Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»

|  |
| --- |
| **Институт дополнительного профессионального образования** |

|  |  |
| --- | --- |
| Направление подготовки: | 38.03.01 Экономика |
| Профиль подготовки: | Международный бизнес |

|  |
| --- |
| **ОТЧЕТ** |
| о прохождении практики |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель от профильной организации/структурного подразделения СПбГУПТД\*: |  | Бахов Герман Геннадьевич | | | | |  |  |
| *(наименование организации)* |  | *(должность, Ф.И.О., печать организации)* | | | | |  | *(подпись, печать)* |
| Руководитель от СПбГУПТД: |  | должность Фамилия Имя Отчество | | | | |  |  |
|  |  | *(должность, ученая степень / звание, Ф.И.О.)* | | | | |  | *(подпись)* |
| Обучающийся: |  | Семичев Егор Андреевич | | | | |  |  |
|  |  | *(Ф.И.О.)* | | | | |  | *(подпись)* |
| Курс |  | 2 |  | Учебная группа: |  | 2-ЭД-4 | |

Санкт-Петербург

2023

Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»

**Совместный рабочий график (план) проведения практики и индивидуальное задание**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обучающийся | | | | | Семичев Егор Андреевич | | | | | | | | |
|  | | | | | *(Ф.И.О.)* | | | | | | | | |
| Институт | | | Экономики и социальных технологий | | | | | | | | | | |
|  | | | | | *(наименование института)* | | | | | | | | |
| Курс |  | 2 | | Учебная группа | | |  | 2-ЭД-4 | | | Форма обучения |  | очная |
|  |  |  | |  | | |  |  | | |  |  |  |
| Направление подготовки (специальность), профиль | | | | | | | | | | 38.03.01 Экономика,  Международный бизнес | | | |
|  | | | | | | | | |  | | | | |
| Сроки прохождения практики | | | | | | с 10 апреля 2023г. по 30 апреля 2023г. | | | | | | | |
|  | | | | | | *(по календарному учебному графику)* | | | | | | | |

**Совместный рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| Дата | Содержание выполняемых работ и заданий | |
| Общие (типовые вопросы, изучаемые в ходе практики) | | |
|  | **Раздел 1. Проведение обследований предметной области**  **области в рамках решаемой задачи** | |
| Основные методы обследования предметной области | |
| Теоретические аспекты решаемой задачи | |
|  | **Раздел 2. Определение алгоритма решения задачи** | |
| Решение задачи Алгоритмизация | |
| Решение задачи Программный код | |
|  | | **Раздел 3. Написание программного кода (реализация алгоритма)** |
|  | Тестирование программного кода | |
| Оформление отчета | |
| Оформление презентации | |

**Требования по выполнению и оформлению индивидуального задания** Пояснительная записка оформляется в соответствии с требованиямиГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» и ГОСТ Р 7.0.100-2018 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления».

**Вид (ы) отчетных материалов по практике и требования к их оформлению**

**в соответствии с индивидуальным заданием**

1. *Пояснительная записка (отчет)*
2. *Презентация по материалам практики*

Руководитель практики

от СПбГУПТД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Кокорин Е.С.

*(подпись, ф.и.о.)*

Принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Семичев Е.А.

*(подпись, ф.и.о. обучающегося)*

Дата получения обучающимся индивидуального задания 13.04.2023 г.

**РЕФЕРАТ**

Данный отчет содержит 13 страниц, 1 рисунoк. Список использованных источников содержит 6 наименований.

PHYTON, ФУНКЦИЯ, МЕТОДЫ, КОД, ПРОГРАМИРОВАНИЕ, АНАЛИЗ, АЛГОРИТМ, ЗАДАЧА

Цель работы – получение практических навыков в написании программного кода на одном из востребованных языков программирования Phyton.

Задачи:

1. Написать код, способный создавать уникальные пароли, которые можно будет использовать в дальнейшем

2. Описать что и почему работает так, а не иначе

Объект исследования – методика разработки программного

Обеспечения для генерации случайных паролей.

Предмет исследования – инструментарий и возможности средств

объектно-ориентированного программирования Phyton.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc133181120)

[РАЗДЕЛ 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В НАПИСАНИИ КОДА ДЛЯ ГЕНЕРАЦИИ СЛУЧАЙНЫХ ПАРОЛЕЙ 7](#_Toc133181121)

Раздел 1.1 Основные методы обследования предметной области Основные методы обследования предметной области Написания кода для создания случайных паролей………………………………………………..7

[1.2 Теоретические аспекты решаемой задачи в Написании кода для создания случайных паролей…………………………………………………………………………](#_Toc133181123) …….8

[РАЗДЕЛ 2. ОПРЕДЕЛЕНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ 9](#_Toc133181124)

[2.1 Решение задачи алгоритмизации и программный код для создания генератора случайных паролей.. 9](#_Toc133181125)

