

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:

Джараян Арег Александрович  
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: Лабораторная работа 2.15 Работа с файлами в языке Python

**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 командной строки.

Ход работы.

1. Создание нового репозитория с лицензией MIT.
2. Клонировал репозиторий на рабочий ПК.
3. Дополнил файл .gitignore необходимыми инструкциями.

```
1  # Byte-compiled / optimized / DLL files
2  __pycache__/
3  *.py[cod]
4  *$py.class
5
6  # C extensions
7  *.so
8
9  # Distribution / packaging
10 .Python
11 build/
12 develop-eggs/
13 dist/
14 downloads/
15 eggs/
16 .eggs/
17 lib/
18 lib64/
19 parts/
20 sdist/
21 var/
22 wheels/
23 share/python-wheels/
```

Рисунок 1 – Файл .gitignore

```
aregd@DESKTOP-5KV9QA9 MINGW64 /d/Рабочий стол/4 семестр/опи/17/17 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

aregd@DESKTOP-5KV9QA9 MINGW64 /d/Рабочий стол/4 семестр/опи/17/17 (develop)
$
```

Рисунок 4 – организация ветки

4. Выполнил примеры лабораторной работы.

5. Задание 1. Составить программу с использованием списков и словарей для решения задачи. Номер варианта определяется по согласованию с преподавателем. Исходный файл, из которого выполняется чтение, необходимо также добавить в репозиторий, каждое предложение в файле должно находиться на отдельной строке.

Задание 2. В данном упражнении вы должны написать программу, которая будет находить самое длинное слово в файле. В качестве результата программа должна выводить на экран длину самого длинного слова и все слова такой длины. Для простоты принимайте за значимые буквы любые непробельные символы, включая цифры и знаки препинания.

Код:

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

def help1():
    """
    Функция для вывода списка команд
    """
    print("Список команд:\n")
    print("r1 - Менять слова местами;")
    print("r2 - Найти самые длинные слова из файла;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

def error1():
    """
    функция для неопознанных команд
    """
    print(f"Неизвестная команда {command}")

def r(filename):
    """
    читает файл и передаёт дальше список
    """
    j = []
    with open(filename, "r") as f:
        for line in f:
            j.extend(line.split())
```

```

        return j

def r1(filename):
    """
    Обращается к функции r() и дальше выводит слова файла меняя их местами
    """
    j = r(filename)

    for i, x in enumerate(j[:-1:2]):
        j[i * 2], j[i * 2 + 1] = j[i * 2 + 1], j[i * 2]

    print(*j)

def r2(filename):
    """
    Обращается к функции r() и дальше находит самые длинные слова
    """
    j = r(filename)
    a = max(len(word) for word in j)
    k = [word for word in j if len(word) == a]
    print(k, a)

if __name__ == "__main__":
    help1()

    while True:
        # Запросить команду из терминала.
        command = input(">>> ")
        # Выполнить действие в соответствие с командой.

        match command:
            case "exit":
                break

            case "r1":
                r1("asd.txt")

            case "r2":
                r2("asd.txt")

            case "help":
                help1()

            case _:
                error1()

```

Рисунок 5 – код для выполнения программ

```

Список команд:

r1 - Менять слова местами;
r2 - Найти самые длинные слова из файла;
help - отобразить справку;
exit - завершить работу с программой.

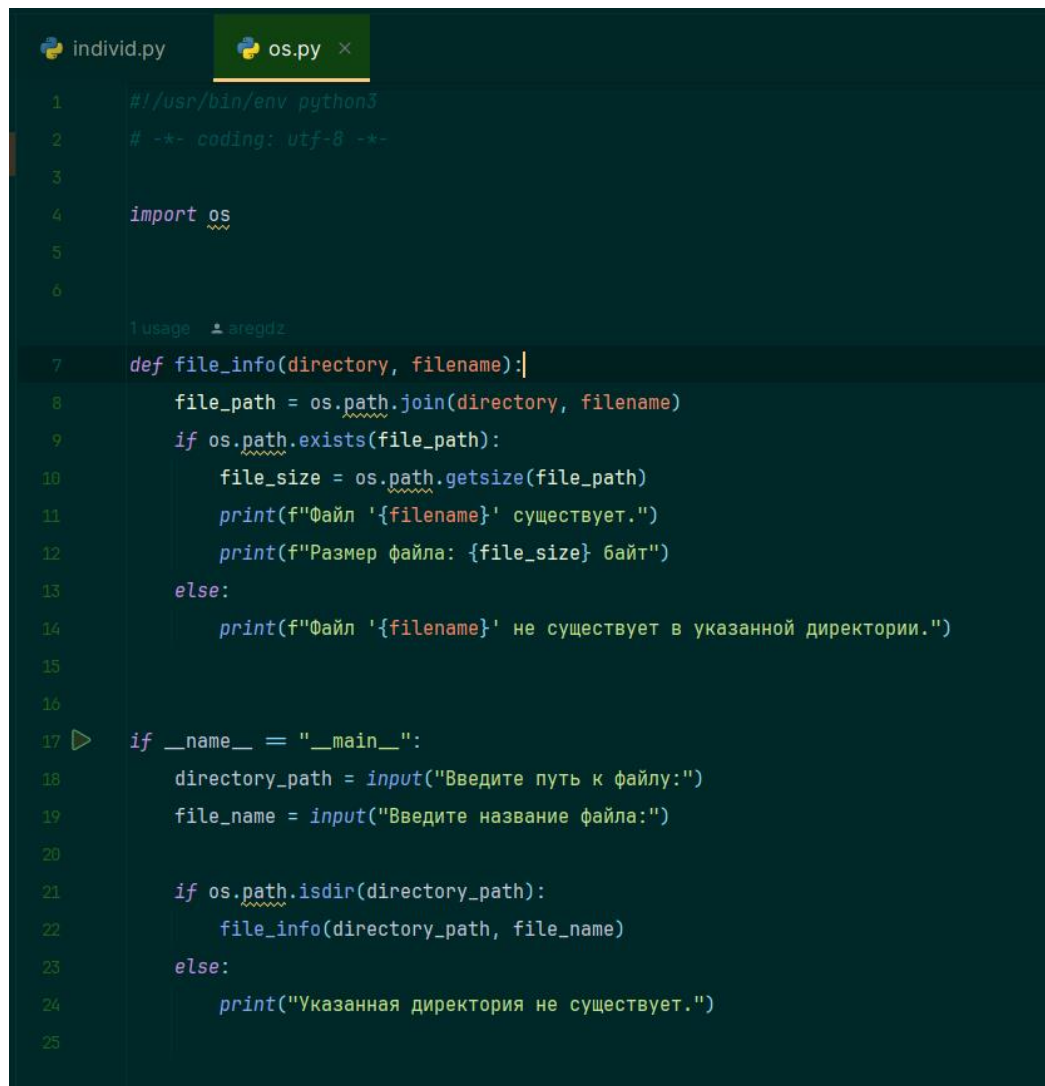
>>> r1
the In of midst misty a when morning, sun's the lazily rays through pierced branches the with covered thin a of layer the frost, woke city slowly up steadily. and emerged People their from
homes, cozy to ready the greet day new a with and smile The hope. filled streets the with of sounds talking footsteps, and voices, faint the of rustle underfoot. leaves such In it moments,
that seemed had time its slowed to pace everyone allow savor to simple the of existence present the moment.

