitBtn, Label, Image, MainMenu, Memo.

3.3 Написание кода для ввода исходных данных, их шифрации и вывода зашифрованных данных (метод “Цезаря”)

Для ввода исходных данных используется компонент Memo1, находящийся на основной форме. Для шифрации введенного текста программа считает количество строк во всем тексте, и в каждой строке находит номер каждого символа, за тем прибавляет к номеру символа число три и выводит в место исходного символа символ который соответствует номеру исходного символа увеличенному на число три в компонент Memo2.

Пример шифрации исходных данных и вывода зашифрованных данных (метод “Цезаря”):

```
//Шифрование методом “Цезаря”
k := Memo1.Lines.Count;
for i := 0 to k do  begin
a[i]:=Memo1.Lines[i];
n:=length(a[i]);
s:=a[i];
s2:='';
for j := 1 to n do  begin
z:=ord(s[j]);
z:=z+3;
S1:=chr(z);
s2:=s2+s1;
a[i]:=s2;
end;
end;
//Вывод зашифрованных данных
 for i := 0 to k do
Memo2.Lines.Add(a[i]);
end;
```

3.4 Написание кода для ввода исходных данных, их шифрации и вывода зашифрованных данных (метод “Атбаш”)

Для ввода исходных данных используется компонент Memo1, находящийся на основной форме. Для шифрации введенного текста программа считает количество строк во всем тексте и если трока не пуста, то программа заменяет каждый символ в этой строке на соответствующий символ, после замены всех символов строки программа переходит на следующую строку и так пока не будут пройдены все строки, зашифрованные данные выводятся в компонент Memo2.

Пример шифрации исходных данных и вывода зашифрованных данных (метод “Атбаш”):

```
//шифрование методом “Атбаш”
s:=memo1.Lines[j];
        n:=length(s);
        For i:=0 to n do begin
        if s<>'' then    case s[i] of
              'A': insert('Z',v,i);
              'B': insert('Y',v,i);
              …//символы от D до X
              'Y': insert('B',v,i);
              'Z': insert('A',v,i);
               //xxx
              'a': insert('z',v,i);
              'b': insert('y',v,i);
               …//символы от d до x
              'y': insert('b',v,i);
              'z': insert('a',v,i);
               //xxx
              'А': insert('Я',v,i);
              'Б': insert('Ю',v,i);
               …//символы от Д до Э
              'Ю': insert('Б',v,i);
              'Я': insert('А',v,i);
              //XXX
             'а': insert('я',v,i);
             'б': insert('ю',v,i);
              …//символы д-э
             'ю': insert('б',v,i);
             'я': insert('а',v,i);
          //xxx
             '0': insert('9',v,i);
             '1': insert('8',v,i);
             …//символы от 2 до №
             '<': insert('<',v,i);
```

```
                '>': insert('>',v,i);
            end;
            end;
    //вывод зашифрованных данных
    memo2.Lines.Add(v);
    end;
```

3.5 Написание кода для ввода зашифрованных данных, их дешифрации и вывода исходных данных (метод “Цезаря”)

Для ввода зашифрованных данных используется компонент Memo2, находящийся на основной форме. Для дешифрации введенного текста программа считает количество строк во всем тексте, и в каждой строке находит номер каждого символа, за тем отнимает от номера символа число три и выводит в место исходного символа символ который соответствует номеру исходного символа уменьшенному на число три в компонент Memo1.

Пример дешифрации зашифрованных данных и вывода исходных данных (метод “Цезаря”):

```
//дешифрование методом “Цезаря”
k := Memo2.Lines.Count;
for i := 0 to k do  begin
a[i]:=Memo2.Lines[i];
n:=length(a[i]);
s:=a[i];
s2:='';
for j := 1 to n do  begin
z:=ord(s[j]);
z:=z-3;
S1:=chr(z);
s2:=s2+s1;
a[i]:=s2;
end;
end;
//Вывод исходных данных
for i := 0 to k do
Memo1.Lines.Add(a[i]);
end;
```

3.6 Написание кода для ввода зашифрованных данных, их дешифрации и вывода исходных данных (метод “Атбаш”)

Для ввода зашифрованных данных используется компонент Memo2, находящийся на основной форме. Для дешифрации введенного текста программа считает количество строк во всем тексте и если трока не пуста, то программа заменяет каждый символ в этой строке на соответствующий символ, после замены всех символов строки программа переходит на

следующую строку и так пока не будут пройдены все строки, исходные данные выводятся в компонент Memo1.

