|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Информационная безопасность

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОМУ ПРОЕКТУ***

***НА ТЕМУ:***

***Разработка программы по хранению данных почтовых сервисов***

Студент ИУ8-33 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

*2020 г.*

Оглавление

[**Введение** 3](#_Toc26805181)

[**Цель работы** 3](#_Toc26805182)

[**Задачи** 3](#_Toc26805183)

[**Ход работы** 4](#_Toc26805184)

[**Результаты**](#_Toc26805187) 18

[**Литература**](#_Toc26805188) 18

# **Введение**

Предметная область курсового проекта включает в себя теоретический материал из области защиты информации, использования шифра «Кузнечик» и генерации паролей.

**Цель работы**

Целью данной работы является разработка программы для хранения зашифрованных данных от почтовых сервисов в файле с возможностью их расшифровки и вывода на экран.

**Задачи**

В данной работе можно выделить следующие поставленные задачи, необходимые для достижения поставленной цели:

1) Выполнить анализ требований и разработать внешние спецификации;

2) Выполнить проектирование программы;

3) Реализовать программу;

4) Протестировать программу;

**Ход работы**

Теоретическая часть

Хранение, защита и своевременное обновление данных от почтовых ресурсов является основной проблемой безопасности данных пользователей. В данный этап времени большинство программ имеют авторизацию по данным почты или любой другой аналог привязки к почтовому ресурсу. Из-за чего организация и хранение данных от почтовых ресурсов в безопасной и защищенной форме является неотъемлемой частью управления и взаимодействия с почтовыми сервисами и сохранения конфиденциальных данных. Также я сам столкнулся с такой проблемой, так как большинство моих сервисов с авторизацией делило один пароль, который был составлен из чисел и только. В следствие чего происходила утечка паролей от моего почтового ящика, а потом и публичное распространение конфиденциальной информации.

Теперь стоит поговорить о платформе, на которой реализуется описанное выше приложение. Реализовано оно на языке С++, поэтому есть возможность запуска на ПК под управлением разных операционных систем.

Как у любого продукта, у С++ есть свои положительные аспекты и отрицательные. К преимуществам можно отнести:

1)Масштабируемость. На языке C++ разрабатывают программы для самых различных платформ и систем.  
2)Возможность работы на низком уровне с памятью, адресами, портами. Что, при неосторожном использовании, может легко превратиться в недостаток.  
3)Возможность создания обобщенных алгоритмов для разных типов данных, их специализация, и вычисления на этапе компиляции, используя шаблоны.

Практическая часть

Перед объяснением процесса реализации курсового проекта, следует разобраться в основных моментах разработки на С++. Для разработки я использовал IDE Visual Studio 2019 Community. Причиной выбора IDE послужило мое хорошее знакомство со средой разработки, а также удобные инструменты по разработке, анализу, сборке и компиляции кода.

Для реализации я решил использовать макет консольного приложения, т. к. имел горький опыт взаимодействия с интерфейсами. С проектом можно взаимодействовать с помощью команд записываемых в консоль. Весь вывод осуществляется в консоль.

Работа программы начинается с ввода путей до текстовых файлов, содержащих ключ и зашифрованные данные. Файлы ключа читаются моментально и сохраняются в специальном классе ByteBlock;

Дальше выводится приветственная фраза, обозначающая готовность программы к вводу команд. Список доступных команд приведен ниже.

Список команд:

open data file – открывает файл содержащий данные с почтой и паролем в зашифрованном виде и расшифровывает их, записывая в масcив данных, при повторном вызове команды программа предлагает ввести назначение нового положения файла данных для ввода дополнительных значений.