>>> r2
['footsteps,', 'underfoot.'] 10
>>>

```

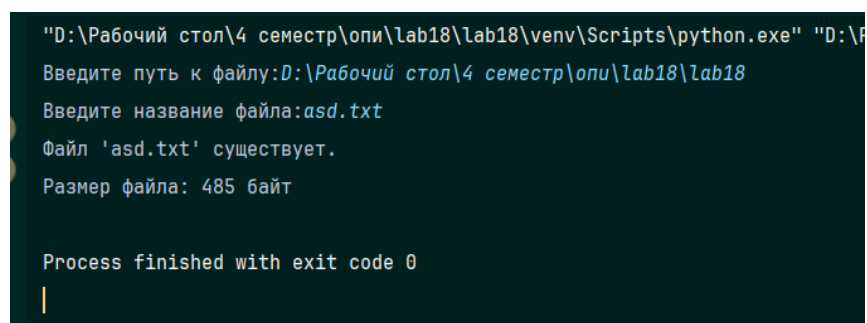
Рисунок 6 – пример выполнения задания

6. Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля `os`. Приведите решение этой задачи.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import os
5
6
7  def file_info(directory, filename):
8      file_path = os.path.join(directory, filename)
9      if os.path.exists(file_path):
10         file_size = os.path.getsize(file_path)
11         print(f"Файл '{filename}' существует.")
12         print(f"Размер файла: {file_size} байт")
13     else:
14         print(f"Файл '{filename}' не существует в указанной директории.")
15
16
17  if __name__ == "__main__":
18     directory_path = input("Введите путь к файлу:")
19     file_name = input("Введите название файла:")
20
21     if os.path.isdir(directory_path):
22         file_info(directory_path, file_name)
23     else:
24         print("Указанная директория не существует.")
25
```

Рисунок 7 – код для выполнения программы



```
"D:\Рабочий стол\4 семестр\опи\lab18\lab18\venv\Scripts\python.exe" "D:\Рабочий стол\4 семестр\опи\lab18\lab18\venv\Scripts\python.exe"
Введите путь к файлу:D:\Рабочий стол\4 семестр\опи\lab18\lab18
Введите название файла:asd.txt
Файл 'asd.txt' существует.
Размер файла: 485 байт

Process finished with exit code 0
```

Рисунок 8 – пример выполнения программы

7. Установил black, flake8, pre-commit. Использовал их во время коммита.

Контрольные вопросы:

1. Для открытия файла только для чтения в Python используется функция `open()` с режимом "r".

2. Для открытия файла только для записи в Python используется функция `open()` с режимом "w".

3. Для чтения данных из файла в Python используется метод `read()` объекта файла.

4. Для записи данных в файл в Python используется метод `write()` объекта файла.

5. Чтобы закрыть файл в Python и освободить ресурсы, используется метод `close()` объекта файла.

6. Конструкция `with ... as` в Python предназначена для создания контекстного менеджера, который автоматически управляет ресурсами. Это может быть полезно не только при работе с файлами, но и при работе с сетевыми соединениями, базами данных и другими ресурсами, требующими явного освобождения. Например, при работе с файлами она обеспечивает автоматическое закрытие файла после выхода из контекста, даже если произошло исключение.

7. Помимо рассмотренных методов чтения/записи данных из файла в Python существуют также методы:

- `readline()`: для чтения одной строки из файла;

- `readlines()`: для чтения всех строк из файла в список;

- `writelines(lines)`: для записи списка строк в файл.

8. Помимо рассмотренных функций модуля `os` для работы с файловой системой, существуют также функции:

- `os.rename(src, dst)`: для переименования файла или директории;

- `os.remove(path)`: для удаления файла;

`-os.mkdir(path)`: для создания директории;

`-os.rmdir(path)`: для удаления директории.