[РАЗДЕЛ 3. НАПИСАНИЕ ПРОГРАМНОГО КОДА. 11](#_Toc133181126)

[3.1 Тестирование программного кода 11](#_Toc133181127)

[ЗАКЛЮЧЕНИЕ 12](#_Toc133181128)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 13](#_Toc133181129)

ВВЕДЕНИЕ

Каждому из нас хотелось бы, чтобы его конфиденциальность была неприкосновенна. Чтобы данные о количестве трафика и самом трафике не попадали в ненужные руки, а переписки или другая информация, хранящаяся в мессенджерах, которыми мы пользуемся каждый день, не попадала в третьи руки, а уж тем более в руки тех людей, кто может использовать конфиденциальную информацию с целью деанонимизации или прямой угрозы личностному состоянию человека путем манипуляций с использованием личных данных : Фотографий интимного и не только характера, скриншотов переписок и банального доступа к общению с определенными людьми от вашего имени.

Для этого каждый год придумываются все новые и новые способы шифрования данных и создаются алгоритмы создания сложных паролей, которые будет достаточно сложно угадать начинающему или даже среднему потенциальному хакеру.

Опираясь на все вышесказанное для практической работы была выбрана тема создания алгоритма для создания сложных паролей с использованием языка программирования Python. Актуальность этой темы растет, и будет расти все время, что в мире существуют соцсети, мессенджеры и другие сервисы, где каждый из нас создает виртуальный образ себя.

Цель работы – написание кода для генерации сложных паролей с использованием языка программирования Python. Для достижения этой цели необходимо :

1.Разработать в Python код для генерации случайных паролей с определенным количеством символов.

Объектом исследования стала методика создания алгоритмов создания случайных паролей, предназначенных для дальнейшего использования пользователем для защиты собственной конфиденциальности в том числе.

РАЗДЕЛ 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В НАПИСАНИИ КОДА ДЛЯ ГЕНЕРАЦИИ СЛУЧАЙНЫХ ПАРОЛЕЙ

* 1. Основные методы обследования предметной области Основные методы обследования предметной области Написания кода для создания случайных паролей.

Я предоставлю вам основные 8 методов обследования , которые мне удалось составить.

1. Длина пароля: Длина пароля должна быть не менее 12 символов. Для увеличения безопасности пароля, можно увеличить длину пароля до 15 или даже 20 символов.

2. Вариативность: Алгоритм должен быть способен генерировать пароли, состоящие из различных символов, таких как буквы, цифры и символы. Предпочтение отдается случайному выбору символов среди всех возможных.

3. Случайность: Чтобы уменьшить вероятность легкой атаки, следует использовать наиболее надежные методы генерации случайных чисел.

4. Безопасность стойкости пароля: Необходимо убедиться, что пароль, сгенерированный алгоритмом, устойчив к расшифровке, перебору или атаке методами словаря.

5. Уникальность: пароль не должен дублироваться для нескольких пользователей.

6. Удобство использования: Персонифицированный вариант пороля может быть неудобным для использования и/или возможности запомнить приходится записывать или использовать менеджер паролей.

7. Дополнительные проверки: алгоритм должен выполнять дополнительные проверки, такие как проверка на простые или слабые сочетания символов, проверка на наличие последовательных символов и т. д.

8. Стратегия обновления: Алгоритм должен обновляться периодически, чтобы учитывать изменение требований безопасности и надежности пароля.

* 1. Теоретические аспекты решаемой задачи в Написании кода для создания случайных паролей.

Основная задача данного кода – бесприрывное создание уникальных ключ-паролей каждый раз без единого повторения. Каждый раз пароль будет генерироваться в определенном количестве символов (Их будет 15) и ни разу он не может повториться.

Задача кода также может в дальнейшем быть использована в обучении машинного оборудования для создания и

доработки ПО для улучшения и усовершенствования алгоритма генерации случайных паролей.

РАЗДЕЛ 2. ОПРЕДЕЛЕНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

* 1. Решение задачи алгоритмизации и программный код для создания генератора случайных паролей.

Задача заключается в написании кода, при активации которого мы будем получать случайно-сгенерированный пароль из 15 символов. Этот код на языке Python представляет собой функцию для генерации случайного пароля некоторой заданной длины, а затем использует эту функцию для создания пароля длиной 15 символов.

Так как мы используем библиотеку `random`, мы должны импортировать ее в начале скрипта с помощью строки `import random`. Этот модуль содержит различные функции для работы со случайными числами, которые могут использоваться для создания случайных паролей и другой случайной информации.

Далее мы импортируем модуль `string`, который содержит строковые константы, такие как `ascii\_letters`, `digits` и `punctuation`, которые мы будем использовать для генерации пароля.

Затем функция `generate\_password` принимает аргумент `length`, представляющий желаемую длину пароля, который мы хотим создать. Мы используем все символы, которые мы определили ранее, как возможные символы для создания пароля с помощью переменной `possible\_chars`.

