

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 №1**  
**дисциплины «анализ данных»**

Выполнил:  
Середа Кирилл Витальевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

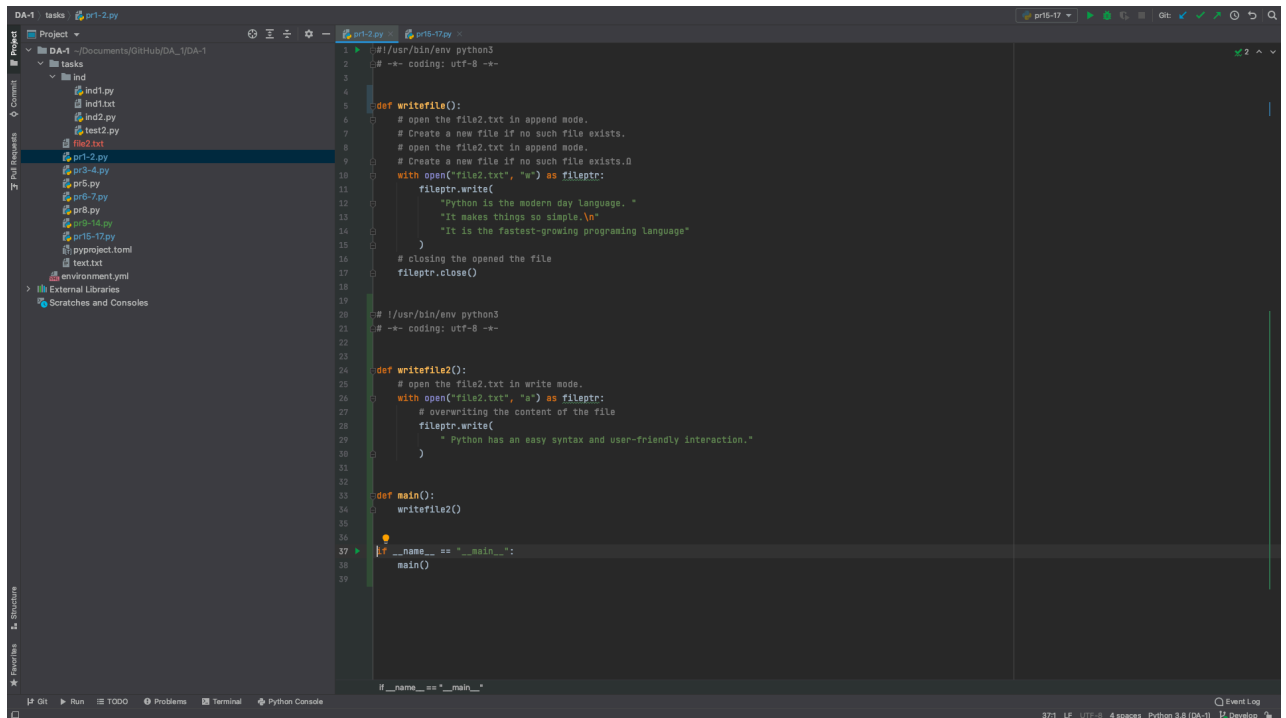
Ставрополь, 2024 г.

