

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №18
дисциплины «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Работа с файлами в языке Python. Виртуальные окружения.

Цель работы: приобретение навыков по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командой строки.

Ход работы.

1. Создание нового репозитория с лицензией MIT.

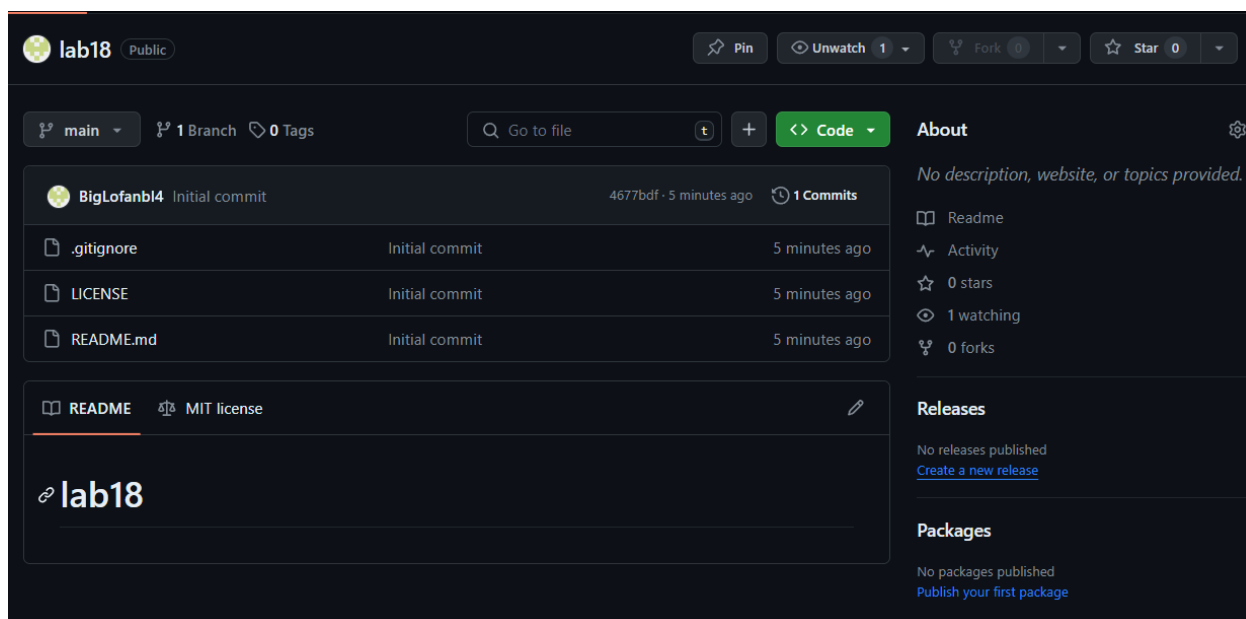


Рисунок 1.1 – Созданный репозиторий

2. Создание виртуального окружения, установка flake8.

```
(base) C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18>conda create -n lab18 python
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: D:\Miniconda\envs\lab18

added / updated specs:
- python

The following packages will be downloaded:



| package      | build      |        |
|--------------|------------|--------|
| tzdata-2024a | h04d1e81_0 | 116 KB |
| Total:       |            | 116 KB |



The following NEW packages will be INSTALLED:

bzip2             pkgs/main/win-64::bzip2-1.0.8-he774522_0
ca-certificates  pkgs/main/win-64::ca-certificates-2023.12.12-haa95532_0
expat             pkgs/main/win-64::expat-2.5.0-hd77b12b_0
libffi           pkgs/main/win-64::libffi-3.4.4-hd77b12b_0
openssl          pkgs/main/win-64::openssl-3.0.13-h2bbff1b_0
pip              pkgs/main/win-64::pip-23.3.1-py312haa95532_0
python           pkgs/main/win-64::python-3.12.1-h1d929f7_0
setuptools       pkgs/main/win-64::setuptools-68.2.2-py312haa95532_0
sqlite           pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
tk               pkgs/main/win-64::tk-8.6.12-h2bbff1b_0
tzdata          pkgs/main/noarch::tzdata-2024a-h04d1e81_0
vc               pkgs/main/win-64::vc-14.2-h21ffa51_1
vs2015_runtime  pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel            pkgs/main/win-64::wheel-0.41.2-py312haa95532_0
xz               pkgs/main/win-64::xz-5.4.5-h8cc25b3_0
zlib             pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0

Proceed ([y]/n)? y

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate lab18
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18>conda activate lab18
(lab18) C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18>
```

