

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ

по лабораторной работе №2.15

Дисциплина: «Анализ данных»
Тема: «Работа с файлами в языке Python»

Выполнил:
Епифанов Алексей Александрович
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 Роман Александрович

(подпись)

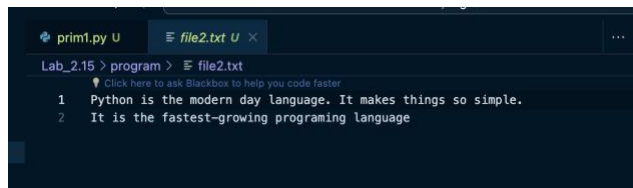
Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

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.

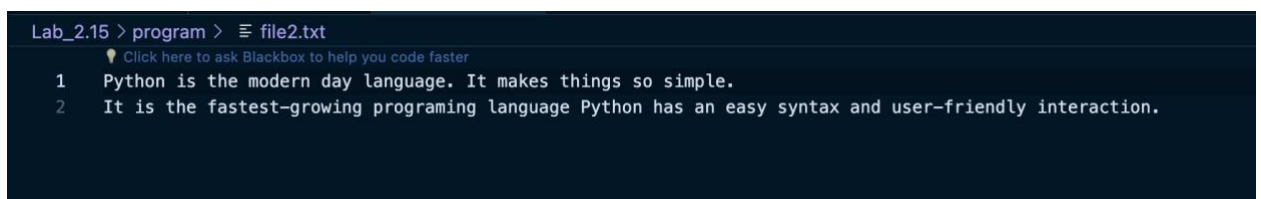
Порядок выполнения работы:

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



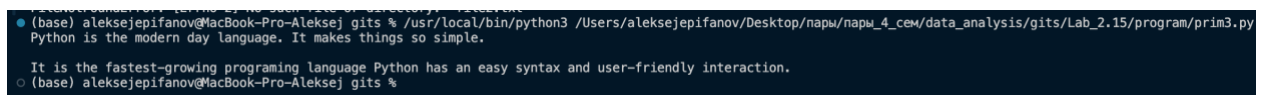
```
prim1.py U file2.txt U
Lab_2.15 > program > file2.txt
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programming language
```

Рисунок 1. Результат работы примера 1



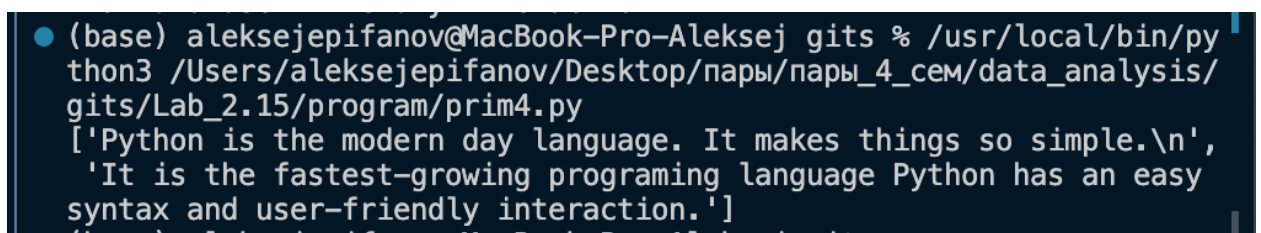
```
Lab_2.15 > program > file2.txt
1 Python is the modern day language. It makes things so simple.
2 It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.
```

Рисунок 2. Результат работы примера 2



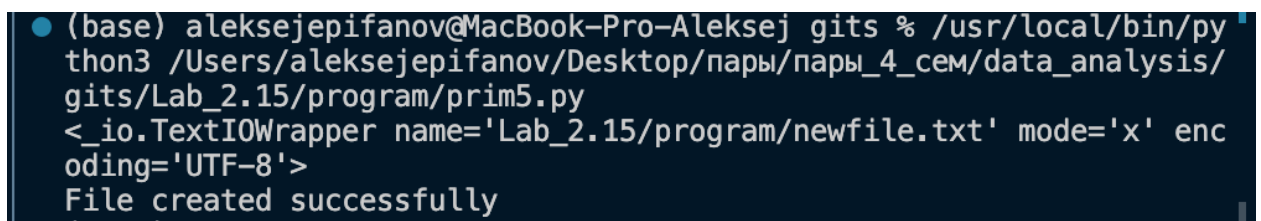
```
(base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/prim3.py
Python is the modern day language. It makes things so simple.
It is the fastest-growing programming language Python has an easy syntax and user-friendly interaction.
(base) aleksejepifanov@MacBook-Pro-Aleksej gits %
```