## Тема: Работа с файлами в языке Python

Цель: 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 выполнения заданий

#### 1) Проработал примеры 1 и 2 в файле pr1-2.py



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def writefile1():
6     # open the file2.txt in append mode.
7     # Create a new file if no such file exists.
8     # open the file2.txt in append mode.
9     # Create a new file if no such file exists.0
10    with open("file2.txt", "a") as fileptr:
11        fileptr.write(
12            "Python is the modern day language. "
13            "It makes things so simple.\n"
14            "It is the fastest-growing programming language"
15        )
16    # closing the opened file
17    fileptr.close()
18
19
20 # !/usr/bin/env python3
21 # -*- coding: utf-8 -*-
22
23
24 def writefile2():
25     # open the file2.txt in write mode.
26     with open("file2.txt", "w") as fileptr:
27         # overwriting the content of the file
28         fileptr.write(
29             "Python has an easy syntax and user-friendly interaction."
30         )
31
32
33 def main():
34     writefile2()
35
36
37 if __name__ == "__main__":
38     main()
39
40
41 if __name__ == "__main__":
```

Рисунок 1 – Файл с примерами 1 и 2

#### 2) Проработал примеры 3 и 4 в файле pr3-4.py

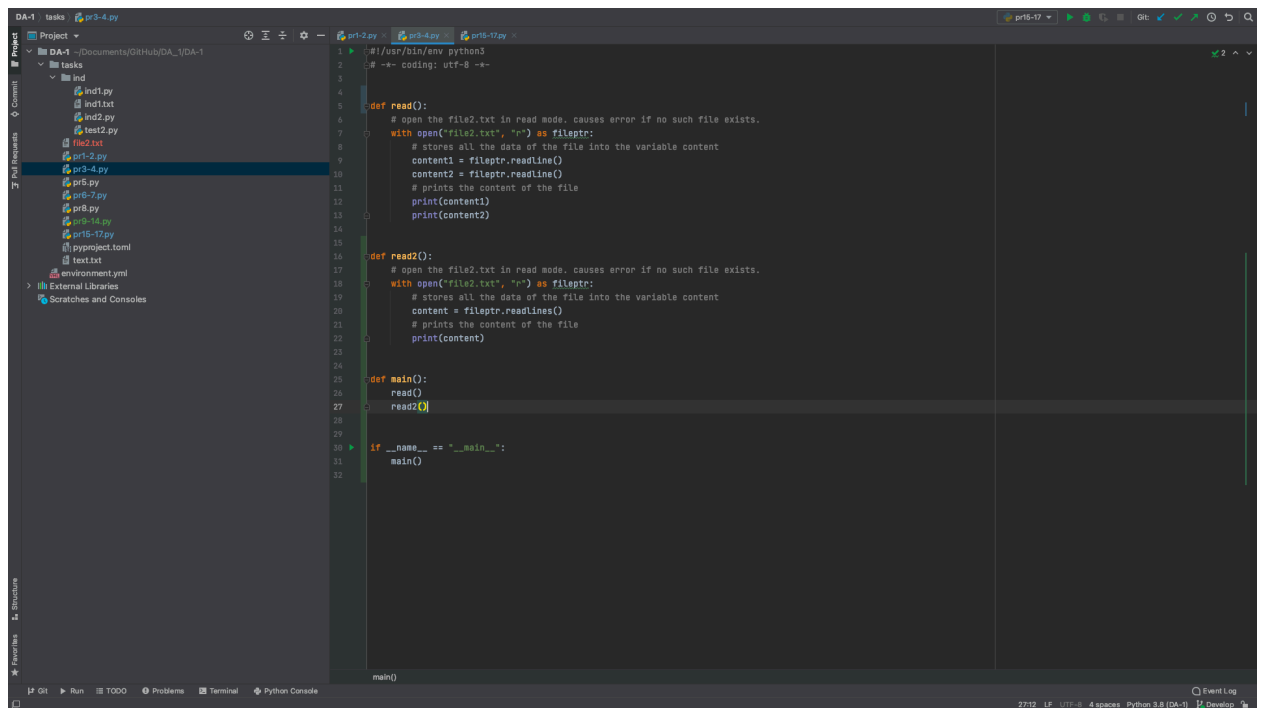


