

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 №2.15
дисциплины «Анализ данных»

Выполнил:
Магдаев Даламбек Магомедович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

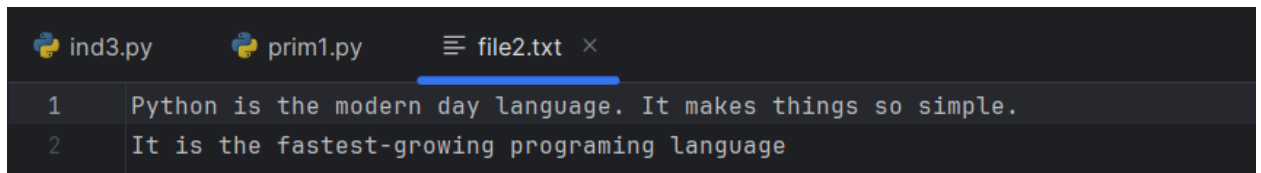
Ставрополь, 2024 г.

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.

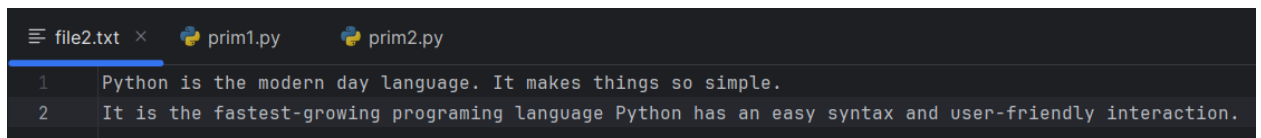
Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку `developer` и перешел на нее.
2. Проработал примеры лабораторной работы:



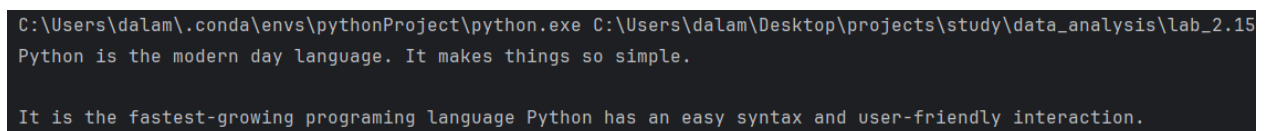
```
ind3.py  prim1.py  file2.txt x
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programing language
```

Рисунок 1. Результат работы примера 1



```
file2.txt x  prim1.py  prim2.py
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programing language Python has an easy syntax and user-friendly interaction.
```

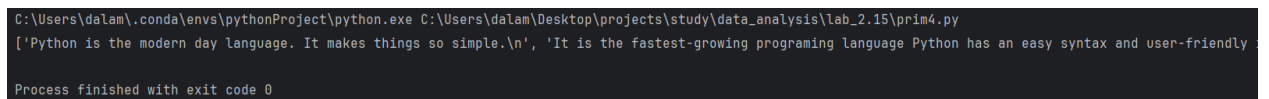
Рисунок 2. Результат работы примера 2



```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15
Python is the modern day language. It makes things so simple.


It is the fastest-growing programing language Python has an easy syntax and user-friendly interaction.
```

Рисунок 3. Результат работы примера 3



```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15\prim4.py
['Python is the modern day language. It makes things so simple.\n', 'It is the fastest-growing programing language Python has an easy syntax and user-friendly
Process finished with exit code 0
```

Рисунок 4. Результат работы примера 4



```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\Desktop\
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully
```

Рисунок 5. Результат работы примера 5

```
UTF-8 is a variable-width character encoding used for electronic communication.  
UTF-8 is capable of encoding all 1,112,064 valid character code points.  
In Unicode using one to four one-byte (8-bit) code units.
```

Рисунок 6. Результат работы примера 6

```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\Desktop\  
UTF-8 is capable of encoding all 1,112,064 valid character code points.  
  
Process finished with exit code 0
```

Рисунок 7. Результат работы примера 7

```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\Desktop\  
The filepointer is at byte : 0  
After reading, the filepointer is at: 10  
  
Process finished with exit code 0
```

Рисунок 8. Результат работы примера 8

≡ file3.txt

Рисунок 9. Результат работы примера 9

≡ newfile.txt

Рисунок 10. Результат работы примера 10

📁 new

Рисунок 11. Результат работы примера 11

```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\  
C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15  
  
Process finished with exit code 0
```

Рисунок 12. Результат работы примера 12

```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\  
Текущий рабочий каталог - C:\Windows
```