Дальше мы используем функцию `random.choice`, чтобы выбрать случайный символ из `possible\_chars`, и повторяем этот выбор для каждого символа пароля в цикле `for`. Результат этого - строка из символов, выбранных случайным образом из `possible\_chars`.

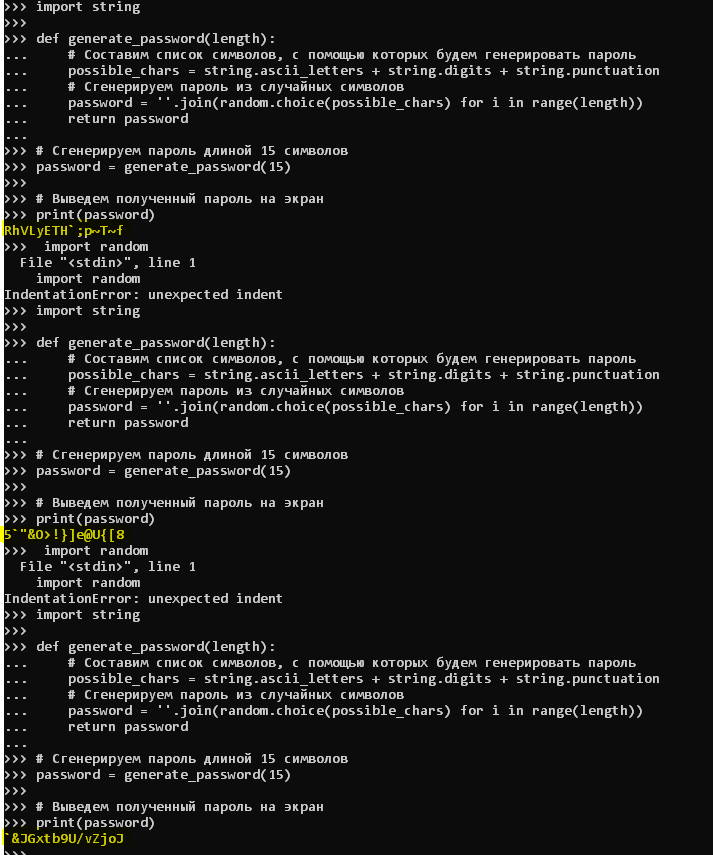
Наконец, мы выводим полученный пароль на экран и возвращаем его с помощью ключевого слова `return`. Вызывая `generate\_password` с аргументом, равным 15, мы создаем пароль длиной 15 символов.

.

РАЗДЕЛ 3. НАПИСАНИЕ ПРОГРАМНОГО КОДА.

* 1. Тестирование программного кода

Пользуясь функциями, которые были разобраны во второй главе, получается такой код:



Выше были рассмотрены 3 возможных пароля после 3-х запусков кода, они выделены желтым цветом.

ЗАКЛЮЧЕНИЕ

В заключение можно сказать, что написание кода для создания генератора случайных паролей на языке Python представляет собой важный аспект разработки безопасного программного обеспечения для защиты конфиденциальности данных. Цель создания генератора пароля - обеспечить случайность, сложность и уникальность пароля, что сделает его устойчивым к различным атакам, путем подбора.

Python предоставляет широкие возможности для написания алгоритмов генерации случайных паролей - от использования встроенных функций и библиотек, таких как `random` и `string`, до более сложных алгоритмов, которые учитывают требования безопасности и стойкости парольной комбинации.

Случайный генератор паролей является важной составляющей безопасности данных и должен использоваться в широком спектре приложений и систем. Надежные пароли - это первый уровень защиты контента данных, а создание эффективных генераторов паролей - это забота как для программистов, так и для конечных пользователей.

Однако, следует отметить, что генераторы паролей не являются панацеей и не могут гарантировать безопасность данных полностью. Вместе с созданием надежных генераторов паролей, всегда необходимо рекомендовать пользователям хранить пароли под рукой, уменьшать использования общих паролей и другие лучшие практики безопасности данных.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Зыков, С. В*.* Программирование. Функциональный подход: учебник и практикум для вузов / С. В. Зыков. — Москва: Издательство Юрайт, 2023. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/512894>
2. Чернышев, С. А*.* Основы программирования на Python: учебное пособие для вузов / С. А. Чернышев. — Москва: Издательство Юрайт, 2023. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519949>
3. Трофимов, В. В*.* Алгоритмизация и программирование: учебник для вузов / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. — Москва: Издательство Юрайт, 2023. — 137 с. — (Высшее образование). — ISBN 978-5-534-07834-3. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513269>
4. [Франсуа Шолле](https://www.litres.ru/author/fransua-sholle-17338303/) - Глубокое обучение на Python
5. Эрик Мэтиз - Изучаем Python. Программирование игр, визуализация данных, веб-приложения.
6. https://www.online-python.com/