**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ГОРОДА МОСКВЫ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ центр профессиональных квалификаций и содействия трудоустройству «ПРОФЕССИОНАЛ»**

**ОТЧЁТ О ВЫПОЛНЕНИИ**

**ИТОГОВОЙ РАБОТЫ**

на тему

**«Разработка программного приложения**

**с использованием инструментов языка программирования Python»**

слушателя Мосина Александра Сергеевича группы № 211-гз

программы повышения квалификации

«Программирование на языке Python»

Москва, 2023 **Постановка задачи**

Задача итоговой работы: разработка программного приложения посредством языка программирования Python.

Назначение приложения: генерация адресов электронной почты, генерация паролей безопасности, валидация данных из файла, сохранение сгенерированной информации.

Программа должна обеспечивать возможность выполнения следующих функций:

* Считывание данных из файла, содержащего сведения о сотрудниках организации.
* Формирование на основе данных файла адресов корпоративной электронной почты для сотрудников организации,
* Генерация паролей безопасности для входа в корпоративную электронную почту,
* Внесение изменений в файл данных и его перезапись.

**Разработка программного приложения**

**Этапы разработки программного приложения.**

Разработка программного приложения заданного функционала включала следующие последовательные этапы:

1. Анализ и исследование задачи.

Подбор переменных, определение последовательности применения операторов и функций.

1. Разработка или выбор алгоритма решения задачи.

Определение типа алгоритма, выбор формы записи, проектирование алгоритма.

1. Разработка программы.

Запись алгоритма на языке Python. Выполнение компоновки последовательностей операторов и функций на языке Python.

1. Тестирование работоспособности программы (отладка).

Проверка программного кода на работоспособность с последующим устранением ошибок программного кода при их выявлении.

1. Анализ результатов.

Выявление соответствия функционала разрабатываемого программного приложения поставленной задаче.

Листинг 1. Исходный программный код.

import re

import zipfile

import random

import string

# 1. Открытие ZIP-архива и извлечение файла

with zipfile.ZipFile('task\_file.txt.zip', 'r') as zip\_file:

zip\_file.extractall()

file\_path = zip\_file.extract('task\_file.txt')

# 1.1 Чтение данных из файла

lst = []

with open('task\_file.txt', 'r') as f:

for line in f:

lst.append(line.rstrip().split(","))

print(f"Всего строк с данными в исходном файле = {len(lst) - 1}")

# 1.2 Валидация данных

result\_lst = []

error\_lst = []

for i in range(1, len(lst)):

if re.match(r"^[A-Z]{1}[a-z]{2,}", lst[i][1].strip()) \

and re.match(r"^[A-Z]{1}[a-z]{2,}", lst[i][2].strip())\

and len(lst[i][3].strip()) >= 7\

and not re.match(r"^[0]{1,}", lst[i][3].strip())\

and lst[i][3].strip().isdigit():

result\_lst.append(lst[i])

else:

error\_lst.append(lst[i])

for i in range(len(result\_lst)):

for j in range(5):

result\_lst[i][j] = result\_lst[i][j].replace(" ", "")

print(f"Всего строк с корректными данными = {len(result\_lst)}")

print(f"Всего строк с ошибками = {len(error\_lst)}")

# 2. Формирование почтовых адресов на основе ФИО

*def email\_gen(list\_of\_names):*

*emails = []*

*for i in list\_of\_names:*

*letter = 1*

*while i[2] + '.' + i[1][0:letter] + '@company.io' in emails:*

*letter += 1*

*emails.append(i[2] + '.' + i[1][0:letter] + '@company.io')*

*return emails*

*emails = email\_gen(result\_lst)*

# 3. Генерация паролей безопасности

def generate\_password(letters=8, digits=2, punctuation=2):

password = list(random.choice(string.ascii\_letters) for i in range(letters))

password += list(random.choice(string.digits) for i in range(digits))

password += list(random.choice('!#$%^&\*()\_-+=[]{}\|/?><.') for i in range(punctuation))

password = random.sample(password, len(password))

return ''.join(password)

# 4. Сохранение информации в файл

# 4.1. Сохранение данных с ошибками

with open('error\_file.txt', 'wt') as f:

print("EMAIL, NAME, LAST\_NAME, TEL, CITY", file=f)

for i in range(len(error\_lst)):

print(f"{error\_lst[i][0]}, {error\_lst[i][1]}, {error\_lst[i][2]}, {error\_lst[i][3]}, {error\_lst[i][4]}", file=f)

print(f"Данные с ошибками сохранены в файл error\_file.txt, всего строк = {len(error\_lst)}")

# 4.2. Сохранение валидных данных с e-mail и паролями

with open('result\_file.txt', 'wt') as f:

print("EMAIL, PASSWORD, NAME, LAST\_NAME, TEL, CITY", file=f)

for i in range(len(result\_lst)):

print(f"{emails[i]}, {generate\_password()}, {result\_lst[i][1]}, {result\_lst[i][2]}, {result\_lst[i][3]}, {result\_lst[i][4]}", file=f)

print(f"Данные с e-mail и password сохранены в файл result\_file.txt, всего строк = {len(result\_lst)}")

**Вывод о работоспособности программного приложения.**

По результату выполнения этапа отладки программного кода сделан вывод, что разработанный исходный код работоспособен, выполнен без синтаксических, семантических и логических ошибок.

**Список литературы:**

1. Нормативные правовые акты:

1. Приказ Министерства труда и соц. Защиты РФ от 18 ноября 2013 г. №679н ''Об утверждении профессионального стандарта 'Программист' (с изменениями и дополнениями).

2. Учебники и учебные пособия:

* + - 1. Гуриков С.Р. Основы алгоритмизации и программирования на Python. Учебное пособие. – М.: Инфра-М, Форум, 2018. - 707 с.
      2. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: пер с англ. – СПб.: БХВ-Петербург. 2019. – 768 с.
      3. Фёдоров Д.Ю. Программирование на языке высокого уровня Python. Учебное пособие для прикладного бакалавриата. – М.: Изд. Юрайт. 2019. -161 с.
      4. Портал <https://habr.com>
      5. Форум программистов: <https://ru.stackoverflow.com>
      6. Лекции от ГБОУ ДПО Центр «Профессионал»
      7. Вебинары от Божко А.В.