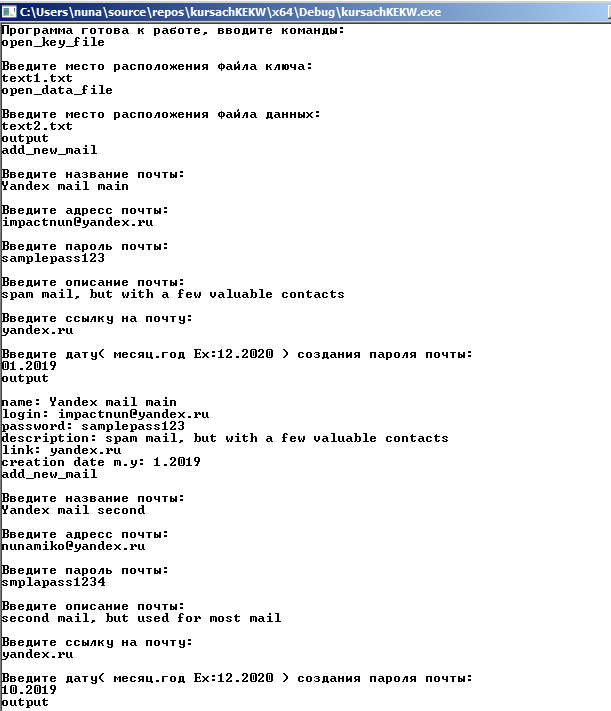
add new mail – добавляет новый элемент в массив почтовых данных, если поле введенных данных пусто, вводится ”<empty>”, чтобы избежать конфликта во время шифрования.

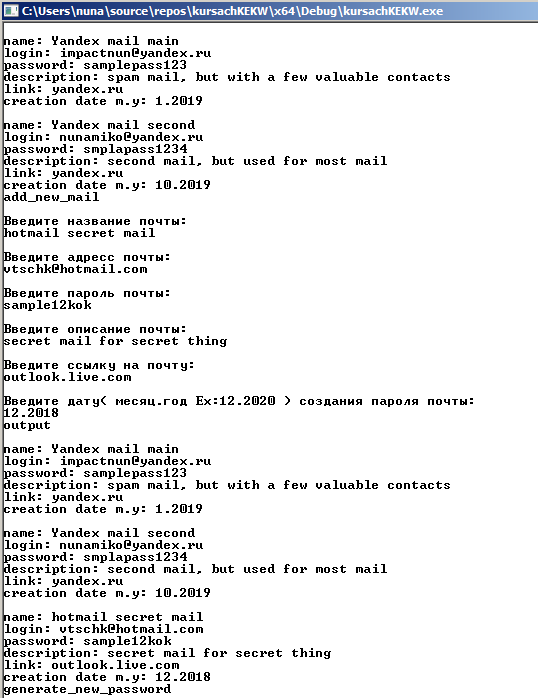
generate new password — генерирует новый пароль для элемента из массива почтовых данных, есть возможность ввести свои символы для дополнения пароля. По умолчанию определены заглавные, прописные буквы Английского алфавита, а также цифры от 0 до 9.

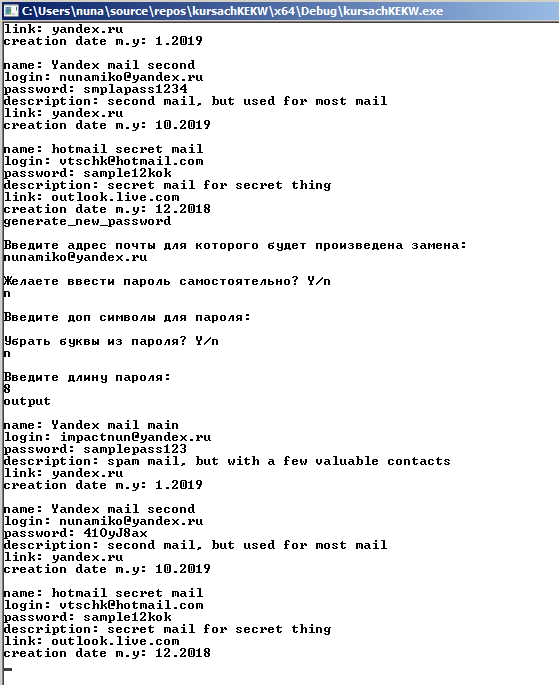
output — выводит содержимое массива почтовых данных на экран при помощи методов класса unit\_class.

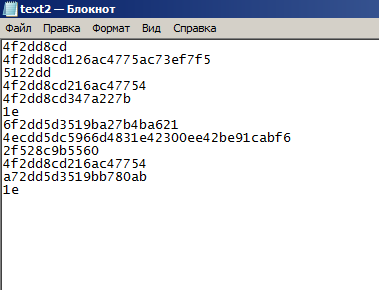
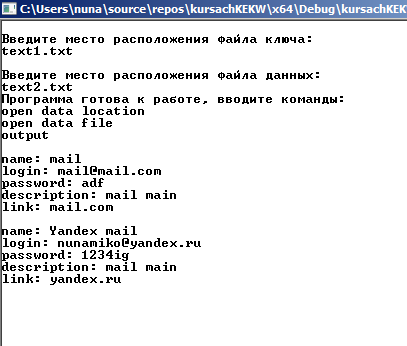
Quit / q *—*  завершает ввод команд и переходит к шифровке и последующему записыванию содержимого массива почтовых данных в исходный файл ввода данных.

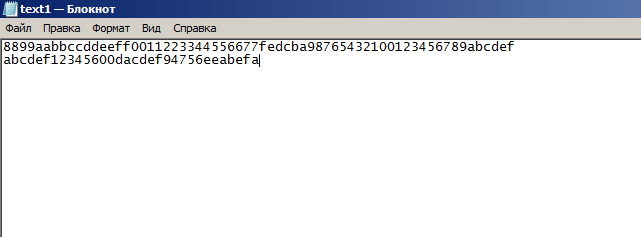
Пример выполнения команд, а также вид текстовых файлов с зашифрованной информацией:

**









Моя программа содержит основной файл выполнения main.cpp

А также дополнительные классы ресурсов и заголовков:

converterofmystupidity.hpp - конвертирует расшифрованные данные перед тем как программа начнет с ними работу и после окончания работы программы методы класса конвертируют строки в подходящий для шифрования вид.

Kyznechik.hpp – блочный шифр Кузнечик(реализует шифрование и расшифровывание кузнечиком для битового блока)

mycrypto.hpp – содержит заголовок класса реализующий хранение данных , а также имеет описание режима обратной связи шифра(превращает блочный шифр в самосинхронизирующийся шифр потока).

unit\_class.hpp – содержит заголовок класса, реализующего хранение данных во время выполнения программы, также имеет функцию генерации пароля.

\*.cpp файлы с аналогичными названиями содержат реализацию данных заголовочных классов.

Теперь я опишу принцип работы шифра кузнечика:

Для написания документации, с помощью приложения Doxygen, необходимым условием является комментирование интерпретируемых файлов на C++. Пример такого комментария можно посмотреть на рисунке 3.2.

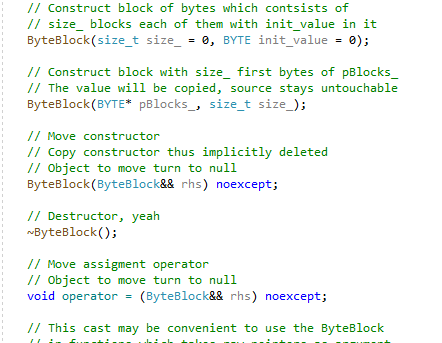


Рисунок 1.2 - пример задокументированного интерпретируемого файла на С++

**Результаты**

Подводя итоги, следует обобщить, что было сделано в ходе курсового проекта. Мною была разработана и реализована программа на платформе для Visual Studio, языке C++. Написаны и успешно пройдены юнит-тесты и интеграционные тесты для проверки игровых моментов. Была сгенерирована документация с помощью Doxygen, опирающаяся на комментарии в интерпретируемых файлах на C#.

**Литература**

[ГОСТ Р 34.12-2015](http://docs.cntd.ru/document/1200121983)«Информационная технология. Криптографическая защита информации. Блочные шифры»

* [ГОСТ 34.12-2018](http://protect.gost.ru/v.aspx?control=7&id=232146)«Информационная технология. Криптографическая защита информации. Блочные шифры»

С. В. Дорохин1,2 , С. С. Качков1,3 , A. A. Сидоренко1,3 1Московский физико-технический институт (государственный университет) 2АО «ПКК Миландр» 3АО «Интел» Реализация блочного шифра «Кузнечик» с иcпользованием векторных инструкций