Рисунок 2.1 – Создание и активация виртуального окружения

```
(lab18) C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18>conda install flake8
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: D:\Miniconda\envs\lab18

added / updated specs:
- flake8

The following packages will be downloaded:



| package            | build           |        |
|--------------------|-----------------|--------|
| flake8-6.1.0       | py312haa95532_0 | 135 KB |
| mccabe-0.7.0       | pyhd3eb1b0_0    | 11 KB  |
| pycodestyle-2.11.1 | py312haa95532_0 | 94 KB  |
| pyflakes-3.1.0     | py312haa95532_0 | 168 KB |
| Total:             |                 | 407 KB |



The following NEW packages will be INSTALLED:

flake8            pkgs/main/win-64::flake8-6.1.0-py312haa95532_0
mccabe            pkgs/main/noarch::mccabe-0.7.0-pyhd3eb1b0_0
pycodestyle       pkgs/main/win-64::pycodestyle-2.11.1-py312haa95532_0
pyflakes          pkgs/main/win-64::pyflakes-3.1.0-py312haa95532_0

Proceed ([y]/n)? y

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Рисунок 2.2 – Установка flake8

3. Выполнение примеров из лабораторной работы.

```
5 # 1 пример
6 with open("file.txt", "r", encoding="utf-8") as f:
7     sentences = f.readlines()
8 # Вывод предложений с запятыми.
9 for sentence in sentences:
10     if "," in sentence:
11         print(sentence)
12
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> flake8 examples.py
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> python examples.py
7 дней в неделе, воскресенье - самый лучший из них.
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> █

Рисунок 3.1 – Код и вывод первого примера

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import secrets
5 import string
6 import sys
7
8 if __name__ == "__main__":
9     # 1 пример
10     # with open("file.txt", "r", encoding="utf-8") as f:
11     #     sentences = f.readlines()
12     # # Вывод предложений с запятыми.
13     # for sentence in sentences:
14     #     if "," in sentence:
15     #         print(sentence)
16
17     # 2 пример
18     if len(sys.argv) != 2:
19         print("The password length is not given!", file=sys.stderr)
20         sys.exit(1)
21
22     chars = string.ascii_letters + string.punctuation + string.digits
23     length_pwd = int(sys.argv[1])
24
25     result = []
26     for _ in range(length_pwd):
27         idx = secrets.SystemRandom().randrange(len(chars))
28         result.append(chars[idx])
29
30     print(f"Secret Password: {''.join(result)}")
31
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> flake8 examples.py
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> python examples.py
The password length is not given!
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> python examples.py 12
Secret Password: p~af_{\$^FV/
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> █

Рисунок 3.2 – Код и вывод второго примера

4. Задание 1: написать программу, которая считывает текст из файла и выводит его на экран, заменив цифры от 0 до 9 на слова «ноль», «один», ..., «девять», начиная каждое предложение с новой строки.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == "__main__":
    digits_words = {
        "0": "ноль",
        "1": "один",
        "2": "два",
        "3": "три",
        "4": "четыре",
        "5": "пять",
        "6": "шесть",
        "7": "семь",
        "8": "восемь",
        "9": "девять",
    }

    with open("file.txt", "r", encoding="utf-8") as fileptr:
        sentences = fileptr.readlines()

    for sentence in sentences:
        sentence = sentence.strip()

        for digit, word in digits_words.items():
            sentence = sentence.replace(digit, word)
        print(sentence)
```

Рисунок 4.1 – Код программы