Рисунок 2 – Файл с примерами 3 и 4

3) Написал программу(5.py), в которой проработал примеры раздела методических указаний «Создание нового файла»

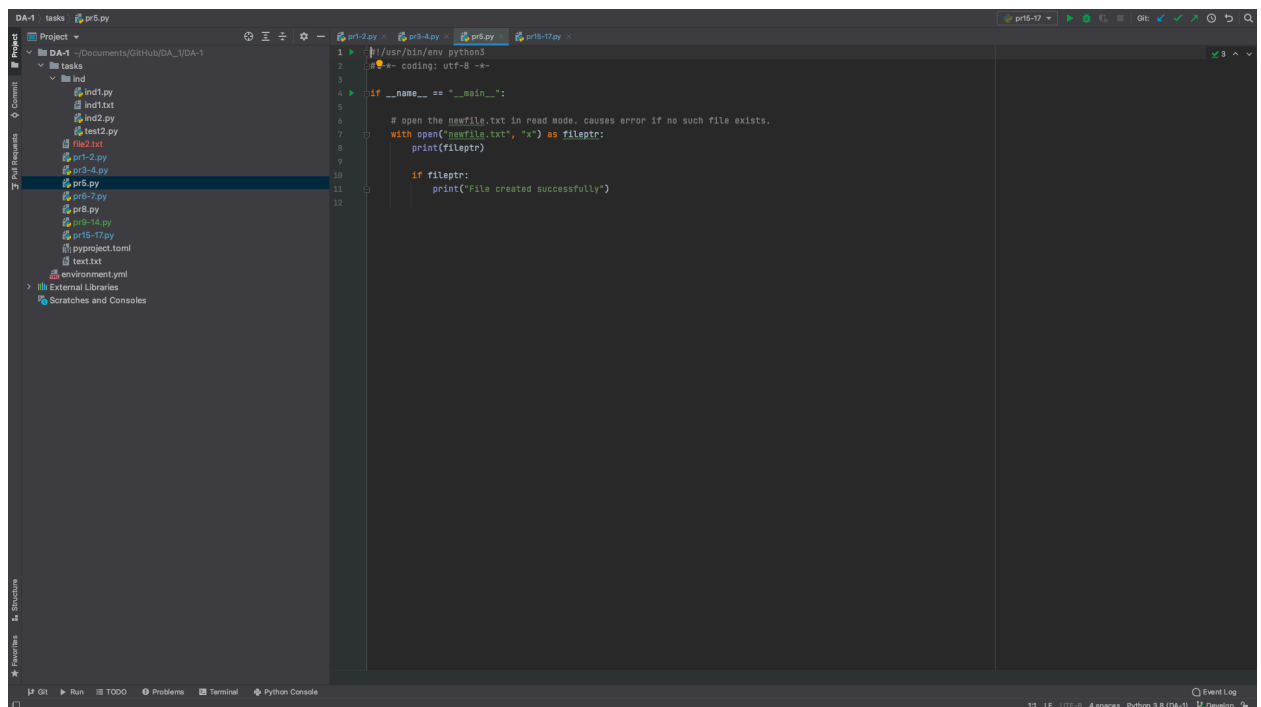


Рисунок 3 – Файл с примером 5

4) Проработал примеры из раздела «Изменение кодировки файла» в файле pr6-7.py

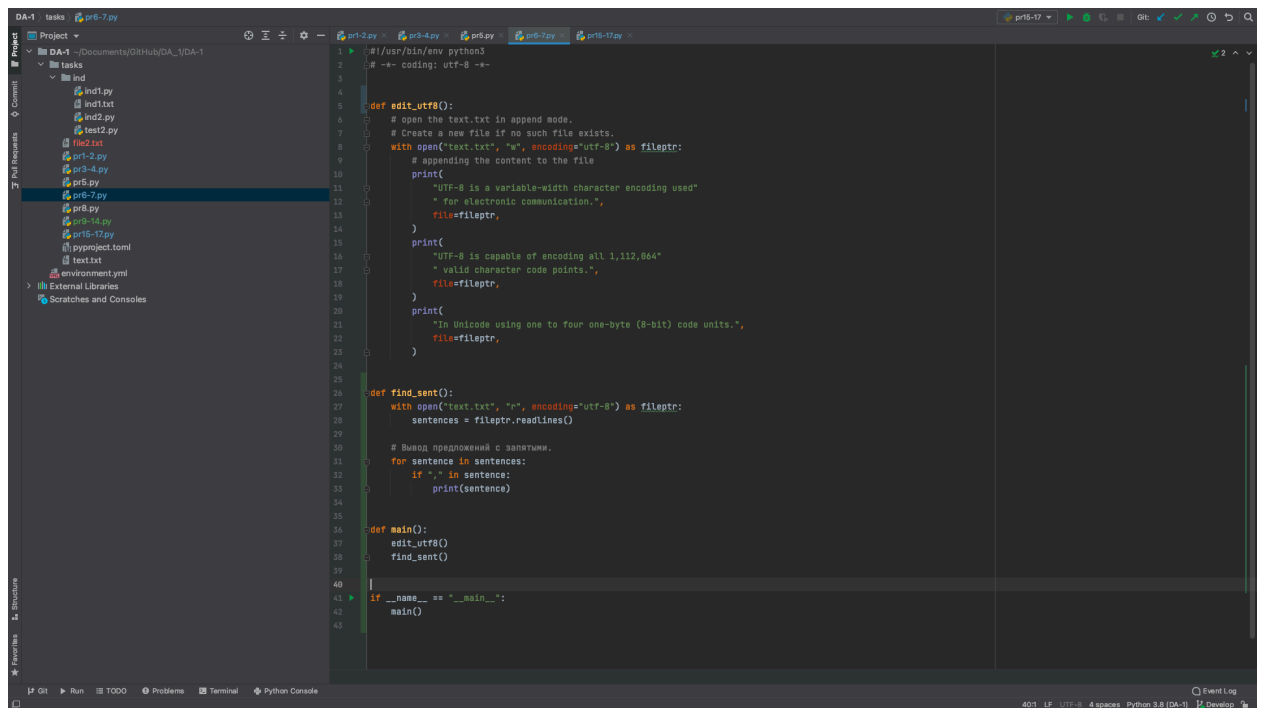


Рисунок 4 – Файл с примерами 6 и 7

## 5) Проработал примеры из раздела «Позиция указателя файла»

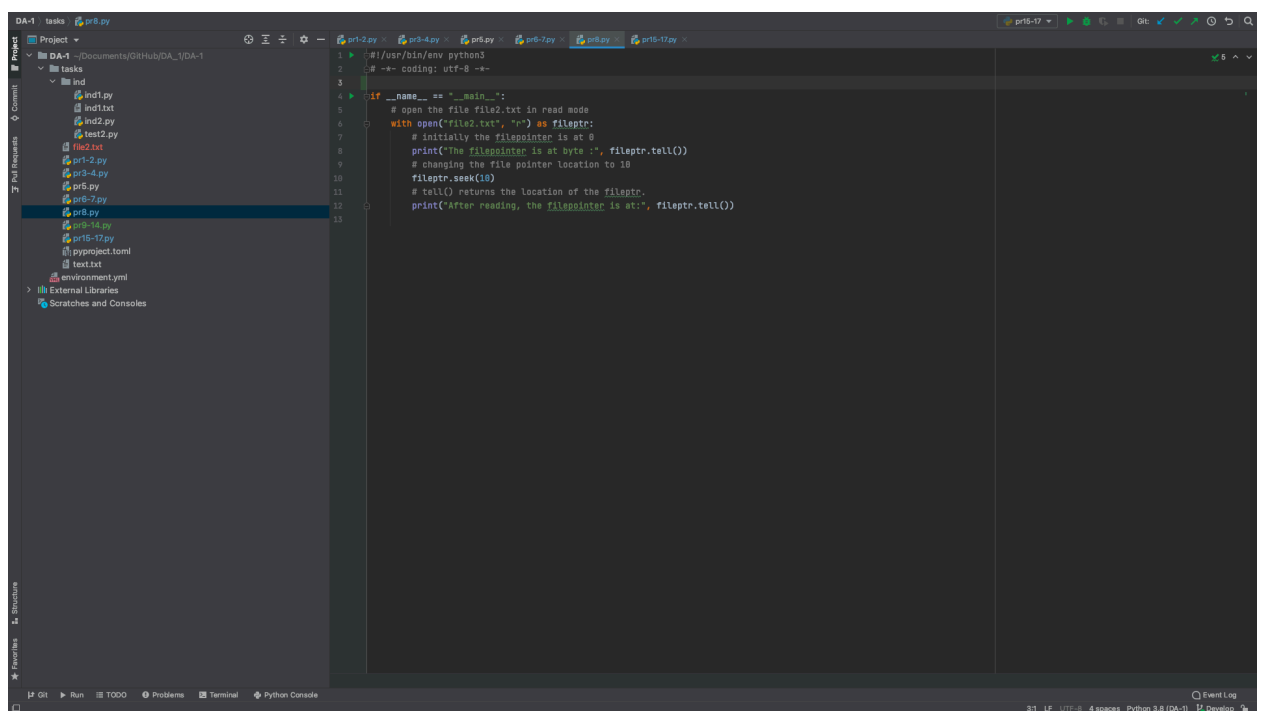


Рисунок 5 – Файл с примером 8

## 6) Проработал примеры из раздела «Модуль OS»

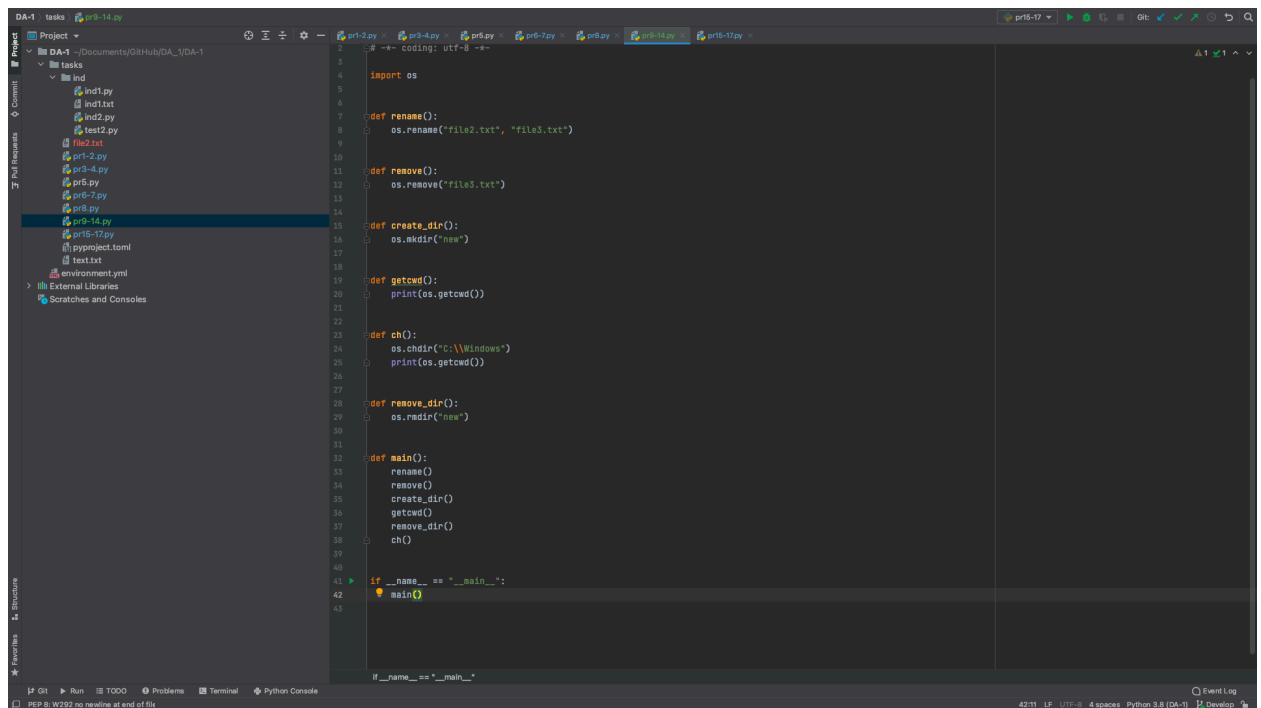


Рисунок 6 – Файл с примерами 9-14