Пример дешифрации исходных данных и вывода зашифрованных данных (метод “Атбаш”):

```
//дешифрование методом “Атбаш”
s:=memo2.Lines[j];
        n:=length(s);
        For i:=0 to n do begin
        if s<>'' then    case s[i] of
            'A': insert('Z',v,i);
            'B': insert('Y',v,i);
             …//символы от D до X
            'Y': insert('B',v,i);
            'Z': insert('A',v,i);
             //xxx
            'a': insert('z',v,i);
            'b': insert('y',v,i);
            …//символы от d до x
            'y': insert('b',v,i);
            'z': insert('a',v,i);
             //xxx
            'А': insert('Я',v,i);
            'Б': insert('Ю',v,i);
            …//символы от Д до Э
            'Ю': insert('Б',v,i);
            'Я': insert('А',v,i);
            //XXX
           'а': insert('я',v,i);
           'б': insert('ю',v,i);
           …//символы д-э
           'ю': insert('б',v,i);
           'я': insert('а',v,i);
         //xxx
           '0': insert('9',v,i);
           '1': insert('8',v,i);
           …//символы от 2 до №
           '<': insert('<',v,i);
           '>': insert('>',v,i);
        end;
        end;
//вывод зашифрованных данных
Memo1.Lines.Add(v);
```

end;

3.7 Проектирование интерфейса программы

Интерфейс приложения содержит следующие формы:

Главную форму программы;
Основную форму программы;

Главное окно содержит строку пояснения, где отображаются для чего разработана данная программа и кнопку “Далее” при нажатии которой выполняется переход на основную форму программы.

Форма главного окна представлена на рисунке 8.

Рисунок 8 – Форма главного окна

Основное окно содержит главное меню содержащее три пункта (Зашифровать, Расшифровать, Справка), поле для исходного текста, поле для зашифрованного текста, две кнопки “Зашифровать”, две кнопки “Расшифровать” (каждая кнопка шифрует дынные разными способами, если была выбрана одна кнопка “Зашифровать”, то та кнопка “Расшифровать” которая предусмотрена не для этого метода станет неактивной), кнопку “Retry” (кнопка рестарта программы) и кнопку “Cancel” (кнопка выхода из программы).

Форма основного окна представлена на рисунке 9.

Рисунок 9 – Форма основного окна

При нажатии в главном меню программы пункта “Зашифровать” появляется вкладка, приведенная на рисунке 10.

Что происходит при нажатии пункта “Зашифровать” 10.

Рисунок 10 – Вкладка 1

При нажатии в главном меню программы пункта “Расшифровать” появляется вкладка, приведенная на рисунке 11.

Что происходит при нажатии пункта “Зашифровать” 10.

Рисунок 10 – Вкладка 2

Программа содержит справочную систему, приведенную на рисунке 12.

Рисунок 12 - Справочник

4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

4.1 Тестирование алгоритмов шифрации

Для проверки правильности выполнения шифрации в программе, были подобраны исходные данные, позволяющие протестировать работу программы в различных случаях. Эти исходные данные были зашифрованы программой и расшифрованы обратно. В итоге программой выведены такие же исходные данные которые были введены изначально.