```
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> flake8 ex1.py
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> python ex1.py
ноль часов - время начала нового дня.
один красный цвет сияет на закате.
В два руках я держу ключи от своего сердца.
три желания у меня на рассвете.
четыре стороны мира встречаются в одной точке.
пять звезд сверкают на ночном небе.
шесть ступеней ведут на вершину горы.
семь дней в неделе, воскресенье - самый лучший из них.
восемь нот звучат в моей мелодии.
девять роз расцветают в моем саду.
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> █
```

Рисунок 4.2 – Вывод программы

5. Задание 2: Проверка орфографии – лишь составная часть расширенного текстового анализа на предмет наличия ошибок. Одной из самых распространенных ошибок в текстах является повторение слов. Например, автор может по ошибке дважды подряд написать одно слово. Некоторые текстовые процессоры умеют распознавать такой вид ошибок при выполнении текстового анализа. В данном упражнении вам предстоит написать программу для определения наличия дублей слов в тексте. При нахождении повтора на экран должен выводиться номер строки и дублирующееся слово. Удостоверьтесь, что программа корректно обрабатывает случаи, когда повторяющиеся слова находятся на разных строках. Имя файла для анализа должно быть передано программе в качестве единственного аргумента командной строки. При отсутствии аргумента или невозможности открыть указанный файл на экране должно появляться соответствующее сообщение об ошибке.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == "__main__":
    if (len(sys.argv) != 2):
        print("FileName is not given!", file=sys.stderr)
        sys.exit(1)
    file_name = sys.argv[1]

    with open(file_name, "r", encoding="utf-8") as fileptr:
        sentences = fileptr.readlines()

    for line_num, sentence in enumerate(sentences):
        sentence_words = sentence.split(" ")
        prev_word = ""

        for curr_word in sentence_words:
            if (prev_word == curr_word):
                print(f"Line: {line_num + 1}, repeating word: {curr_word}")
            prev_word = curr_word
```

Рисунок 5.1 – Код программы

```
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> flake8 ex2.py
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> python ex2.py file2.txt
Line: 1, repeating word: люблю
Line: 1, repeating word: парк
Line: 2, repeating word: время
Line: 3, repeating word: уют
Line: 4, repeating word: фонтана
Line: 4, repeating word: воды
Line: 5, repeating word: время
PS C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> █
```

Рисунок 5.2 – Вывод программы

6. Подобранный задача: создать скрипт на Python, который автоматизирует управление файлами на компьютере с помощью функций модуля `os`. Скрипт выполняет следующие функции:

1. Переместиться в рабочую директорию (путь рабочей директории указывается как аргумент командой строки), где будет проводиться управление файлами.

2. Создать поддиректории 'Images', 'Docs', 'Audio' и 'Others' для сортировки файлов.

3. Перенести файлы из рабочей директории в соответствующие поддиректории на основе их расширений. Например, изображения (.jpg, .png) должны быть перемещены в 'Images', документы (.pdf, .txt) - в 'Docs', аудиофайлы (.mp3, .wav) - в 'Audio'. Все остальные файлы должны быть перемещены в 'Others'.

4. Удалить пустую директорию 'Temp', если она существует в рабочей директории.