7) Проработал примеры из раздела «Доступ к элементам командной строки в языке программирования Python»

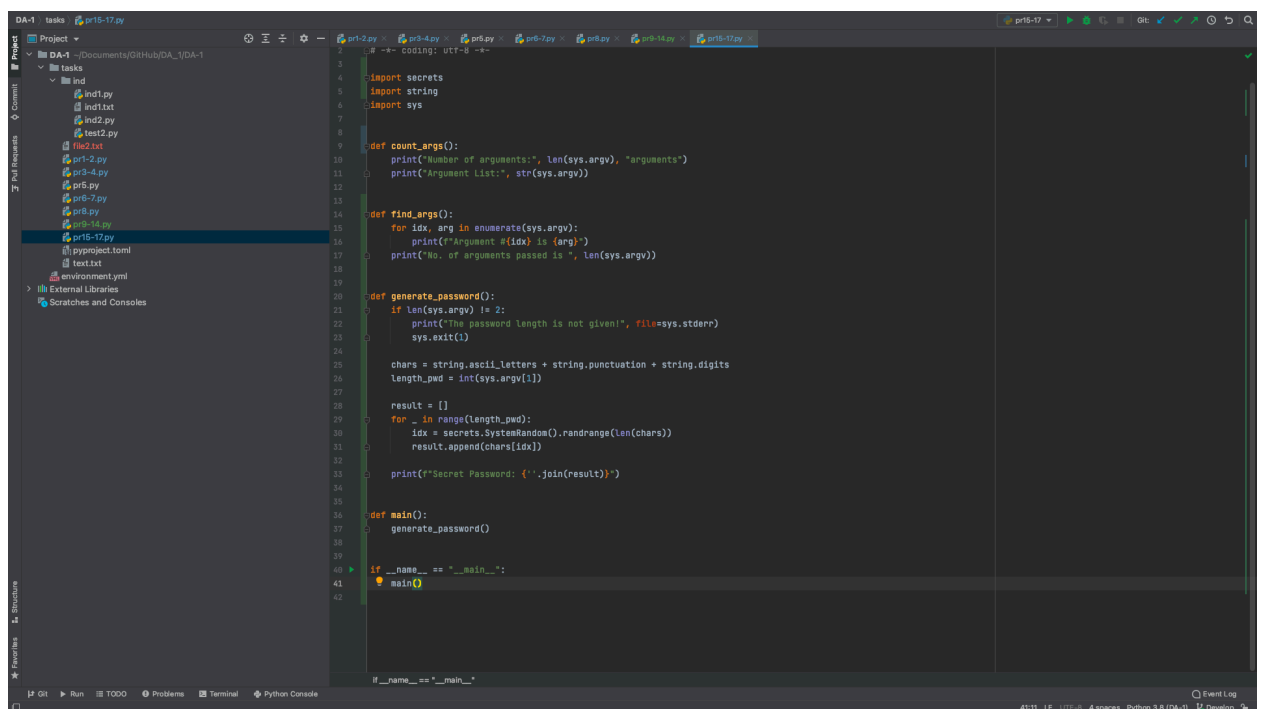


Рисунок 7 – Файл с примерами 15-17

8) Выполнил индивидуальное задание 1(Вариант 11) в котором написал программу, которая считывает текст из файла и выводит на экран

только те предложения, которые начинаются с тире и перед которым могут находиться только пробельные символы.

Считываемый файл:

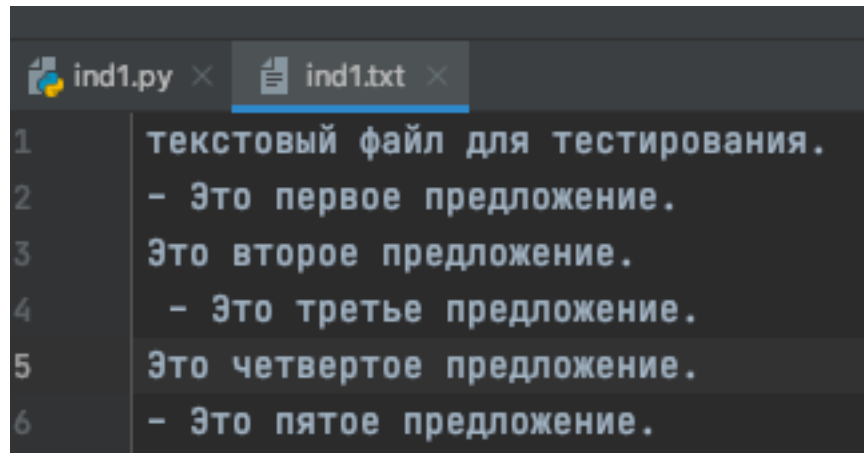


Рисунок 8 – Файл для теста программы ind1.py

Код программы:

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

import re

def main():
    filename = "ind1.txt"

    # Чтение текста из файла
    with open(filename, "r", encoding="utf-8") as file:
        text = file.read()

    # Разделение текста на предложения
    sentences = re.split(r"(?<=[.!?])\s+", text)

    # Вывод предложений, начинающихся с тире и пробельных символов перед ним
    for sentence in sentences:
        if re.match(r"\s*-", sentence):
            print(sentence)

if __name__ == "__main__":
    main()
```

```
/Users/MelancholySeal/opt/anaconda3/envs/DA-1/bin/python /Users/MelancholySeal/Documents/GitHub/DA_1/DA-1/tasks/ind/ind1.py
- Это первое предложение.
- Это третье предложение.
- Это пятое предложение.

Process finished with exit code 0
```

