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 | | Республика Коми, корткеросский район, |
|  | *(наименование организации, структурного подразделения организации)* | |
| Администрация муниципального района **«**Корткеросский**»** села Корткерос, Служба автоматизации, информатизации и защиты информации | | |
| *(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.

**Содержание**

[Введение 3](#_Toc140282023)

[1. Отчет о выполнении индивидуального задания 5](#_Toc140282024)

[1.1. Постановка задания: 5](#_Toc140282025)

[1.2. Цель, задачи, которые необходимо решить для достижения поставленной цели: 5](#_Toc140282026)

[1.3. Обзор имеющихся подходов к решению 5](#_Toc140282027)

[1.4. Обоснование выбора методов 5](#_Toc140282028)

[1.5. Описание решения 7](#_Toc140282029)

[2. Экспериментальное исследование качества реализованного решения или тестирование полученного программного продукта 11](#_Toc140282030)

[Результаты и выводы 13](#_Toc140282031)

[Заключение 14](#_Toc140282032)

[Библиографический список 15](#_Toc140282033)

[Приложение 16](#_Toc140282034)

# Введение

Производственная практика, технологическая (проектно-технологическая) практика проходила в администрации муниципального района **«**Корткеросский**»** села Корткерос, служба автоматизации, информатизации и защиты информации 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 текстовой информации служебных документов 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

* + - анализ основных методов шифрования.
    - исследование особенностей защиты операционной системы компьютеров сотрудников.
    - определение наиболее подходящей программной среды для создания приложения.

## Обоснование выбора методов

|  |  |  |  |
| --- | --- | --- | --- |
|  | Простая подстановка | Перестановка по ключу | Гаммирование |
| Методы взлома | Шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифре текста | При своей несложности система уязвима, если злоумышленник имеет шифрованный и соответствующий исходный текст | Метод гаммирования становится бессильным, если злоумышленнику становится известен фрагмент исходного текста и соответствующая ему шифрограмма |
| Параметры ключа | Число возможных ключей мало | Число возможных ключей ограничено | Длина гаммы шифрующей должна быть не менее длины защищаемого сообщения |
| Передача ключа | Отправитель и получатель должны некоторым тайным образом получить копии секретного ключа и сохранять их в тайне | | |
| Стойкость | Имеет низкую стойкость | Имеет более высокую стойкость, чем подстановка | Исходный текст практически невозможно восстановить без ключа |
| Недостатки | Использовать этот метод целесообразно для шифрования только короткого текста | Если ключ используется несколько раз, то его можно проанализировать и взломать | Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей |

По результатам исследования был выбран Шифр Плейфера **[1]**: он имеет простой, но надежный метод кодирования информации с возможностью изменения алгоритма путем генерирования нового **«**ключа**»** постоянной длины, симметричный.

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:

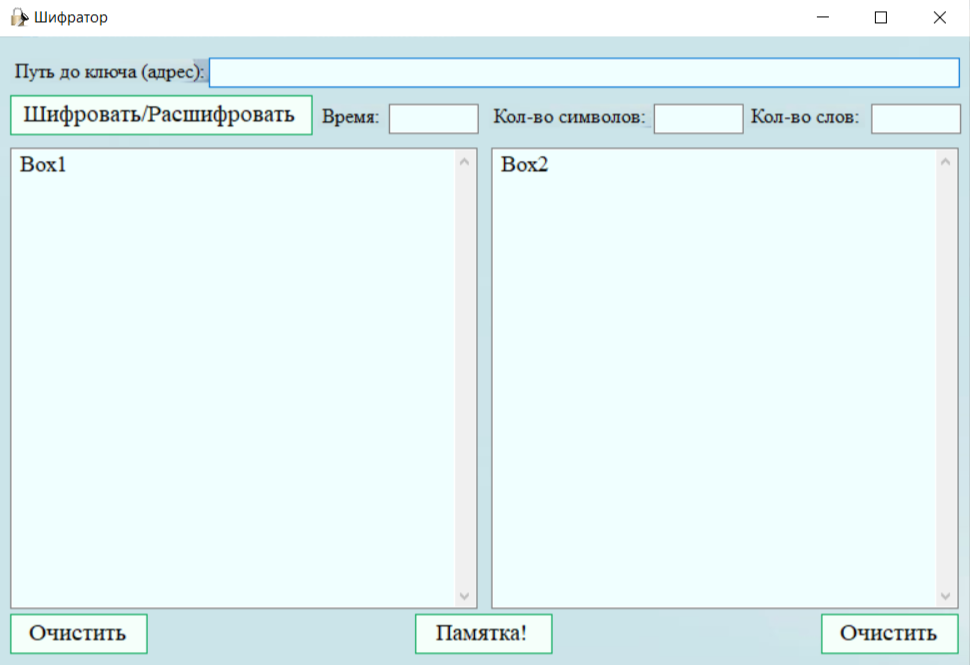


Рисунок 1 - Интерфейс ПО “Шифратора”

* 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 **«**Генераторключей**»**:

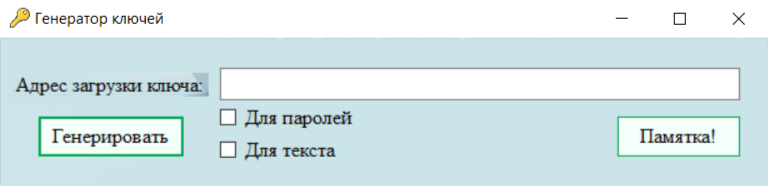


Рисунок 2 - Интерфейс ПО **«**Генератора ключей**»**

* 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.

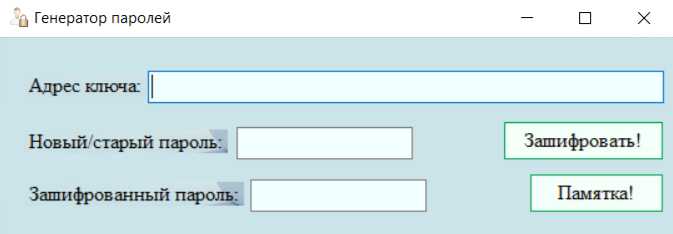


Рисунок 3 - Интерфейс ПО

* 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»:

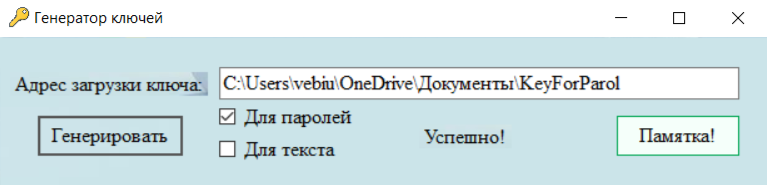


Рисунок 4 - Генерация ключа (пароль)

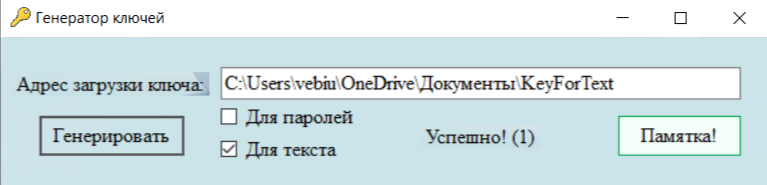


Рисунок 5 - Генерация ключа (текст)

Для контрольного теста проверки мощности «Шифратора» с алгоритмом Плейфера проведем обработку произведения М.A. Булгакова «Мастер и Маргарита», содержащего 748900 символов и 113681 слово. Результат представлен на рисунке 6 и 7. Можно заметить, что времени на шифровку требуется больше, чем на дешифровку.