5. В конце скрипта вывести полный путь к рабочей директории.

```

import os
import sys

if __name__ == "__main__":

    if len(sys.argv) != 2:
        print("Word directory path is not given!", file=sys.stderr)
        sys.exit(1)

    work_directory = sys.argv[1]
    os.chdir(work_directory)

    file_map = {
        ".jpg": "Images",
        ".png": "Images",
        ".mp3": "Audio",
        ".wav": "Audio",
        ".txt": "Doc",
        ".pdf": "Doc",
    }

    for folder in ["Images", "Audio", "Doc", "Others"]:
        if not os.path.exists(folder):
            os.mkdir(folder)

    for file_name in os.listdir("."):
        if not os.path.isfile(file_name):
            continue
        file_ext = os.path.splitext(file_name)[1]
        os.rename(file_name, f'{file_map.get(file_ext, "Others")}/{file_name}')

    if os.path.exists("Temp"):
        os.rmdir("Temp")

    print(f"Word directory is {os.getcwd()}")

```

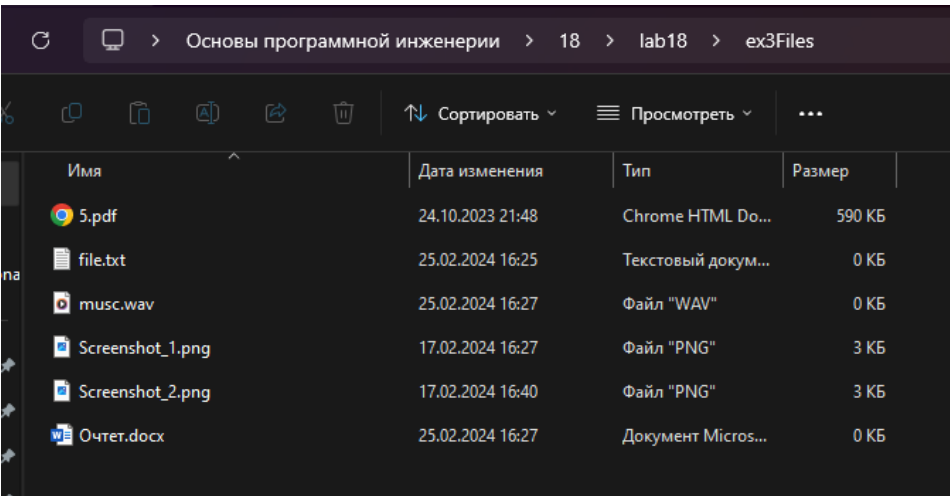
Рисунок 6.1 – Код программы

```

PS C:\Users\vlad\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> flake8 ex3.py
PS C:\Users\vlad\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18> python ex3.py "C:\Users\vlad\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18\ex3Files"
Word directory is C:\Users\vlad\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18\ex3Files
PS C:\Users\vlad\OneDrive\Рабочий стол\Основы программной инженерии\18\lab18>

```

Рисунок 6.2 – Вывод программы



Имя	Дата изменения	Тип	Размер
5.pdf	24.10.2023 21:48	Chrome HTML Do...	590 КБ
file.txt	25.02.2024 16:25	Текстовый докум...	0 КБ
musc.wav	25.02.2024 16:27	Файл "WAV"	0 КБ
Screenshot_1.png	17.02.2024 16:27	Файл "PNG"	3 КБ
Screenshot_2.png	17.02.2024 16:40	Файл "PNG"	3 КБ
Отчет.docx	25.02.2024 16:27	Документ Micros...	0 КБ