Рисунок 9 – Результат выполнения программы

9) Выполнил второе индивидуальное задание(Вариант 15), в котором написал программу, которая ищет функции к которым не оставили комментарии.

Код программы:

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

import sys

def find_undocumented_functions(file_path):
    try:
        with open(file_path, 'r', encoding='utf-8') as file:
            lines = file.readlines()
    except FileNotFoundError:
        print(f"Файл '{file_path}' не найден.")
        return
    except IOError:
        print(f"Ошибка при открытии файла '{file_path}'.")
        return

    undocumented_functions = []

    in_function = False
    for line_num, line in enumerate(lines, 1):
        line = line.strip()
        if line.startswith('def '):
            function_name = line.split('(')[0].split('def ')[1]
            in_function = True
            if not (line_num > 1 and lines[line_num - 2].strip().startswith('#')):
                undocumented_functions.append((function_name, file_path, line_num))
        elif line.startswith('class '):
            in_function = False
        elif line.startswith('#') and in_function:
            in_function = False

    return undocumented_functions

def main():
    if len(sys.argv) < 2:
        print("Использование: python script.py file1.py file2.py ...")
        return

    for file_path in sys.argv[1:]:
        undocumented_functions = find_undocumented_functions(file_path)
        if undocumented_functions:
            print(f"Недокументированные функции в файле '{file_path}':")
            for function in undocumented_functions:
                print(f"    - Функция '{function[0]}' начинается на строке {function[2]}")
        else:
            print(f"В файле '{file_path}' нет недокументированных функций.")

if __name__ == "__main__":
    main()
```

```
(BA-1) MelancholySeal@Kiras-MacBook-Air DA-1 % python /Users/MelancholySeal/Documents/GitHub/DA_1/DA-1/tasks/ind/ind2.py /Users/MelancholySeal/Documents/GitHub/DA_1/DA-1/tasks/ind/test2.py /Users/MelancholySeal/Documents/GitHub/DA_1/DA-1/tasks/ind/test2.1.py
Недокументированные функции в файле '/Users/MelancholySeal/Documents/GitHub/DA_1/DA-1/tasks/ind/test2.py':
- Функция 'doc_function' начинается на строке 9
- Функция 'another_documented_function' начинается на строке 16
- Функция 'another_undocumented_function' начинается на строке 21
- Функция 'main' начинается на строке 25
Недокументированные функции в файле '/Users/MelancholySeal/Documents/GitHub/DA_1/DA-1/tasks/ind/test2.1.py':
- Функция 'documented_function' начинается на строке 2
- Функция 'undocumented_function' начинается на строке 9
(BA-1) MelancholySeal@Kiras-MacBook-Air DA-1 %
```

Рисунок 10 – Результат выполнения программы ind2.py

### Ответы на вопросы

1. Для открытия файла только для чтения в Python используется функция **open()** с режимом **'r'**:

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

2. Для открытия файла только для записи в Python используется функция **open()** с режимом **'w'**:

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

3. Для чтения данных из файла в Python можно использовать методы **read()**, **readline()** или **readlines()**. Например:

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

4. Для записи данных в файл в Python используется метод **write()**.  
Например:

```
file.write("This is some data to write to the file.")
```

5. Для закрытия файла в Python используется метод **close()**.  
Например:

```
file.close()
```

6. Конструкция **with ... as** используется в Python для автоматического управления контекстом выполнения. Это гарантирует, что ресурсы будут правильно освобождены при выходе из блока **with**. Она может использоваться не только для работы с файлами, но и для других ресурсов, требующих управления контекстом, например, сокетов, баз данных и т. д.

7. Помимо уже рассмотренных методов, для работы с файлами в Python существует ряд других методов, таких как **seek()**, **tell()**, **truncate()**, **flush()** и др., которые позволяют перемещаться по файлу, определять текущую позицию, обрезать файл и принудительно записывать буфер в файл соответственно.



8. Помимо рассмотренных функций, модуль **os** в Python содержит множество других функций для работы с файловой системой. Некоторые из них включают **os.rename()**, **os.remove()**, **os.mkdir()**, **os.rmdir()** и т. д. Эти функции предоставляют функциональность для переименования файлов, удаления файлов и директорий, создания директорий и других операций на уровне файловой системы

Вывод: приобрел навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучил основные методы модуля **os** для работы с файловой системой, получение аргументов командной строки.