Рисунок 3. Результат работы примера 3



```
(base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/prim4.py
['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.']
```

Рисунок 4. Результат работы примера 4



```
(base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/prim5.py
<_io.TextIOWrapper name='Lab_2.15/program/newfile.txt' mode='x' encoding='UTF-8'>
File created successfully
```

Рисунок 5. Результат работы примера 5

```
Lab_2.15 > program > ≡ text.txt
Click here to ask Blackbox to help you code faster
1 UTF-8 is a variable-width character encoding used for electronic communication.
2 UTF-8 is capable of encoding all 1,112,064 valid character code points.
3 In Unicode using one to four one-byte (8-bit) code units.
4
```

Рисунок 6. Результат работы примера 6

```
• (base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/prim7.py
UTF-8 is capable of encoding all 1,112,064 valid character code points.
```

Рисунок 7. Результат работы примера 7

```
• (base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/prim8.py
The filepointer is at byte : 0
After reading, the filepointer is at: 10
○ (base) aleksejepifanov@MacBook-Pro-Aleksej gits %
```

Рисунок 8. Результат работы примера 8

```
program
≡ file3.txt
≡ newfile.txt
```

Рисунок 9. Результат работы примера 9

```
▼ program
≡ newfile.txt
```

Рисунок 10. Результат работы примера 10

```
▼ program
> new
≡ newfile.txt
```

Рисунок 11. Результат работы примера 11

```
• (base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/prim12.py
/Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits
```

Рисунок 12. Результат работы примера 12

```

• (base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары 4 сем/data analysis/gits/Lab_2.15/program/prim13.py /Users/aleksejepifanov

```

Рисунок 13. Результат работы примера 13

```

▼ program
  newfile.txt

```

Рисунок 14. Результат работы примера 14

```

• (base) aleksejepifanov@MacBook-Pro-Aleksej gits % python Lab_2.15/program/prim15.py arg1 arg2 arg3
Number of arguments: 4 arguments
Argument List: ['Lab_2.15/program/prim15.py', 'arg1', 'arg2', 'arg3']

```

Рисунок 15. Результат работы примера 15

```

• (base) aleksejepifanov@MacBook-Pro-Aleksej gits % python Lab_2.15/program/prim16.py Knowledge Hut 21
Argument #0 is Lab_2.15/program/prim16.py
Argument #1 is Knowledge
Argument #2 is Hut
Argument #3 is 21
No. of arguments passed is 4

```

Рисунок 16. Результат работы примера 16

```

• (base) aleksejepifanov@MacBook-Pro-Aleksej gits % python Lab_2.15/program/prim17.py 12
Secret Password: ?H{&oEJ>%xP0

```

Рисунок 17. Результат работы примера 17

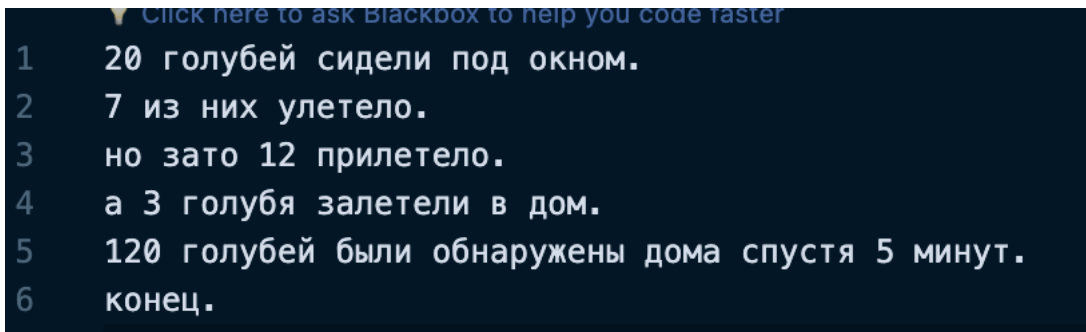
3. Выполнил индивидуальное задание номер 1 вариант 10: Написать программу, которая считывает текст из файла и выводит на экран только строки, не содержащие двузначных чисел.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import re
if __name__ == "__main__":
    with open("Lab_2.15/program/file_ind_1.txt", 'r') as file:
        text = file.readlines()
        result = []
        for line in text:
            if not re.search(r'\b\d{2}\b', line):