Рисунок 13. Результат работы примера 13

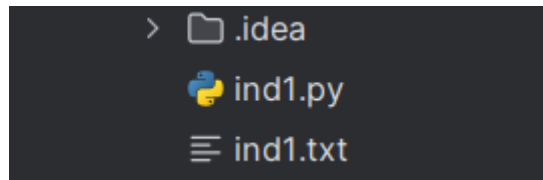


Рисунок 14. Результат работы примера 14

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15> python prim15.py test.py arg1 arg2 arg3
Number of arguments: 5 arguments
Argument List: ['prim15.py', 'test.py', 'arg1', 'arg2', 'arg3']
```

Рисунок 15. Результат работы примера 15

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15> python prim16.py argv_count.py Knowledge Hut 21
Argument #0 is prim16.py
Argument #1 is argv_count.py
Argument #2 is Knowledge
Argument #3 is Hut
Argument #4 is 21
No. of arguments passed is 5
```

Рисунок 16. Результат работы примера 16

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15> python prim17.py 12
Secret Password: @\>YwowZn/_M
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15>
```

Рисунок 17. Результат работы примера 17

3. Выполнил индивидуальное задание №1: Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

1	Laboratory work number 2.15.
2	Working with files in Python.
3	Magdaev Dalambek.

Рисунок 18. Содержимое файла ind1.txt

```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam
Magdaev Dalambek.

Working with files in Python.

Laboratory work number 2.15.
```

Рисунок 19. Результат работы индивидуального задания №1

4. Выполнил индивидуальное 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.

```
1 Задача организации, в особенности же реализация намеченного плана развития позволяет выполнить важнейшие задания по
2 разработке дальнейших направлений развития системы массового участия. Таким образом, выбранный нами инновационный путь
3 требует от нас системного анализа экономической целесообразности принимаемых решений! Соображения высшего порядка, а
4 также сложившаяся структура организации напрямую зависит от дальнейших направлений развития проекта?
```

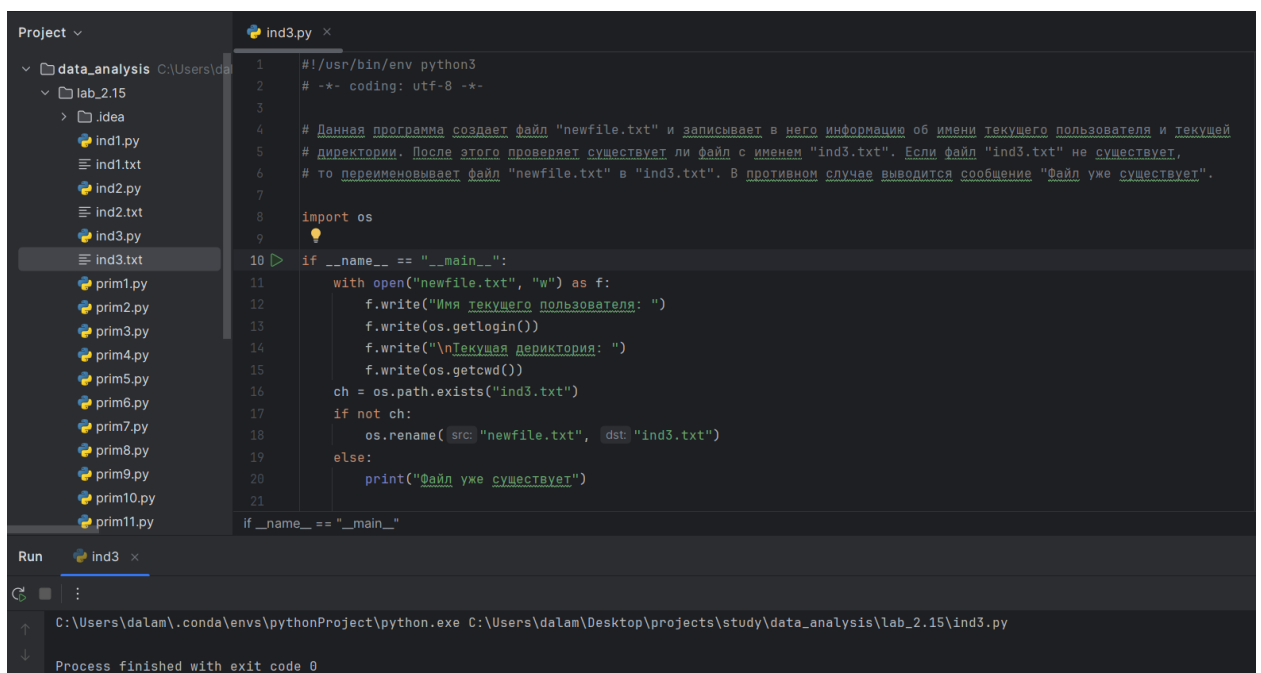
Рисунок 18. Содержимое файла ind1.txt

```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam
Длина самого длинного слова: 16
Самые длинные слова: ['целесообразности']

Process finished with exit code 0
```

Рисунок 19. Результат работы индивидуального задания №2

5. Выполнил индивидуальное задание №3: Данная программа создает файл "newfile.txt" и записывает в него информацию об имени текущего пользователя и текущей директории. После этого проверяет существует ли файл с именем "ind3.txt". Если файл "ind3.txt" не существует, то переименовывает файл "newfile.txt" в "ind3.txt". В противном случае выводится сообщение "Файл уже существует".



```
Project
└─ data_analysis C:\Users\dalam\...
   └─ lab_2.15
      └─ idea
         ├── ind1.py
         ├── ind1.txt
         ├── ind2.py
         ├── ind2.txt
         ├── ind3.py
         └── ind3.txt
            ├── prim1.py
            ├── prim2.py
            ├── prim3.py
            ├── prim4.py
            ├── prim5.py
            ├── prim6.py
            ├── prim7.py
            ├── prim8.py
            ├── prim9.py
            ├── prim10.py
            └── prim11.py

ind3.py
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 # Данная программа создает файл "newfile.txt" и записывает в него информацию об имени текущего пользователя и текущей
5 # директории. После этого проверяет существует ли файл с именем "ind3.txt". Если файл "ind3.txt" не существует,
6 # то переименовывает файл "newfile.txt" в "ind3.txt". В противном случае выводится сообщение "Файл уже существует".
7
8 import os
9
10 if __name__ == "__main__":
11     with open("newfile.txt", "w") as f:
12         f.write("Имя текущего пользователя: ")
13         f.write(os.getlogin())
14         f.write("\nТекущая директория: ")
15         f.write(os.getcwd())
16     ch = os.path.exists("ind3.txt")
17     if not ch:
18         os.rename(src="newfile.txt", dst="ind3.txt")
19     else:
20         print("Файл уже существует")
21
if __name__ == "__main__"
```

Run ind3

```
C:\Users\dalam\.conda\envs\pythonProject\python.exe C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.15\ind3.py
Process finished with exit code 0
```

Рисунок 20. Код и результат работы индивидуального задания №3

Ответы на контрольные вопросы:

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

Ответ: Чтобы открыть файл для чтения, мы используем режим `r` или `rb`.
`file = open("text.txt", 'r').`

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

Ответ: Чтобы открыть файл для записи, мы используем режим `w` или `wb`. `file = open("text.txt", 'w')`. Также можно использовать режим `a` или `ab`, чтобы не перезаписывать файл, а дозаписывать в него.

6. Как прочитать данные из файла в языке Python?

Ответ: Чтобы прочитать файл с помощью сценария Python, Python предоставляет метод `read()`. Метод `read()` считывает строку из файла. Он может читать данные как в текстовом, так и в двоичном формате. Также можно использовать цикл `for`: `for i in fileptr: print(i)`; также можно использовать метод `readline()`, который читает строки файла с самого начала, т. е. если мы используем его два раза, мы можем получить первые две строки файла. Python также предоставляет метод `readlines()`, который используется для чтения строк. Возвращает список строк до конца файла (EOF).

7. Как записать данные в файл в языке Python?

Ответ: Запись данных в файл. Записать данные в файл можно с помощью метода `write()`.

8. Как закрыть файл в языке Python?

Ответ: После того, как мы открыли файл, и выполнили все нужные операции, нам необходимо его закрыть. Для закрытия файла используется функция `close()`.

9. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке?

Ответ: Конструкция `with ... as` в языке Python предназначена для обеспечения управления ресурсами с автоматическим освобождением этих ресурсов после завершения блока кода. Она часто используется с объектами, которые поддерживают протокол менеджеров контекста.

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

Ответ: Метод `writelines()`: Данный метод принимает список строк в качестве аргумента и записывает каждую строку списка в файл. В Python также существует метод `readinto()`, который читает данные из файла и записывает их в предварительно выделенный буфер. Это может быть полезно, если нужно работать с данными напрямую в буфере, минуя дополнительные копирования в памяти.

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

Ответ: `os.chmod (path, mode, *, dir_fd=None, follow_symlinks=True)` - смена прав доступа к объекту (`mode` - восьмеричное число). `os.chown (path, uid, gid, *, dir_fd=None, follow_symlinks=True)` - меняет `id` владельца и группы (Unix). `os.link (src, dst, *, src_dir_fd=None, dst_dir_fd=None, follow_symlinks=True)` - создаёт жёсткую ссылку. `os.listdir (path=".")` - список файлов и директорий в папке. `os.makedirs (path, mode=0o777, exist_ok=False)` - создаёт директорию, создавая при этом промежуточные директории.

Вывод: в результате выполнения программы были приобретены навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучены основные

методы модуля os для работы с файловой системой, изучено получение аргументов командной строки.