**Дневник по производственной практике**

*3.07.2023* - Прохождение инструктажа по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте.

*4.07.2023* - Выработка цели и задач исследования в соответствии с индивидуальным заданием: разработка механизма защиты информации и персональных данных.

**Цель:** Обеспечить простое использование программного продукта с возможностью обновления алгоритма кодирования.

**Задачи:**

* + - предусмотреть все системные ошибки, которые могут возникнуть в результате работы сотрудника с приложением и постараться их устранить.
    - Выбрать подходящий метод шифрования с двунаправленной кодировкой.
    - Изучить стандарты возможных систем защиты.

*5.07.2023* - Сбор информации по выбранной области исследования: документации по обеспечению защиты персональных компьютеров сотрудников, алгоритмы шифрования и методы их реализации.

Обзор подходов к решению задачи:

* + - Исследовать основные методы шифрования.
    - Изучить особенности защиты операционной системы компьютеров сотрудников.
    - Выбрать подходящую программную среду для создания приложения.

*6.07.2023 - 12.07.2023* - Разработка методов, алгоритмов, проектирование и реализация компонентов, направленных на решение поставленной задачи.

По результатам исследования был выбран Шифр Плейфера: он имеет простой, но надежный метод кодирования информации с возможностью изменения алгоритма путем генерирования нового “ключа” для шифра, двунаправленный.

# Функций языка C# позволяют создавать надежные и устойчивые приложения [2]:

# [*****Сборка мусора*****](https://learn.microsoft.com/ru-ru/dotnet/standard/garbage-collection/) автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами.

# [*****Типы, допускающие значение null****,***](https://learn.microsoft.com/ru-ru/dotnet/csharp/nullable-references) обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты.

# [*****Обработка исключений*****](https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/exceptions/) предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них.

# [*****Лямбда-выражения*****](https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/lambda-expressions) поддерживают приемы функционального программирования.

# [*****Синтаксис LINQ*****](https://learn.microsoft.com/ru-ru/dotnet/csharp/linq/) создает общий шаблон для работы с данными из любого источника.

# Поддержка языков для  [*****асинхронных операций*****](https://learn.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/async/) предоставляет синтаксис для создания распределенных систем.

# [*****Единая система типов*****](https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/types/)***.*** Все типы C#, включая типы-примитивы, такие как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом.

Интерфейс ПО приложения “Шифратор” представляет собой не сложную структуру и понятную для любого пользователя:

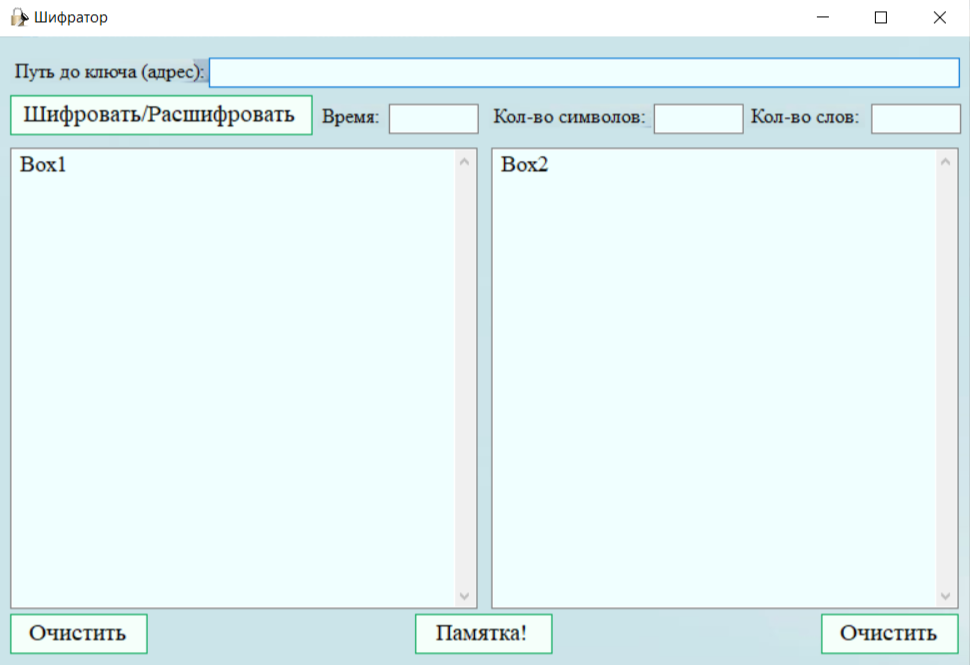


Рисунок 1 - Интерфейс ПО

* В Box1 заполняется текстом для шифрования.
* В Box2 выводится шифрованный текст из Box1.
* Кнопка “Шифровать/Расшифровать” переводить текст из Box1 в Box2 по шифру Плейфера.
* Кнопки “Очистить” стирают информацию из соответствующих над ними Box1 и Box2.
* В строку напротив надписи “Путь до ключа” вводится адрес файла с “ключом” для шифрования, который генерируется отдельным приложением “Генератор”.
* Кнопка “Памятка” содержит руководство по использованию приложения и информацию о нем.
* “Время”, “Кол-во слов”, ”Кол-во символов” отображают статистическую характеристику о выполненной работе.

Приложением “Генератор” для создания “ключей” шифрования:

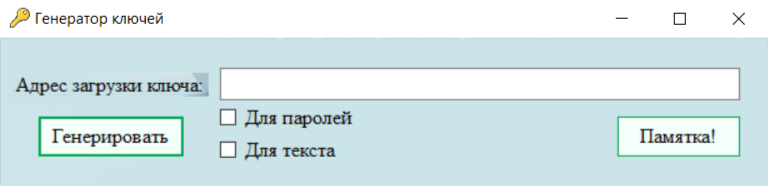


Рисунок 1 - Интерфейс ПО “Генератор ключей”

* Ячейки “Для паролей” и “Для текста” позволяют выбрать тип ключа отдельно для шифрования текста и паролей.
* Строка напротив “Адреса загрузки ключа” принимает адрес файла, куда загружается новый сгенерированный “ключ”.
* Кнопка “Памятка” содержит руководство по использованию приложения и информацию о нем.
* Кнопка “Генерировать” составляет новый “ключ” кодировки текста и перезаписывает его в файл по введенному адресу.

При новом запуске приложения адрес “ключа” заполняется данными, которые были использованы или сгенерированы до этого, что позволяет облегчить работу пользователя.

Также к “Шифратору” присутствует дополнение “Генератор паролей”, который генерирует “новый” пароль путем шифрования “старого” по такому же алгоритму с учетом требований установления паролей систем защиты ОС и различных сайтов.

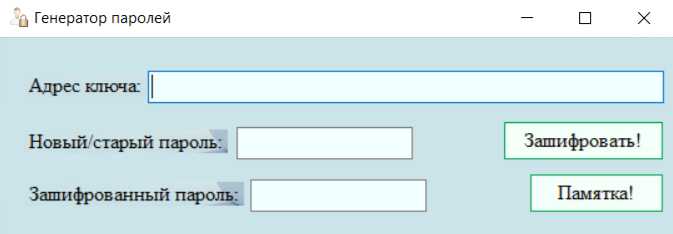


Рисунок 3 - Интерфейс ПО “Генератор паролей”

* Строка напротив надписи “Адрес ключа” принимает путь к файлу, где находиться ключ для шифрования.
* Строка напротив надписи “Новый/Старый пароль” принимает текст произвольного характера, состоящий из букв русского алфавита и цифр от 0до 9.
* Ячейка рядом с “Зашифрованным паролем” отображает отшифрованный текст из ячейки выше.
* Кнопка “Памятка” содержит руководство по использованию приложения и информацию о нем.
* Кнопка “Зашифровать” кодирует введенный пароль и выводит его в ячейку нового пароля.

При введении некорректного пароля отобразится предупреждение о просьбе изменить его. Также, как и в “Генераторе ключей”, адрес ранее введенного или сгенерированного файла с “ключом” отобразиться при повторном запуске приложения.

