

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

Отчет по лабораторной работе № 2.19

Работа с файловой системе в Python3 с использованием модуля pathlib

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.

Проверил(а) _____

Ставрополь 2022

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

Создан общедоступный репозиторий на GitHub. Дополнен файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.

Выполнено индивидуальное задание, рис. 1-3

```
(lab2.19) C:\Users\maks\doc\lab2.19>python ind1.py add j.json -d="Киев" -n="666" --time="6:36" --home  
(lab2.19) C:\Users\maks\doc\lab2.19>
```

Рисунок 1 – Добавление маршрута

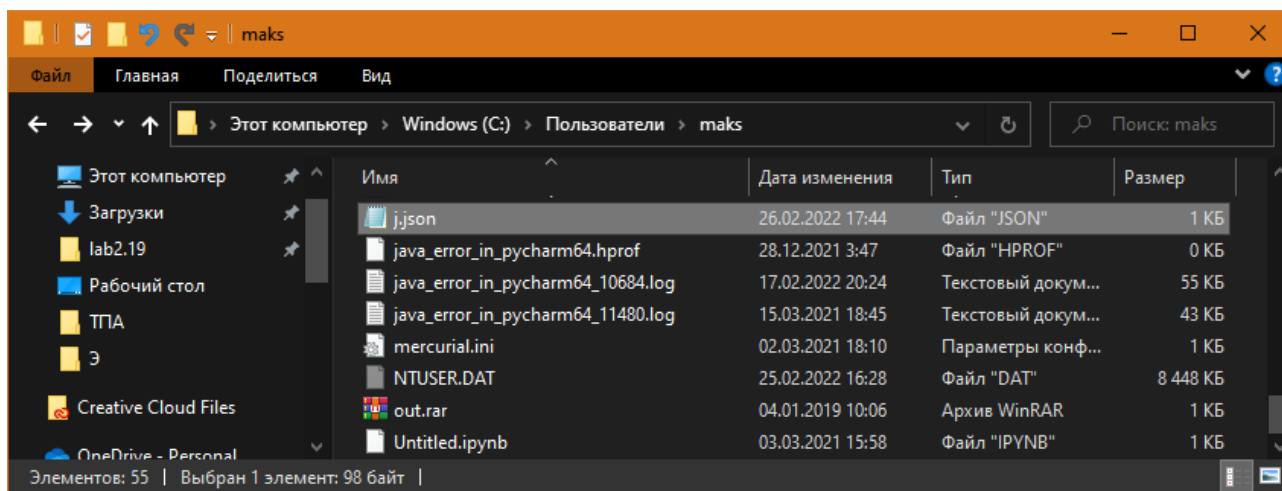


Рисунок 2 – Выгружаемые программой данные

Выполнено второе задание, рис. 3-4

```
Terminal: Local x +  
(lab2.19) C:\Users\maks\doc\lab2.19>python ind2.py  
> C:\Users\maks\doc\lab2.19  
  > 1  
    < 1.mp3  
    < 1.mp4  
    < 1.py  
  > 2  
    < primer4.py  
    < i.png  
    < i.py  
  < ind1.py  
  < ind2.py  
  < LICENSE  
  < primer.py  
  < primer2.py  
  < primer3.py  
  < README.md  
(lab2.19) C:\Users\maks\doc\lab2.19>
```

Рисунок 3 – Вывод дерева каталога

```
Terminal: Local x +
(lab2.19) C:\Users\maks\doc\lab2.19>python ind2.py find .py
> C:\Users\maks\doc\lab2.19
  < ind1.py
  < ind2.py
  < primer.py
  < primer2.py
  < primer3.py
> C:\Users\maks\doc\lab2.19\1
  < 1.py
  < i.py
> C:\Users\maks\doc\lab2.19\1\2
  < primer4.py

(lab2.19) C:\Users\maks\doc\lab2.19>
```

Рисунок 4 – Поиск файлов по расширению

Программы проверены на flake8, рис. 5

```
Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\maks> activate tools
(base) PS C:\Users\maks> cd doc\lab2.19
(base) PS C:\Users\maks\doc\lab2.19> flake8
(base) PS C:\Users\maks\doc\lab2.19>
```

Рисунок 5 – Проверка заданий

Контрольные 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?

Когда мы создаем экземпляр `pathlib.Path`, возвращается либо объект `WindowsPath`, либо `PosixPath`. Тип объекта будет зависеть от операционной системы. Эта функция позволяет довольно легко писать кроссплатформенный код. Можно явно запросить `WindowsPath` или `PosixPath`, но код будет ограничен только этой системой без каких-либо преимуществ. Такой конкретный путь не может быть использован в другой системе.

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