

**РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное**  
**образовательное учреждение высшего образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**«Работа с файловой системой в Python3 с использованием модуля  
pathlib»**

**Отчет по лабораторной работе № 2.19**  
**по дисциплине «Программирование на Python»**

Выполнил студент группы ИВТ-б-о-21-1

Стригалов Дмитрий.

«19» декабря 2022г.

Подпись студента \_\_\_\_\_

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

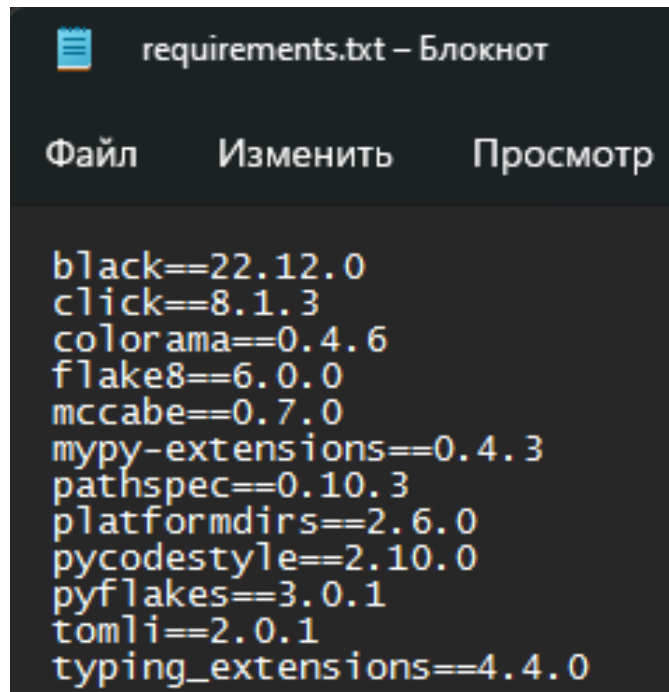
Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2022

**Цель работы:** приобретение навыков по работе с файловой системой с помощью библиотеки pathlib языка программирования Python версии 3.x.

**Порядок выполнения работы:**

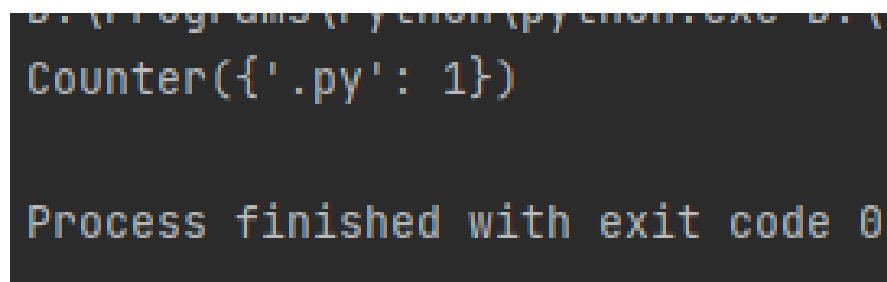
Формирование файла requirements.txt.



```
black==22.12.0
click==8.1.3
colorama==0.4.6
flake8==6.0.0
mccabe==0.7.0
mypy-extensions==0.4.3
pathspec==0.10.3
platformdirs==2.6.0
pycodestyle==2.10.0
pyflakes==3.0.1
tomli==2.0.1
typing_extensions==4.4.0
```

Рисунок 1 - Файл requirements.txt

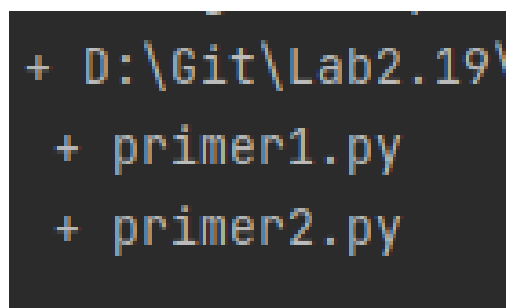
Проработать примеры лабораторной работы.



```
D:\Programs\Python\python.exe D:\p
Counter({' .py': 1})

Process finished with exit code 0
```

Рисунок 2 – Результат выполнения примера 1



```
+ D:\Git\Lab2.19\
+ primer1.py
+ primer2.py
```

Рисунок 3 - Результат выполнения примера 2

```
D:\Programs\Python\python.exe D:\Git\Lab2.19\Пример\primer3.py
2023-01-11 15:53:00.601528 D:\Git\Lab2.19\Пример\primer3.py

Process finished with exit code 0
```

Рисунок 4 - Результат выполнения примера 3

```
D:\Programs\Python\python.exe D:\Git\Lab2.19\Пример\primer4.py
D:\Git\Lab2.19\Пример\test001.txt

Process finished with exit code 0
```

Рисунок 5 - Результат выполнения примера 4

Выполнить индивидуальные задания.

### Задание 1

Для своего варианта лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения.

```
D:\Git\Lab2.19\Индивидуальные задания>python ind1.py display products.json
```

№	Название товара	Цена	Название магазина
1	шляпа	2500.00	adidas
2	носки	300.00	nike

Рисунок 6 - Результат выполнения задания 1

### Задание 2

Самостоятельно изучите работу с пакетом python-dotenv. Модифицируйте программу задания 1 таким образом, чтобы значения необходимых переменных окружения считывались из файла .env.

```
d:\Git\Lab2.19\Индивидуальные задания>python ind2.py
>>> d:\Git\Lab2.19\Индивидуальные задания
>> ind1.py
>> ind2.py

d:\Git\Lab2.19\Индивидуальные задания>python ind2.py mkdir .abc
>>> d:\Git\Lab2.19\Индивидуальные задания
>> .abc
>> ind1.py
>> ind2.py

d:\Git\Lab2.19\Индивидуальные задания>python ind2.py rmdir .abc
>>> d:\Git\Lab2.19\Индивидуальные задания
>> ind1.py
>> ind2.py

d:\Git\Lab2.19\Индивидуальные задания>python ind2.py touch Products.txt
>>> d:\Git\Lab2.19\Индивидуальные задания
>> ind1.py
>> ind2.py
>> Products.txt

d:\Git\Lab2.19\Индивидуальные задания>python ind2.py rm Products.txt
>>> d:\Git\Lab2.19\Индивидуальные задания
>> ind1.py
>> ind2.py
```

Рисунок 7 - Результат выполнения задания 2

### Контрольные вопросы:

**1. Какие существовали средства для работы с файловой системой до Python 3.4?**

- Методы строк, например `path.rsplitt('\\', maxsplit=1)[0]`
- Модуль `os.path`

**2. Что регламентирует PEP 428?**

Модуль `Pathlib` – Объектно-ориентированные пути файловой системы

**3. Как осуществляется создание путей средствами модуля `pathlib`?**

Есть несколько разных способов создания пути. Прежде всего, существуют classmethods наподобие `.cwd()` (текущий рабочий каталог) и `.home()` (домашний каталог вашего пользователя)

**4. Как получить путь дочернего элемента файловой системы с помощью модуля `pathlib`?**

При помощи метода `resolve()`.

**5. Как получить путь к родительским элементам файловой системы с помощью модуля `pathlib`?**

При помощи свойства `parent`.

**6. Как выполняются операции с файлами с помощью модуля `pathlib`?**

- перемещение;
- удаление файлов;
- подсчёт файлов;
- найти последний изменённый файл;
- создать уникальное имя файла; – чтение и запись файлов.

**7. Как можно выделить компоненты пути файловой системы с помощью модуля `pathlib`?**

`.name`  
`.parent`  
`.stem`  
`.suffix`  
`.anchor`

**8. Как выполнить перемещение и удаление файлов с помощью модуля `pathlib`?**

`.replace()` – метод перемещения файлов  
`.unlink()` – метод удаления файлов

**9. Как выполнить подсчет файлов в файловой системе?**

Метод `.iterdir()`

**10. Как отобразить дерево каталогов файловой системы? `def tree(directory): print(f' {directory}') for path in sorted(directory.rglob('*')):`**

`depth = len(path.relative_to(directory).parts)`  
`spacer = ' ' * depth print(f'{spacer} {path.name}')`

**11. Как создать уникальное имя файла?** `def unique_path(directory, name_pattern):`  
    `counter = 0 while True: counter += 1 path =`  
    `directory/name_pattern.format(counter) if not`  
    `path.exists():`  
        `return path`  
    `path = unique_path(pathlib.Path.cwd(), 'test{:03d}.txt')`

**12. Каковы отличия в использовании модуля `pathlib` для различных операционных систем?**

Ранее мы отмечали, что когда мы создавали экземпляр `pathlib.Path`, возвращался либо объект `WindowsPath`, либо `PosixPath`. Тип объекта будет зависеть от операционной системы, которую вы используете. Эта функция позволяет довольно легко писать кроссплатформенный код. Можно явно запросить `WindowsPath` или `PosixPath`, но вы будете ограничивать свой код только этой системой без каких-либо преимуществ. Такой конкретный путь не может быть использован в другой системе

**Вывод:** были приобретены навыки по работе с файловой системой с помощью библиотеки `pathlib` языка программирования Python версии 3.x.