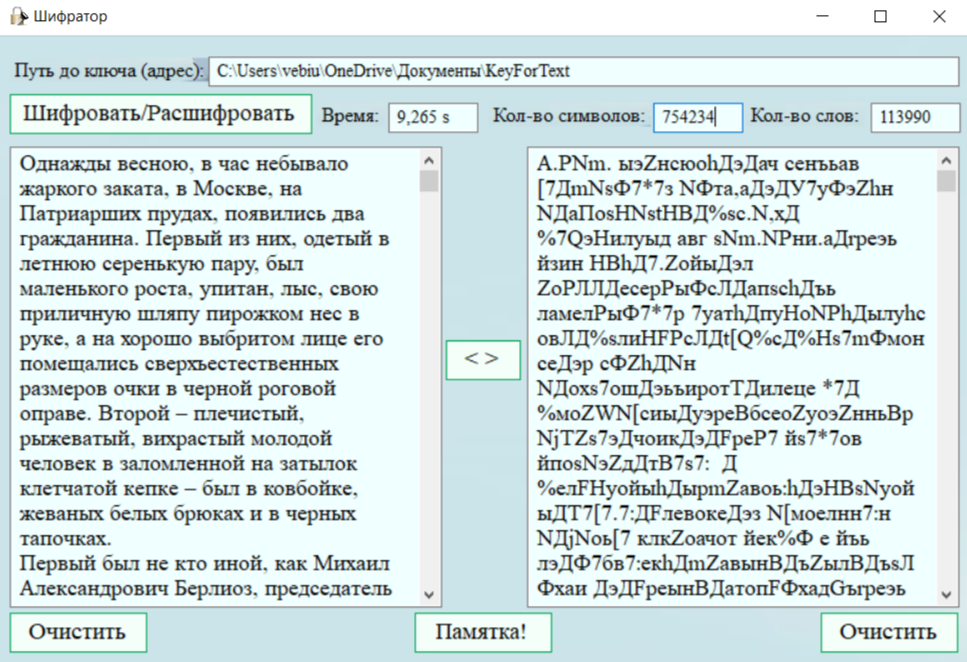


Рисунок 6 - Тест приложения «Шифратор»

Заполним Box1 полученным шифром и попробуем расшифровать его:

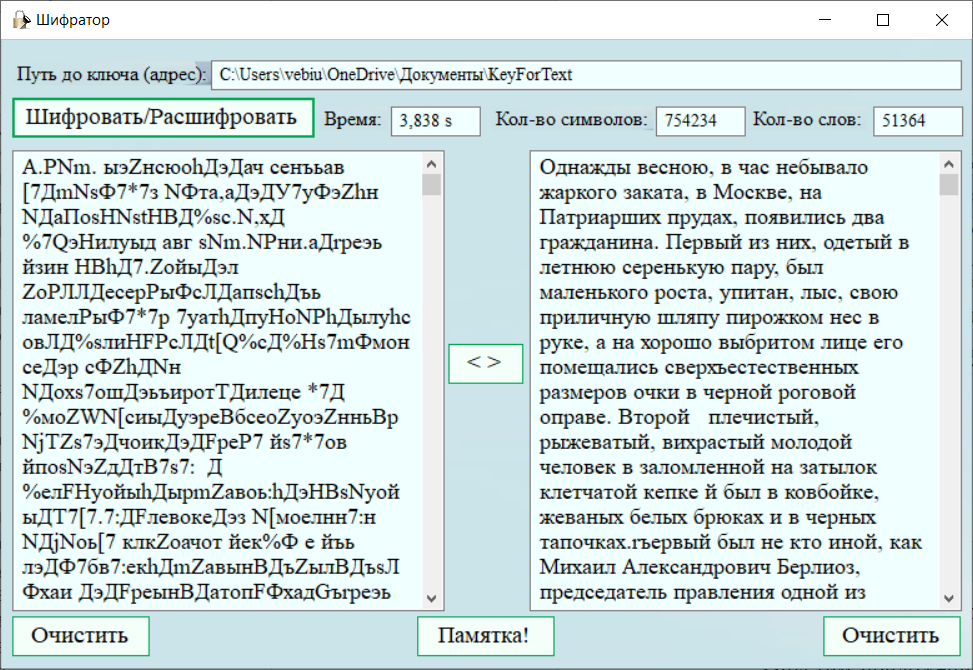


Рисунок 7 - обратная кодировка

# Результаты и выводы

В ходе исследования была проведена работа по разработке приложения шифрования информации и генерирования паролей для персональных данных сотрудников. Новая система защиты данных позволит повысить уровень защиты документов от взлома и дальнейшего вредоносного использования. По окончании практики были приобретены практические навыки и компетенций, проведена подготовка к профессиональной деятельности, в основном путём самостоятельного решения задач, сформулированных в рамках индивидуального задания.

## 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: <https://github.com/sashastahiev/ProisvodPractica.git>