

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №2.15  
Работа с файлами в языке Python  
по дисциплине «Технологии программирования»**

Выполнил студент группы ИВТ-б-о-20-1

Ищенко Т.С. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверила Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2021

Цель работы: 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 работы:

### 1. Произвёл выполнение 1-го примера

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

if __name__ == '__main__':
    # open the file2.txt in append mode. Create a new file if no such file exists.
    with open("file2.txt", "w") as fileptr:
        # appending the content to the file
        fileptr.write(
            "Python is the modern day language. It makes things so simple.\n"
            "It is the fastest-growing programming language"
        )
```

Рисунок 1 – Результат выполнения 1-го примера

### 2. Произвёл выполнение 2-го примера

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

if __name__ == '__main__':
    # open the file2.txt in write mode.
    with open("file2.txt", "a") as fileptr:
        # overwriting the content of the file
        fileptr.write(" Python has an easy syntax and user-friendly interaction.")
```

Рисунок 2 – Результат выполнения 2-го примера

### 3. Произвёл выполнение 3-го примера

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

if __name__ == '__main__':
    # open the file2.txt in read mode. causes error if no such file exists.
    with open("file2.txt", "r") as fileptr:
        # stores all the data of the file into the variable content
        content1 = fileptr.readline()
        content2 = fileptr.readline()
        # prints the content of the file
        print(content1)
        print(content2)
```

Рисунок 3 – Результат выполнения 3-го примера

#### 4. Произвёл выполнение 4-го примера

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

if __name__ == '__main__':
    # open the file2.txt in read mode. causes error if no such file exists.
    with open("file2.txt", "r") as fileptr:
        # stores all the data of the file into the variable content
        content = fileptr.readlines()
        # prints the content of the file
        print(content)
```

Рисунок 4 – Результат выполнения 4-го примера

#### 5. Произвёл выполнение 5-го примера

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

if __name__ == '__main__':
    # open the newfile.txt in read mode. causes error if no such file exists.
    with open("newfile.txt", "x") as fileptr:
        print(fileptr)

    if fileptr:
        print("File created successfully")
```

Рисунок 5 – Результат выполнения 5-го примера

#### 6. Произвёл выполнение 6-го примера

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

if __name__ == "__main__":
    # open the text.txt in append mode. Create a new file if no such file exists.
    with open("text.txt", "a", encoding="utf-8") as fileptr:
        # appending the content to the file
        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 6 – Результат выполнения 6-го примера

## 7. Произвёл выполнение 7-го примера

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

if __name__ == "__main__":
    with open("text.txt", "r", encoding="utf-8") as f:
        sentences = f.readlines()

    # Вывод предложений с запятыми.
    for sentence in sentences:
        if "," in sentence:
            print(sentence)
```

Рисунок 7 – Результат выполнения 7-го примера

## 8. Произвёл выполнение 8-го примера

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

if __name__ == "__main__":
    # open the file file2.txt in read mode
    with open("file2.txt", "r") as fileptr:
        # initially the filepointer is at 0
        print("The filepointer is at byte :", fileptr.tell())
        # changing the file pointer location to 10.
        fileptr.seek(10)

        # tell() returns the location of the fileptr.
        print("After reading, the filepointer is at:", fileptr.tell())
```

Рисунок 8 – Результат выполнения 8-го примера

#### 9. Произвёл выполнение 9-го примера

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

import os

if __name__ == "__main__":
    # rename file2.txt to file3.txt
    os.rename("file2.txt", "file3.txt")
```

Рисунок 9 – Результат выполнения 9-го примера

#### 10. Произвёл выполнение 10-го примера

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

import os

if __name__ == "__main__":
    # deleting the file named file3.txt
    os.remove("file3.txt")
```

Рисунок 10 – Результат выполнения 10-го примера

11.Произвёл выполнение 11-го примера

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

import os

if __name__ == "__main__":
    # creating a new directory with the name new
    os.mkdir("new")
```

Рисунок 11 – Результат выполнения 11-го примера

12.Произвёл выполнение 12-го примера

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

import os

if __name__ == "__main__":
    path = os.getcwd()
    print(path)
```

Рисунок 12 – Результат выполнения 12-го примера

13.Произвёл выполнение 13-го примера

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

import os

if __name__ == "__main__":
    # Changing current directory with the new directory
    os.chdir("C:\\Windows")
    # It will display the current working directory
    print(os.getcwd())
```

Рисунок 13 – Результат выполнения 13-го примера

14.Произвёл выполнение 14-го примера

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

import os

if __name__ == "__main__":
    # removing the new directory
    os.rmdir("new")
```

Рисунок 14 – Результат выполнения 14-го примера

15. Произвёл выполнение 15-го примера

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

import sys

if __name__ == "__main__":
    print("Number of arguments:", len(sys.argv), "arguments")
    print("Argument List:", str(sys.argv))
```

Рисунок 15 – Результат выполнения 15-го примера

16. Произвёл выполнение 16-го примера



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

import sys

if __name__ == "__main__":
    for idx, arg in enumerate(sys.argv):
        print(f"Argument #{idx} is {arg}")
    print("No. of arguments passed is ", len(sys.argv))
```

Рисунок 16 – Результат выполнения 16-го примера

#### 17. Произвёл выполнение 17-го примера

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

import ...

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("The password length is not given!", file=sys.stderr)
        sys.exit(1)

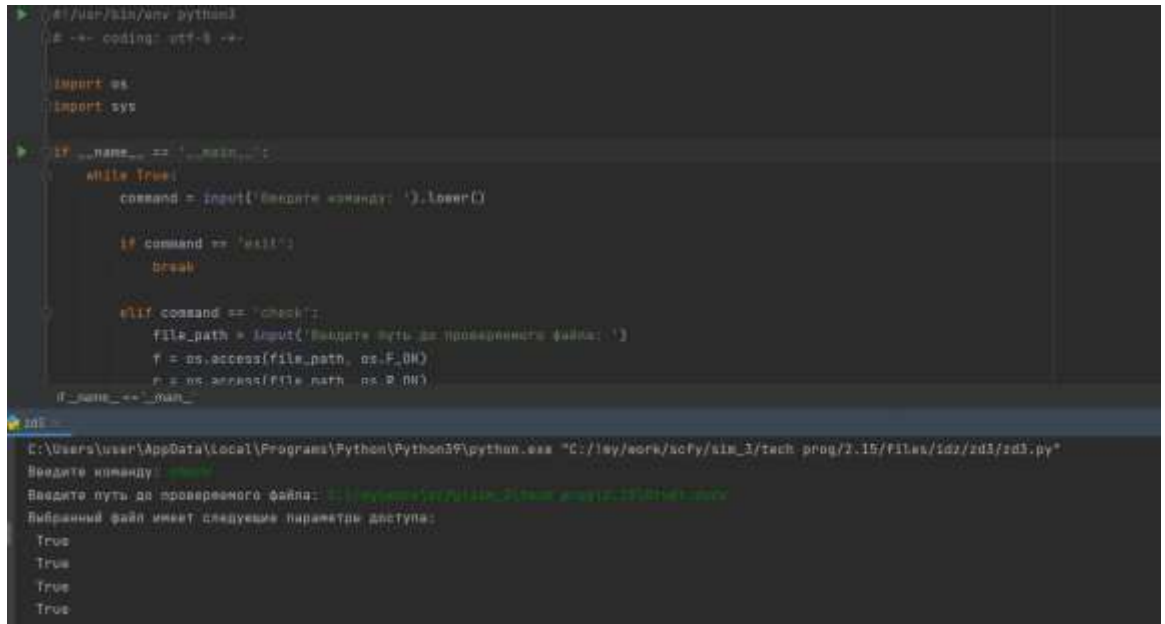
    chars = string.ascii_letters + string.punctuation + string.digits
    length_pwd = int(sys.argv[1])

    result = []
    for _ in range(length_pwd):
        idx = secrets.SystemRandom().randrange(len(chars))
        result.append(chars[idx])

    print(f"Secret Password: {''.join(result)}")
```

Рисунок 17 – Результат выполнения 17-го примера

18.Произвёл выполнение самостоятельно придуманного задания с модулем os.



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

import os
import sys

if __name__ == '__main__':
    while True:
        command = input('Введите команду: ').lower()

        if command == 'exit':
            break

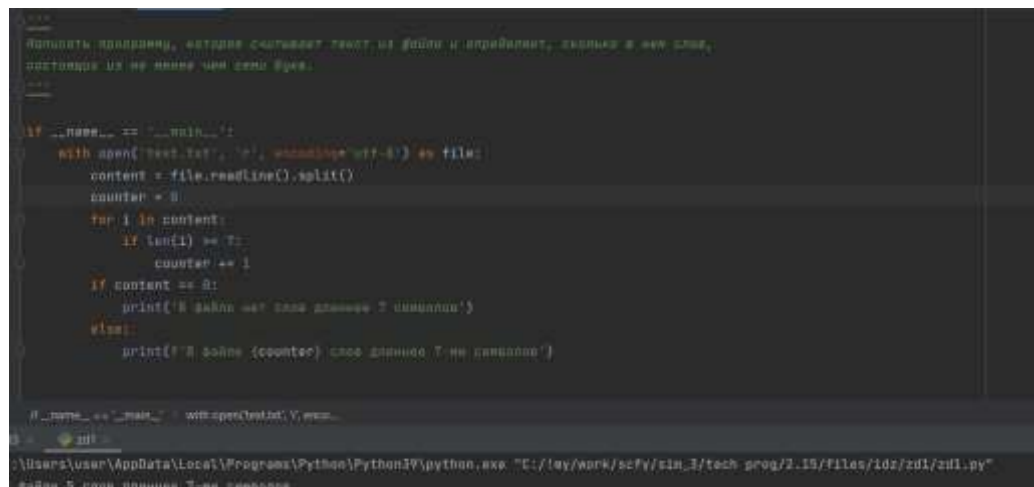
        elif command == 'check':
            file_path = input('Введите путь до проверяемого файла: ')
            f = os.access(file_path, os.F_OK)
            p = os.access(file_path, os.R_OK)

            if __name__ == '__main__':

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "C:/my/work/scfy/sim_3/tech_prog/2.15/files/1d2/1d1/1d1.py"
Введите команду: check
Введите путь до проверяемого файла: C:/my/work/scfy/sim_3/tech_prog/1d2/1d1/1d1.py
Выбранный файл имеет следующие параметры доступа:
True
True
True
True
```

Рисунок 18 – Результат выполнения задания с модулем os

19.Выполнил первое индивидуальное задание



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

import os

if __name__ == '__main__':
    with open('text.txt', 'r', encoding='utf-8') as file:
        content = file.readline().split()
        counter = 0

        for i in content:
            if len(i) == 1:
                counter += 1

            if content == 'a':
                print('В файле нет слов, начинающихся с буквы a')
            else:
                print(f'В файле {counter} слов, начинающихся с буквы a')

if __name__ == '__main__':
    with open('text.txt', 'r', encoding='utf-8') as file:

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "C:/my/work/scfy/sim_3/tech_prog/2.15/files/1d2/1d1/1d1.py"
В файле 5 слов, начинающихся с буквы a
```

Рисунок 19 – Результат выполнения 1-го индивидуального задания

20.Выполнил второе индивидуальное задание

```
import collections

if __name__ == '__main__':
    file_name = input('Введите имя файла с расширением: ')
    table = str.maketrans({' ': None, '-': None, '_': None, '!': None})
    slova = collections.Counter()

    with open(file_name, 'r', encoding='utf-8') as file:
        for i in file:
            content = i.lower().translate(table).split()
            for path in content:
                slova[path] += 1

    print(
        slova.most_common(
            int(input("Сколько самых частых слов вы хотите увидеть? "))
        )
    )

if __name__ == '__main__': with open(file_name, 'r', encoding='utf-8') as file: for i in file: for path in content
```

Python Shell

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe "C:/ispy/work/scfy/sim\_3/tech prog/2.15/files/ldz/zd2/ldz2.py"

Введите имя файла с расширением: text.txt

Сколько самых частых слов вы хотите увидеть? 4

[('ut', 4), ('voluptatem', 4), ('quia', 4)]

Рисунок 20 – Результат выполнения второго индивидуального задания

#### Контрольные вопросы:

1. Как открыть файл в языке Python только для чтения? – Используя функцию `open()`, после ввода имени файла через запятую указать режим “r”.
2. Как открыть файл в языке Python только для записи? – Используя функцию `open()`, после ввода имени файла через запятую указать режим “w”.
3. Как прочитать данные из файла в языке Python? – Сначала необходимо открыть файл, вызвав функцию `open()`, затем использовать метод `read()`.
4. Как записать данные в файл в языке Python? – Сначала необходимо открыть файл, вызвав функцию `open()`, затем использовать метод `write()`.
5. Как закрыть файл в языке Python? – Использовать метод `close()` или открывать файл при помощи оператора `with`, который закрывает файл, после окончания работы с ним.
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` гарантирует, что критические функции выполнятся в любом случае. В основном она используется для работы с файлами разного типа, но также может использоваться для фиксации

или отката транзакции базы данных, для перенаправления стандартного вывода однопоточных программ.

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла? – Метод `writelines()` – добавляет последовательность строк в файл. Метод `tell()` - возвращает текущую позицию “условного курсора” в файле.

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

- `os.name` - имя операционной системы.
- `os.environ` - словарь переменных окружения.
- `os.getpid()` - текущий `id` процесса.
- `os.uname()` - информация об ОС.
- `os.access ()` - проверка доступа к объекту у текущего пользователя.
- `os.chdir (path)` - смена текущей директории.
- `os.chmod ()` - смена прав доступа к объекту.
- `os.link ()` - создаёт жёсткую ссылку.
- `os.listdir ()` - список файлов и директорий в папке.
- `os.makedirs ()` - создаёт директорию, создавая при этом промежуточные директории.
- `os.symlink ()` - создаёт символическую ссылку на объект.
- `os.truncate ()` - обрезает файл до длины `length`.
- `os.utime ()` - модификация времени последнего доступа и изменения файла.
- `os.walk ()` - генерация имён файлов в дереве каталогов.
- `os.system ()` - исполняет системную команду, возвращает код её завершения.
- `os.urandom (n)` - `n` случайных байт.

- `os.path` - модуль, реализующий некоторые полезные функции на работы с путями.

Вывод: в ходе выполнения лабораторной работы 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 и получения аргументов командной строки.