

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №3
по дисциплине основы программной инженерии

Выполнил: Грובה
Софья Кирилловна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил: Доцент кафедры
инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2022 г

```

fileptr = open('file2.txt', 'w')

fileptr.write(
    'Python is the modern day language. It makes so
    simple.\n' 'it is the fastest-growing programing
    language'
)

fileptr.close()

with open('file2.txt', 'w') as
fileptr: fileptr.write(
    'Python is the modern day language. It makes
    simple.\n' 'it is the fastest-growing programing
    language'
)

```

Рисунок 3.1 – Пример 1

```

with open('file2.txt', 'a') as fileptr:
    fileptr.write('python has an easy syntax and user-friendly

```

Рисунок 3.2 – Пример 2

```

with open('file2.txt', 'r') as
fileptr: content1 =
fileptr.readline() content2 =
fileptr.readline() print(content1)
print(content2)

```

Рисунок 3.3 – Пример 3

```

with open('file2.txt', 'r') as
fileptr: content =
fileptr.readlines() print(content)

```

Рисунок 3.4 – Пример 4

```

with open('newfile.txt', 'x') as
fileptr: print(fileptr)
if fileptr:
    print('File created successfully')

```

Рисунок 3.5 – Пример 5

```

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

if __name__ == '__main__':
    with open('text.txt', 'w', encoding='utf-8') as
        fileptr: print(
            'UTF-8 is a variable-width character encoding used for electronic
            communication.',
            file=fileptr
        )
        print(
            'UTF-8 is capable of encoding all 1,112,064 valid
            character code
            points.',
            file=fileptr
        )

```

```
print(
    'In Unicode using one to four one-byte (8-
    bit) code units.', file=fileptr
)
```

Рисунок 3.6 – Пример 6

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

if __name__ == '__main__':
    with open('text.txt', 'r', encoding='utf-
    8') as f: text = f.read()
        save
    = False
    frase = ''
    result = []

        for
    l in text:
        if l == '"':
            if save:
                r
                result.append(frase)
                frase = ''
            s
        ave = not save
        continue
```

Рисунок 3.7 – Индивидуальное задание 1

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

if __name__ == '__main__':
    new_text = ''
    with open('fileForProgram.py',
    'r') as f: save = True
        for line in f:
            f
            or s in
            line: if s
            == '#':
                save = not save
                elif s ==
                '\n' and save == False: save
                = not save
            if save:
                new t
```

Рисунок 3.8 – Индивидуальное задание 2

Контрольные вопросы:

1. Как открыть файл в языке Python только для чтения?

Использовать в open параметр 'r'

2. Как открыть файл в языке Python только для записи?

Использовать в open параметр 'w'

3. Как прочитать данные из файла в языке Python?

Open('FileName','type of open')

4. Как записать данные в файл в языке Python?

File.write()

5. Как закрыть файл в языке Python?

File.close()

6. Изучите самостоятельно работу конструкции with ... as. Каково ее назначение в языке

Python? Где она может быть использована еще, помимо работы с файлами?

Конструкция with ... as используется для оборачивания выполнения блока инструкций менеджером контекста. Иногда это более удобная конструкция, чем [try...except...finally](#).

Для чего применяется конструкция with ... as? Для гарантии того, что критические функции выполнятся в любом случае. Самый распространённый пример использования этой конструкции - открытие файлов. Я уже рассказывал об [открытии файлов с помощью функции open](#), однако конструкция with ... as, как правило, является более удобной и гарантирует закрытие файла в любом случае.

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

Режим	Обозначение
'r'	открытие на чтение (является значением по умолчанию).
'w'	открытие на запись, содержимое файла удаляется, если файла не существует, создается новый.
'x'	открытие на запись, если файла не существует, иначе исключение.
'a'	открытие на дозапись, информация добавляется в конец файла.
'b'	открытие в двоичном режиме.
't'	открытие в текстовом режиме (является значением по умолчанию).
'+'	открытие на чтение и запись

8. Какие существуют, помимо рассмотренных, функции модуля os для работы с файловой системой?

Так же есть модуль pathlib