# *13.07.2023* - Проведение экспериментов и анализ результатов. Экспериментальное исследование качества реализованного решения или тестирование полученного программного продукта.

Сгенерируем “ключи” перед использованием “Шифратора” и “Парольника”: первый ключ – “для пароля” в текстовый файл C:\Users\vebiu\OneDrive\Документы\KeyForParol, второй ключ – “для текста” в файл C:\Users\vebiu\OneDrive\Документы\KeyForText в приложении “Генератор ключей”:

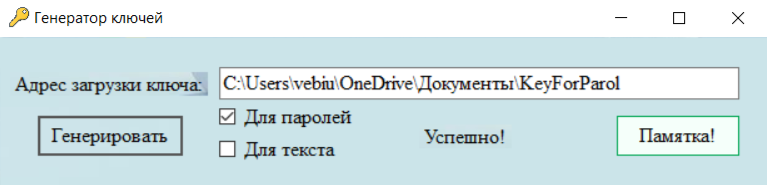


Рисунок 4 - Генерация ключа (пароль)

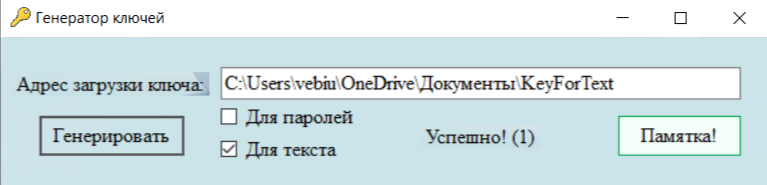


Рисунок 5 - Генерация ключа (текст)

# Запускаем приложение “Шифратор” и введём в Box1 текст произвольного характера: *“На какое дело хочу покуситься и в то же время каких пустяков боюсь! — подумал он с странною улыбкой. — Гм... да... всё в руках человека, и всё-то он мимо носу проносит, единственно от одной трусости... это уж аксиома... Любопытно, чего люди больше всего боятся?”.* Нажимаем кнопку “Шифровать/Расшифровать”*.*

После обработки текст примет такой вид: *“аНк NФ7Zд ле7ДохучД%7ФсуНoыу я иэДотДm еэsZT яФNикBДупуoкявоДъюоуыЖД —оп.сам л7Рс с osNРР7ЛДс[ьъФ7:д— Д3.м..д .а..ДэуXДэр сФхаДFлевоке,аи ДэуXwo7Д7РДTНT7ДР7усД%s7Р7ис,те идснвтнеР7Д7oД7.Р7 йosсу7уoН..дДвo7ДжуДNскН7ам..дДюЛоб%ьoР,оДFге7Д[ЛидДъ7[шь еэуге7ДобQoяс ?”.*