	Исходные данные	Результат шифрации	Результаты дешифрования
1	Ввод: Привет!	Тулеих$	Привет!
2	Ввод: Привет!	Поцэъм!	Привет!
3	Ввод: gkjsdfkgkdkfl gkdkgkl;jkgg 34567	jnmvginjngnio#jngnjno>mnjj 6789:	gkjsdfkgkdkfl gkdkgkl;jkgg 34567
4	Ввод: gkjsdfkgkdkfl gkdkgkl;jkgg 34567	tpqhwuptpwpuotpwptpo;qptt 65432	gkjsdfkgkdkflgkdkgkl;jkgg 34567

Таблица 2 – результаты тестирования

Рисунок 14 – Результаты тестирования 1 (Шифр “Цезаря”)

=Рисунок 15 – Результаты тестирования 2 (Шифр “Атбаш”)

Рисунок 16 – Результаты тестирования 3 (Шифр “Цезаря”)

Рисунок 17 – Результаты тестирования 4 (Шифр “Атбаш”)

Проанализировав полученные результаты, можно сделать вывод о том, что программный продукт формирует верные результаты.

4.2 Тестирование программы

Тестирование программного продукта проводилось по всем функциям программы. Результаты тестирования представлены в таблице 3.

Таблица 3 – Журнал тестирования

Действие актера	Действие программы	Отметка о правильной работе или описание ошибки
1	2	3
Зашифровать исходные данные (методом “Цезаря”).	Ввести в поля для исходного текста дынные, выбрать пункт меню “Зашифровать”, “Шифр Цезаря” или первую кнопку “Зашифровать”	Исходные данные введены и выведены зашифрованные данные. Действие выполнено успешно.
1	2	3
Зашифровать исходные данные (методом “Атбаш”).	Ввести в поля для исходного текста дынные, выбрать пункт меню “Зашифровать”, “Шифр Атбаш” или вторую кнопку “Зашифровать”	Исходные данные введены и выведены зашифрованные данные. Действие выполнено успешно.
Дешифровать зашифрованные данные (методом “Цезаря”).	Ввести в поля для зашифрованного текста дынные, выбрать пункт меню “Расшифровать”, “Шифр Цезаря” или первую кнопку “Расшифровать”	Зашифрованные данные введены и выведены расшифрованные данные. Действие выполнено успешно.
Дешифровать зашифрованные данные (методом “Атбаш”).	Ввести в поля для зашифрованного текста дынные, выбрать пункт меню “Расшифровать”, “Шифр Атбаш” или вторую кнопку “Расшифровать”	Зашифрованные данные введены и выведены расшифрованные данные. Действие выполнено успешно.
Вызов справки	Выбрать пункт меню “Справка”	Открылось окно справки. Действие выполнено успешно.
Рестарт программы	Нажать кнопку “Retry”	Программа очистит все поля и все кнопки стали активными. Действие выполнено успешно.
Выход из программы	Нажать кнопку “Cancel”	Программа закрылось. Действие выполнено успешно.

Заключение

ЗАКЛЮЧЕНИЕ В результате выполнения курсового проекта был разработан про-граммный

продукт для шифрования текстовой информации. Программа пре-образует текст введенный пользователем в шифр. Также программа расшиф-ровывает данный шифр. Результаты работы программы выводятся на экран. В ходе выполнения курсового проекта закреплены и расширены знания по программированию в среде Delphi. При реализации проекта пройдены эта-пы постановки задачи, проектирования программного модуля, кодирования программы на алгоритмический язык и тестирования готовой программы. Цели и задачи курсового проекта выполнены. Созданный программ-ный продукт является актуальным, так как он выполняет все поставленные за-дачи.

Список использованных источников

1. [печатное издание] **Хомоненко, А.Д. и др. Delphi 7 / А.Д. Хомоненко. – Спб.: БХВ-Петербург, 2004. – 1216 с.**
2. [печатное издание] **Фаронов, В.В. Система программирования Delphi. – СПб.: БХВ-Петербург, 2003. – 912 с.**
3. [url] **Ссыслка на программу**
https://drive.google.com/drive/folders/1yhNAKRAjp26YvRpwgbwwSlcT8Sjs4Psb?usp=sharing

Приложения

1. [Приложение] **Приложение** 5ed3b704a087b_записка.doc
2. [Приложение] **Приложение** 5ed3b704c8df9_Титульник.docx
3. [Задание] **Задание** 5ed3c3d510097_kk.docx