Рисунок 6.3 – Папка «ex3Files» до выполнения скрипта

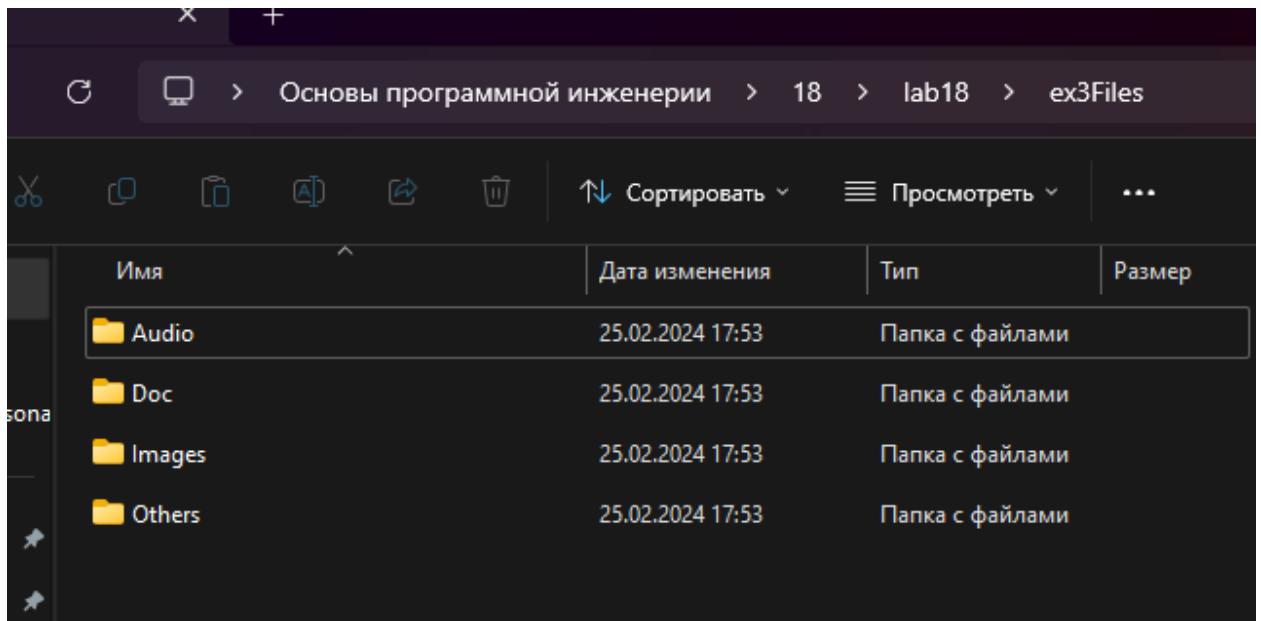


Рисунок 6.4 - Папка «ex3Files» после выполнения скрипта

Ответы на контрольные вопросы

1. Открыть файл только для чтения:

```
file = open('file.txt', 'r')
```

2. Открыть файл только для записи:

```
file = open('file.txt', 'w')
```

Здесь 'w' означает режим записи ('write'), который создаст файл, если он не существует, или перезапишет существующий.

3. Чтение данных из файла:

```
content = file.read() # Читает весь файл в строку
```

```
line = file.readline() # Читает одну строку из файла
```

```
lines = file.readlines() # Читает весь файл в список строк
```

4. Запись данных в файл:

```
file.write('Hello, World!') # Записывает строку в файл
```

5. Закрывать файл:

```
file.close()
```

6. Конструкция «with ... as»:

«with ... as» является менеджером контекста и предназначена для обеспечения правильного использования ресурсов, которые требуют «закрытия» или особого освобождения в конце выполнения блока.

Конструкция «with ... as» может быть использована с другими типами ресурсов, такими как подключения к базам данных, блокировки потоков и др.

7. Дополнительные методы чтения/записи:

В документации Python указаны другие методы, такие как «file.read(size)» для чтения данных определенного размера, «file.readline(size)» для чтения строки с ограничением длины и «file.write(str)» для записи. Есть также буферизированные версии чтения и записи, такие как «file.buffer.read()» и «file.buffer.write()».

8. Дополнительные функции модуля «os»:

Помимо уже рассмотренных функций, модуль «os» также предоставляет:

- «os.listdir()» - получение списка файлов и директорий в указанной директории.
- «os.stat()» - получение информации о файле.
- «os.walk()» - генерация имен файлов в дереве каталогов путем обхода дерева сверху вниз или снизу вверх.
- «os.path.join()» - соединение путей с учетом особенностей операционной системы.
- «os.path.exists()» - проверка существования пути.
- «os.path.isfile()» - проверка, является ли путь файлом.
- «os.path.isdir()» - проверка, является ли путь директорией.
- «os.path.getsize()» - получение размера файла.

И много других функций для работы с путями, директориями и файлами.