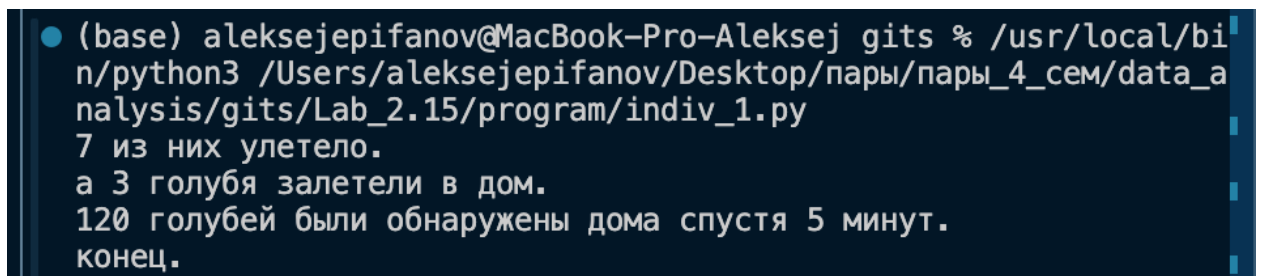
```

```
result.append(line)
print("".join(result))
```



```
1 20 голубей сидели под окном.
2 7 из них улетело.
3 но зато 12 прилетело.
4 а 3 голубя залетели в дом.
5 120 голубей были обнаружены дома спустя 5 минут.
6 конец.
```

Рисунок 18. Файл file_ind_1.txt



```
(base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/indiv_1.py
7 из них улетело.
а 3 голубя залетели в дом.
120 голубей были обнаружены дома спустя 5 минут.
конец.
```

Рисунок 19. Результат работы индивидуального задания 1

4. Выполнил индивидуальное задание номер 2 вариант 10: Ученикам, желающим запомнить правила написания слов в английском языке, часто напоминают следующее рифмованное одностишие: «I before E except after C» (I перед E, если не после C). Это правило позволяет запомнить, в какой последовательности писать буквы I и E, идущие в слове одна за другой, а именно: буква I должна предшествовать букве E, если непосредственно перед ними не стоит буква C. Если стоит – порядок гласных будет обратным. Примеры слов, на которые действует это правило: believe, chief, fierce, friend, ceiling и receipt. Но есть и исключения из этого правила, и одним из них является слово weird (странный). Напишите программу, которая будет построчно обрабатывать текстовый файл. В каждой строке может присутствовать много слов, а может и не быть ни одного. Слова, в которых буквы E и I не соседствуют друг с другом, обработке подвергать не следует. Если же такое соседство присутствует, необходимо проверить, соответствует ли написание анализируемого слова указанному выше правилу. Создайте и выведите на экран два списка. В первом должны располагаться слова,

следующие правилу, а во втором – нарушающие его. При этом списки не должны содержать повторяющиеся слова. Также отобразите на экране длину каждого списка, чтобы пользователю было понятно, сколько слов в файле не отвечает правилу.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import re
if __name__ == "__main__":
    words_follow_rule = set()
    words_break_rule = set()
    # Функция для проверки слова на соответствие правилу
    def check_word(word):
        if 'cie' in word:
            return False
        if 'cei' in word:
            return True
        if 'ie' in word:
            return True
        if 'ei' in word:
            return False
    with open('Lab_2.15/program/file_ind_2.txt', 'r') as file:
        for line in file:
            for word in re.findall(r'\b\w*(?:ei|ie)\w*\b', line, re.I):
                if word.lower() == 'weird': # Исключение
                    words_follow_rule.add(word.lower())
                elif word.lower() == 'wierd': # Неправильное исключение
                    words_break_rule.add(word.lower())
                elif check_word(word.lower()):
                    words_follow_rule.add(word.lower())
                else:
                    words_break_rule.add(word.lower())
    print("Слова, следующие правилу:")
    for word in words_follow_rule:
        print(word)
    print("Количество слов:", len(words_follow_rule))
    print("\nСлова, не следующие правилу:")
    for word in words_break_rule:
        print(word)
    print("Количество слов:", len(words_break_rule))
```

```
Lab_2.15 > program > ≡ file_ind_2.txt
Click here to ask Blackbox to help you code faster
1 I have a friend who has a weird belief about science.
2 The ceiling is a lie, there is no pie in the sky!
3
4 Their neighbor Keith received a beige sleigh from feisty reindeer.
5 |
```

Рисунок 20. Исходный файл file_ind_2.txt

```
● (base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/indiv_2.py
Слова, следующие правилу:
lie
pie
weird
friend
belief
ceiling
received
Количество слов: 7

Слова, не следующие правилу:
science
keith
their
sleigh
feisty
reindeer
beige
neighbor
Количество слов: 8
(base) aleksejepifanov@MacBook-Pro-Aleksej gits %
```

Рисунок 21. Результат работы программы индивидуального задания 2

5. Самостоятельно придумал задачу для работы с изученными функциями модуля `os`: необходимо создать директорию, перейти в нее, создать файл и записать в него текст, а затем переименовать его.

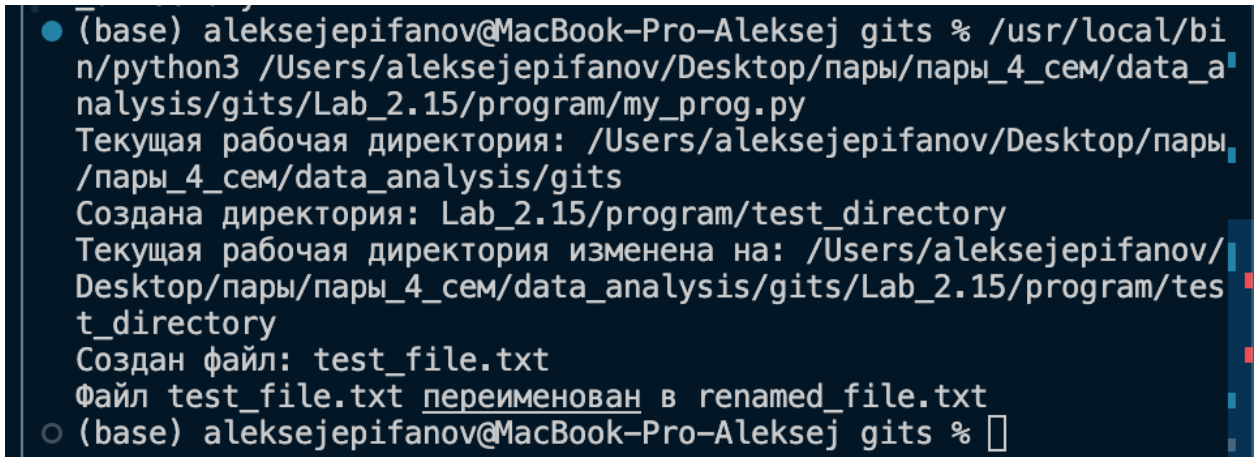
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import os
if __name__ == "__main__":
    current_directory = os.getcwd()
    print(f"Текущая рабочая директория: {current_directory}")
    new_dir = "Lab_2.15/program/test_directory"
    os.mkdir(new_dir)
    print(f"Создана директория: {new_dir}")
```



```

os.chdir(new_dir)
print(f"Текущая рабочая директория изменена на: {os.getcwd()}")
file_name = "test_file.txt"
with open(file_name, 'w') as my_file:
    my_file.write("Привет, мир!")
print(f"Создан файл: {file_name}")
new_file_name = "renamed_file.txt"
os.rename(file_name, new_file_name)
print(f"Файл {file_name} переименован в {new_file_name}")

```



```

(base) aleksejepifanov@MacBook-Pro-Aleksej gits % /usr/local/bin/python3 /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/my_prog.py
Текущая рабочая директория: /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits
Создана директория: Lab_2.15/program/test_directory
Текущая рабочая директория изменена на: /Users/aleksejepifanov/Desktop/пары/пары_4_сем/data_analysis/gits/Lab_2.15/program/test_directory
Создан файл: test_file.txt
Файл test_file.txt переименован в renamed_file.txt
(base) aleksejepifanov@MacBook-Pro-Aleksej gits %

```

Рисунок 22. Результат работы программы my_prog.py

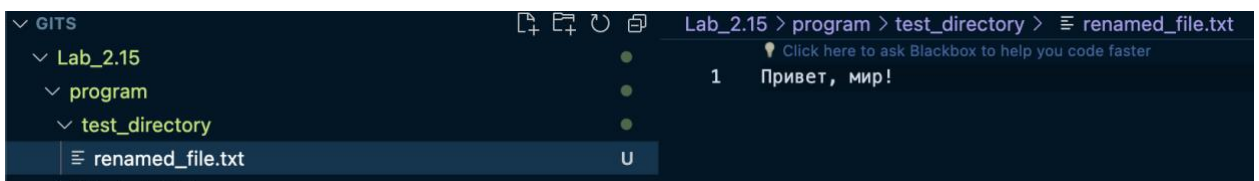


Рисунок 23. Полученные директория и файл

Ответы на контрольные вопросы:

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

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?

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.

3. Как прочитать данные из файла в языке Python?

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).

4. Как записать данные в файл в языке Python?

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

5. Как закрыть файл в языке Python?

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

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

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.

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

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.

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

`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.