

МИНИСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.15**  
Дисциплина «Анализ данных»

Выполнил:  
Бабенко Артём Тимофеевич  
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)

Проверил:  
Воронкин Р.А., доцент кафедры  
инфокоммуникаций

---

(подпись)

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

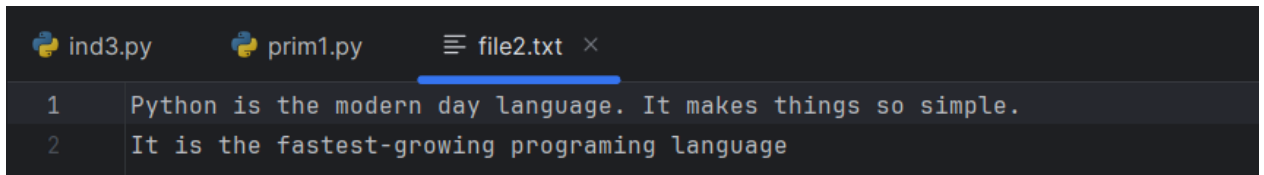
г. Ставрополь, 2024 г.

## 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.

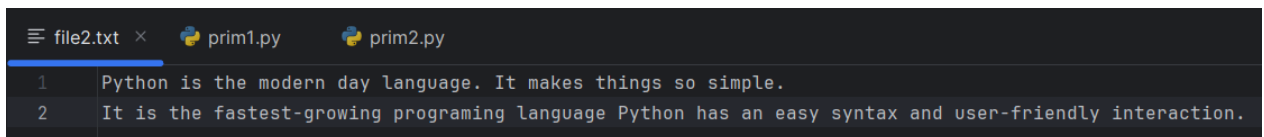
### Порядок выполнения работы:

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



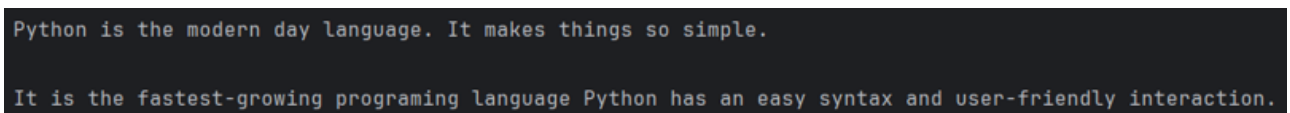
```
ind3.py  prim1.py  file2.txt x
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programming language
```

Рисунок 1. Результат работы примера 1



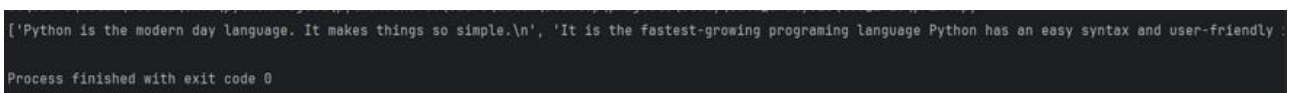
```
file2.txt x  prim1.py  prim2.py
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.
```

Рисунок 2. Результат работы примера 2



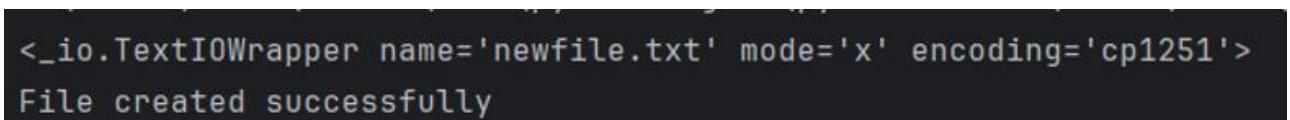
```
Python is the modern day language. It makes things so simple.
It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.
```

Рисунок 3. Результат работы примера 3



```
['Python is the modern day language. It makes things so simple.\n', 'It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.\n']
Process finished with exit code 0
```

Рисунок 4. Результат работы примера 4



```
<_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

```
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

```
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

```
Process finished with exit code 0
```

Рисунок 12. Результат работы примера 12

```
Текущий рабочий каталог - C:\Windows
```

Рисунок 13. Результат работы примера 13

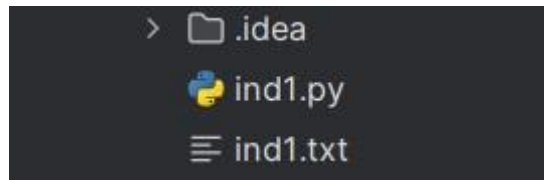


Рисунок 14. Результат работы примера 14

```
Number of arguments: 5 arguments  
Argument List: ['prim15.py', 'test.py', 'arg1', 'arg2', 'arg3']
```

Рисунок 15. Результат работы примера 15

```
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

```
Secret Password: @\>YwowZn/_M
```

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.

```
Laboratory work number 2.15  
Working with files in Python  
Babenko Artem
```

Рисунок 18. Содержимое файла ind1.txt

```
C:\Users\baben\.conda\envs\lab_2.15\Scripts\python.exe D:\Gitlab\AD\2.15\prog\ind\ind1.py  
Babenko Artem  
Working with files in Python  
Laboratory work number 2.15
```

Рисунок 19. Результат работы индивидуального задания №1

### 4. Выполнил индивидуальное задание №2:

В операционных системах на базе Unix обычно присутствует утилита с названием head.

Она выводит первые десять строк содержимого файла, имя которого передается в качестве

аргумента командной строки. Напишите программу на Python, имитирующую поведение

этой утилиты. Если файла, указанного пользователем, не существует, или не задан

аргумент командной строки, необходимо вывести соответствующее

```
сообщение об ошибке.  
print(10.532) # => 10.234  
print(10.621) # => 10.621  
print(10.292) # => 10.292  
print(10.133) # => 10.133  
print(10.155) # => 10.155  
print(10.555) # => 10.555  
print(10.333) # => 10.333  
print(10.123) # => 10.123  
print(10.321) # => 10.321
```

Рисунок 18. Содержимое файла ind2.txt

```
C:\Users\baben\.conda\envs\2.15\Scripts\python.exe D:\Gitlab\AD\2.15\prog\ind\ind2.txt  
10.532  
10.621  
10.292  
10.133  
10.155  
10.555  
10.333
```

Рисунок 19. Результат работы индивидуального задания №2

### **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.

<https://github.com/lorianss/2.15>