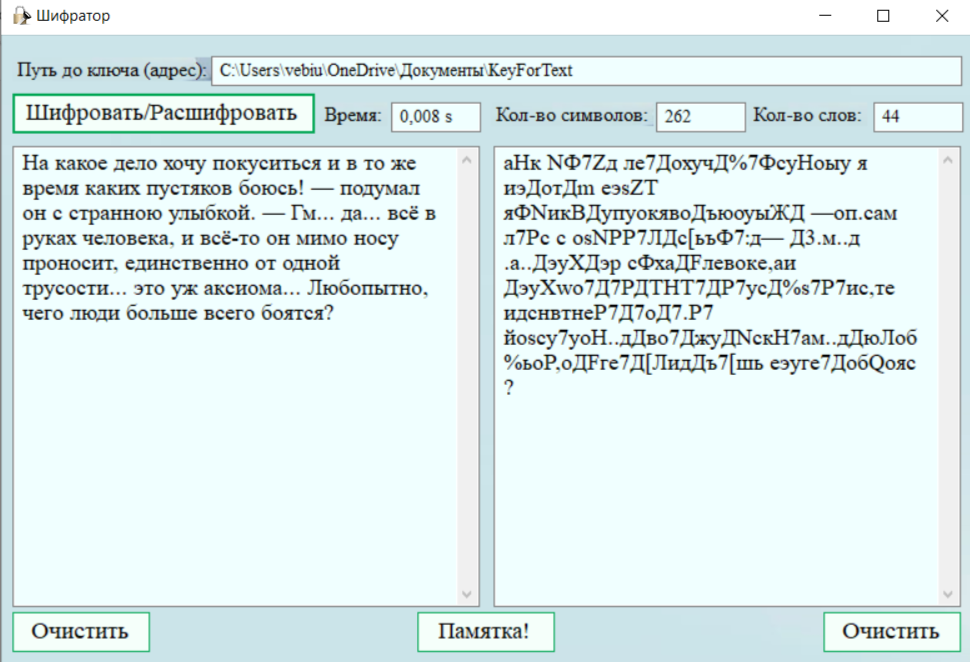


Рисунок 6 - Тест приложения “Шифратор”

Заполним Box1 полученным шифром и попробуем расшифровать его:

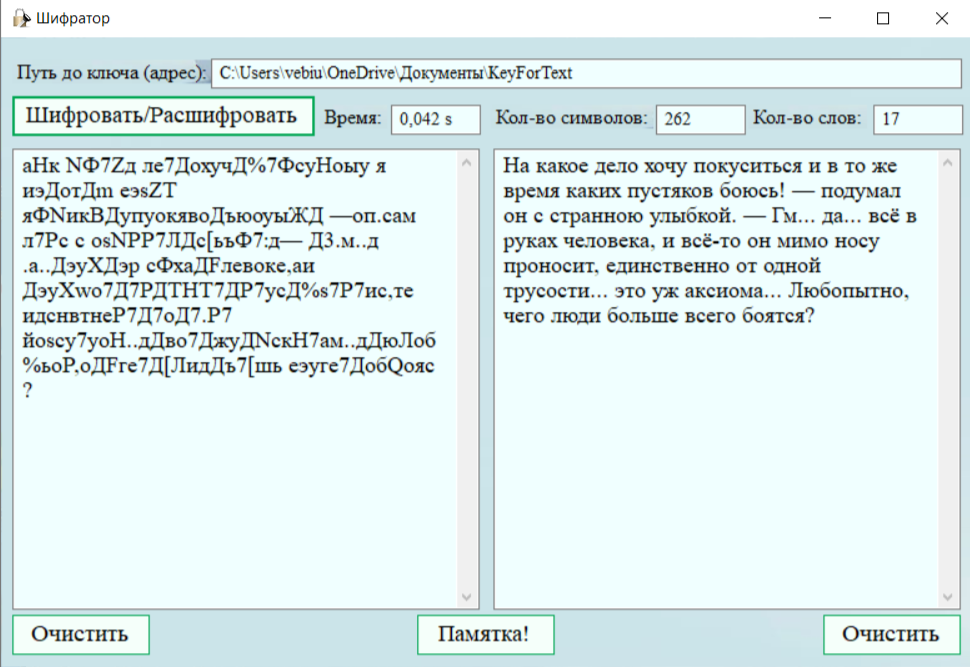


Рисунок 7 - обратная кодировка

Запустим приложение “Парольник” и попробуем сгенерировать новый пароль:

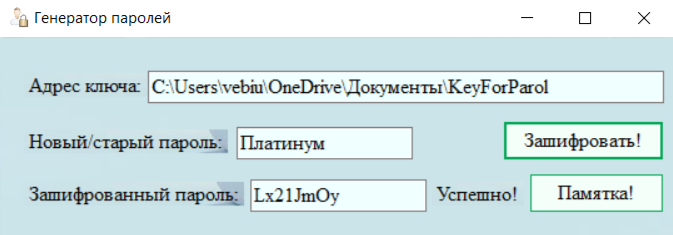


Рисунок 8 - Тест “Парольника”

Как можно увидеть, пароль генерируется из латинских букв большого и малого регистра, а также цифр, что не нарушает политики создания пароля для ОС и различных документов и сайтов.

*14.07.2023* - Подготовка и оформление отчета по практике. Все материалы по отчету вместе с программами и презентацией можно скачать на репозитории github: