Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

(ВятГУ)

**ОТЧЕТ**

**по Производственной практике,**

**технологической (проектно-технологической) практике**

|  |  |  |
| --- | --- | --- |
| Стахиев Александр Сергеевич | | |
| *(Ф.И.О. обучающегося)* | | |
| 01.03.02.52 Прикладная математика и информатика.  Математическое и программное обеспечение информационных систем | | |
| *(направление подготовки (специальность), направленность (профиль))* | | |
|  | | |
| Место прохождения | | Администрация муниципального района |
|  | *(наименование организации, структурного подразделения организации)* | |
| “Корткеросский” | | |
| *(наименование организации, структурного подразделения организации)* | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Итоговая оценка: |  |  | | | | |
| Руководитель практики от университета |  | 15.07.2023 |  |  |  | И.Н. Фищева |
|  |  | *(дата)* |  | *(подпись)* |  | *(Ф.И.О.)* |

Киров, 2023 г.

# Введение

Производственная практика, технологическая (проектно-технологическая) практика проходила в администрации муниципального района “Корткеросский с 3.07.2023 г. по 16.07.2023 г.

**Целью производственной практики** является закрепление и углубление студентами полученных теоретических знаний, приобретение ими практических навыков и компетенций, подготовка к профессиональной деятельности, в основном путём самостоятельного решения задач, сформулированных в рамках индивидуального задания.

**Задачи производственной практики:**

* профессиональная ориентация студентов, формирование полного представления о своей профессии;
  + - решение прикладных задач в области применения современных математических методов и информационных технологий;
    - получение практических навыков и компетенций в организации работы в области применения современных математических методов и информационных технологий
    - приобретение студентами навыков выполнения работ по выбранному направлению в рамках реального производственного или учебного процесса на базе конкретного предприятия, организации или учреждения, включая учебные заведения;
    - получение навыков научно-исследовательской деятельности;
    - получение опыта применения математического аппарата и современных информационных технологий для анализа и решения конкретных производственных задач
    - развитие и закрепление практических навыков по применению пакетов прикладных программ, разработке прикладного и системного программного обеспечения;
    - моделирование производственных, финансовых, управленческих процессов;
    - получение навыков самостоятельного или коллективного решения конкретных профессиональных задач;
    - получение практических навыков в области организации и управления при проведении исследований;
    - анализ возможности внедрения результатов исследований в производственный процесс.

Индивидуальное задание на период прохождения производственной практики – разработка механизма защиты информации и персональных данных.

# Отчет о выполнении индивидуального задания

## Постановка задания:

# Разработать приложение с возможностью шифрования информации и персональных данных сотрудников с понятным интерфейсом и высокой степенью надежности.

## Цель, задачи, которые необходимо решить для достижения поставленной цели:

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.

## Обзор имеющихся подходов к решению

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

## Обоснование выбора методов

По результатам исследования был выбран Шифр Плейфера **[1]**: 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.

## Описание решения

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

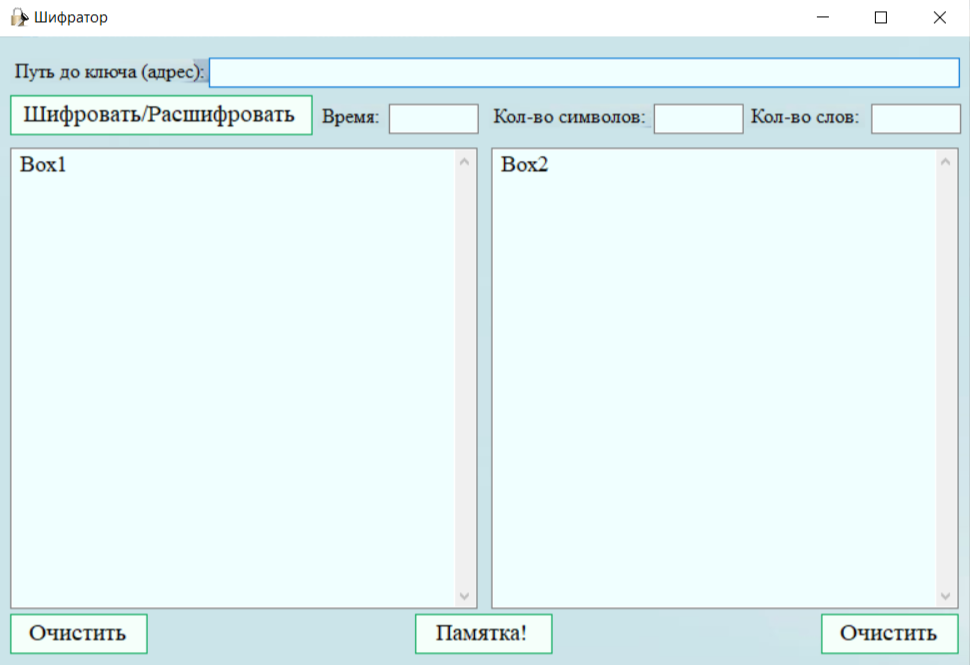


Рисунок 1 - Интерфейс ПО “Шифратора”

* 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.

Перед использованием “Шифратора” необходимо сгенерировать “ключ” для кодирования введенных сообщений. Это можно сделать, воспользовавшись приложением “Генератор”:

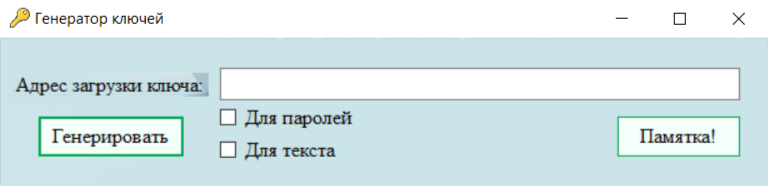


Рисунок 2 - Интерфейс ПО “Генератора ключей”

* 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 “Парольник”, который генерирует “новый” пароль путем шифрования “старого” по такому же алгоритму с учетом требований установления паролей систем защиты ОС и различных сайтов.

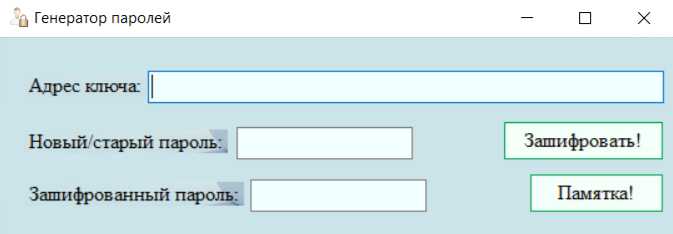


Рисунок 3 - Интерфейс ПО

* 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.

# 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”:

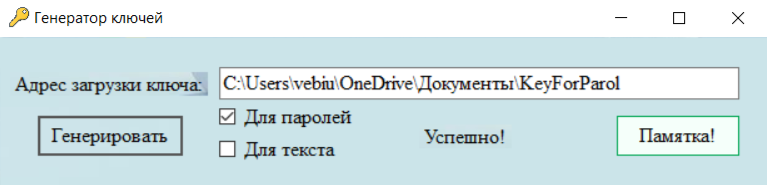


Рисунок 4 - Генерация ключа (пароль)

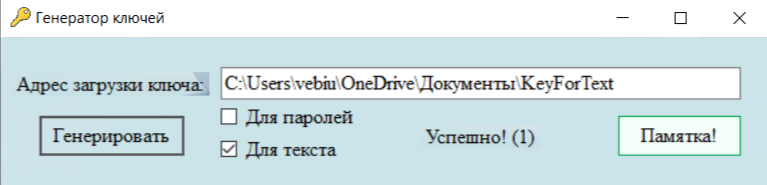


Рисунок 5 - Генерация ключа (текст)

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 ?”.*

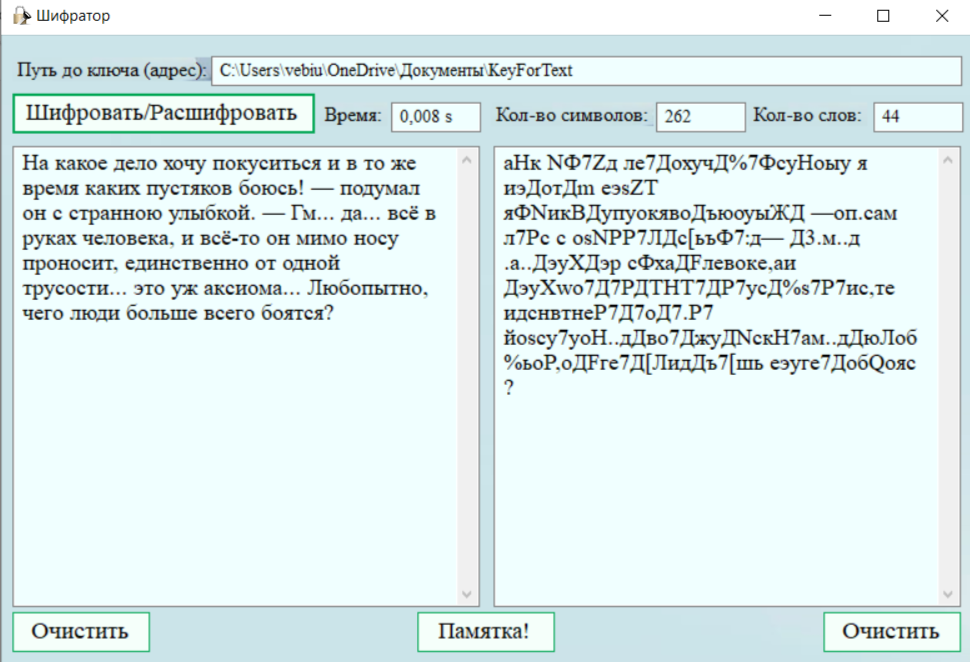


Рисунок 6 - Тест приложения “Шифратор”

Заполним Box1 полученным шифром и попробуем расшифровать его:

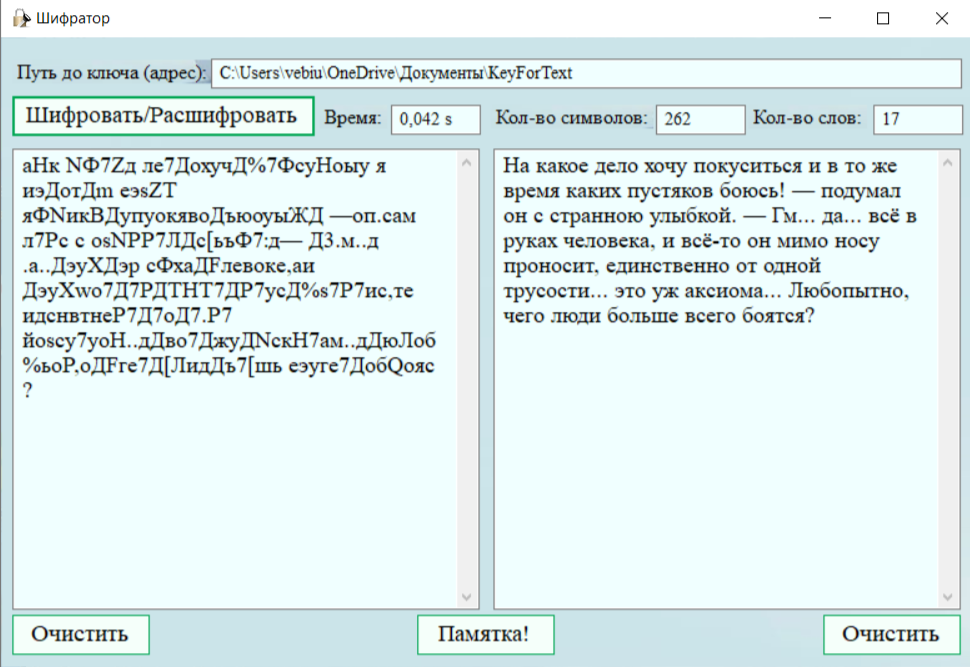


Рисунок 7 - обратная кодировка

Запустим приложение “Парольник” и попробуем сгенерировать новый пароль:

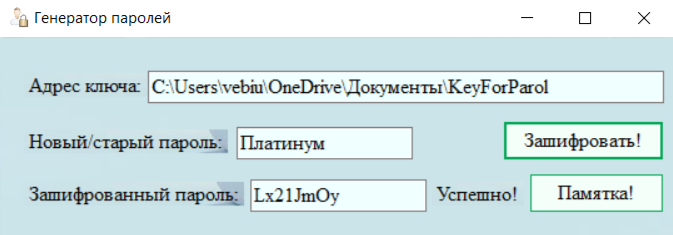


Рисунок 8 - Тест “Парольника”

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.

# Результаты и выводы

В ходе работы было проведено исследование на тему разработки приложений шифрования информации и генерирования паролей для персональных данных сотрудников, что позволило повысить уровень защиты документов от взлома и дальнейшего использования.

## Заключение

Производственная практика способствовала закреплению и углублению полученных теоретических знаний, приобретению практических навыков и компетенций, подготовке к профессиональной деятельности путём самостоятельного решения задач, сформулированных в рамках индивидуального задания.

В результате выполненного исследования показано, что: метод планового обновления системы защиты компьютеров сотрудников был далеко не всегда приветствован положительными отзывами; файлы с конфиденциальной информацией, находящиеся на общей сетевой папке, были общедоступны не только для сотрудников, но и любому злоумышленнику, желающему воспользоваться ими в своих целях.

К достоинствам/недостаткам реализованного программного продукта можно отнести:

* Удобность использования.
* Быструю скорость кодирования.
* Надежный и универсальный алгоритм шифрования с возможностью обновления.

С учетом указанных недостатков

* Ограниченность в количестве типов объектов шифрования.
* При выявлении системных ошибок возможно зависание всего приложения.

## Библиографический список

1. Акимов, О. Е. Дискретная математика: логика, группы, алгоритмы, фракталы / О. Е. Акимов. М., 2005, 656 с.
2. Половко, Бутусов. Программная реализация. Методы и компьютерные технологии их реализации. BHV, 2004 г.
3. Д. Кнут. Искусство программирования. Том 1. Основные алгоритмы. Вильямс, 2019 г.

## Приложение

Все материалы по производственной практике можно найти на репозитории github: