

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ

УНИВЕРСИТЕТ» Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.19

Работа с файловой системой в Python3 с использованием модуля pathlib

по дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-21-1

Лысенко И.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверила 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.

Ход работы:

1. Проработал и разобрал примеры.

```
C:\Users\user\AppData\Local\Programs\Python\Python39
Counter({'': 6, '.py': 3, '.json': 1, '.md': 1})
```

Рисунок 1.1 – Результат работы примера 1

```
C:\Users\user\Documents\Pa6Cтол\R\lab_2_19\test001.txt

Process finished with exit code 0
```

Рисунок 1.2 – Результат работы примера 2

```
C:\Users\user\AppData\Local\Programs\Python\Pyt
+ C:\Users\user\Documents\Pa6Cтол\R\lab_2_19
+ .git
+ config
+ description
+ FETCH_HEAD
+ HEAD
+ hooks
+ applypatch-msg.sample
+ commit-msg.sample
```

Рисунок 1.3 – Результат работы примера 3

2. Добавил возможность хранения файла данных в домашнем каталоге пользователя для своего варианта.

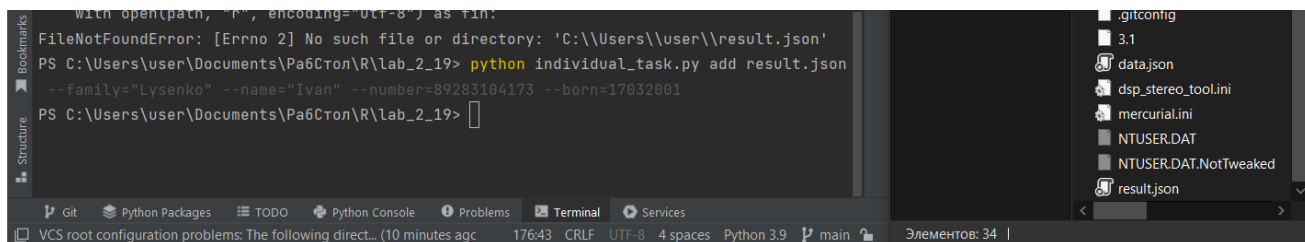


Рисунок 2.1 – Результат работы

3. Разработал аналог утилиты tree в Linux.

```
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтол\R\lab_2_19\individ> python individual_task
2.py
>>> C:\Users\user\Documents\Pa6Cтол\R\lab_2_19\individ
>>> Individual_Task.py
>>> Individual_Task2.py
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтол\R\lab_2_19\individ>
```

Рисунок 3.1 – Проверка содержимого каталога

```
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ> python individual_task
2.py mkdir .Task1
>>> C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ
>> .Task1
>> Individual_Task.py
>> Individual_Task2.py
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ> python individual_task
2.py rmdir .Task1
>>> C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ
>> Individual_Task.py
>> Individual_Task2.py
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ>
```

Рисунок 3.2 – Создание и удаление каталога

```
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ> python individual_task
2.py touch Task1.txt
>>> C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ
>> Individual_Task.py
>> Individual_Task2.py
>> Task1.txt
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ> python individual_task
2.py rm Task1.txt
>>> C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ
>> Individual_Task.py
>> Individual_Task2.py
(lab_2_19) PS C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ>
```

Рисунок 3.3 – Создание и удаление файла

4. Проверил программы через flake8.

```
C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ> flake8
C:\Users\user\Documents\Pa6Cтoл\R\lab_2_19\individ>
```

Рисунок 4.1 – Проверка flake8

Ответы на вопросы.

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

- Методы строк, например `path.split("\\", 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.

