

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 21(2.19)  
Работа с файловой системой Python3 с использованием модуля pathlib  
по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Колбасов В.С. « » \_\_\_\_\_ 20\_\_ г.

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. Ход работы

1. Изучив методические указания, приступил к выполнению индивидуальных заданий.
2. Используя код из лабораторной работы 2.17, сделал задание, дополнив его новой изученной функцией библиотеки `pathlib`.

```
(lab2.19) C:\Users\zligo\Documents\GitHub\demo-2.19>python zadanie.py add new.json -s London -v Airbus -n RF-86123
```

Рисунок 21.1 – Ввод данных

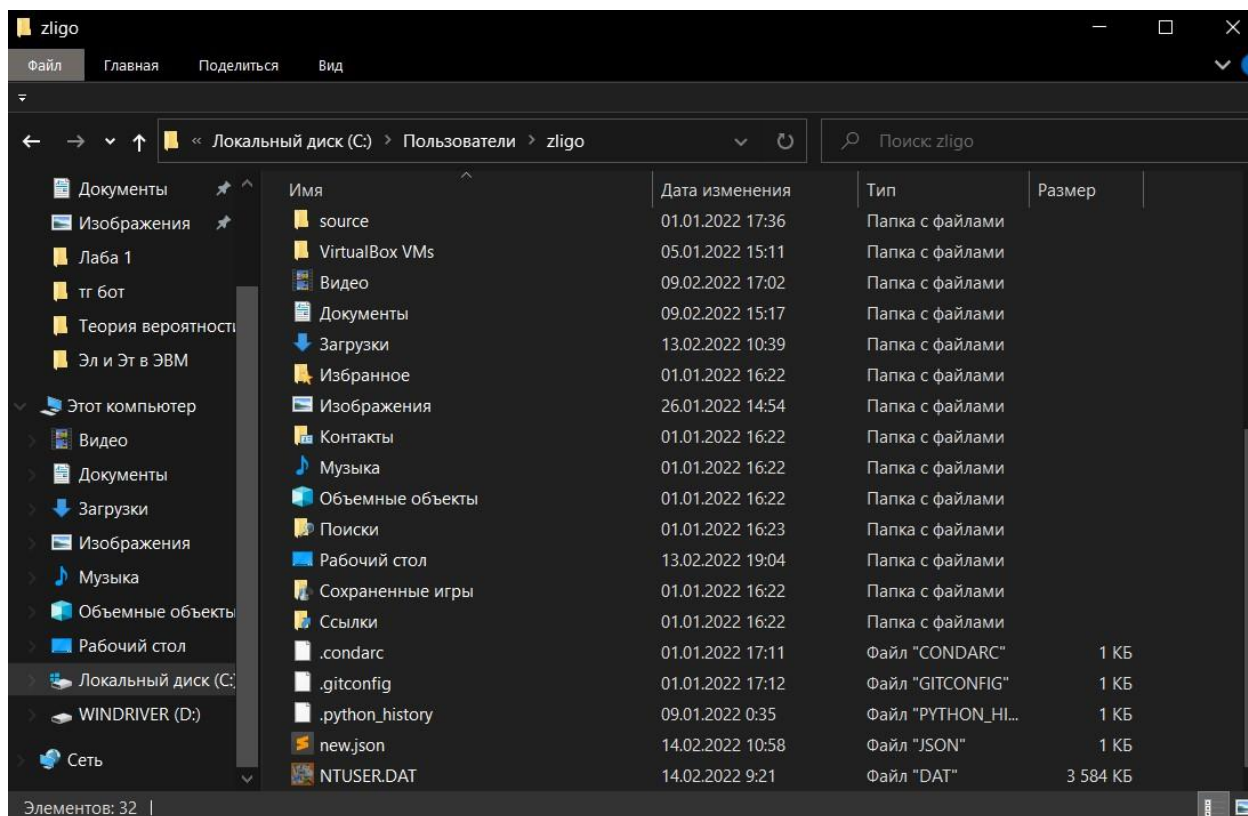


Рисунок 21.2 – Проверка наличия нового файла

```
[
    {
        "stay": "London",
        "number": "RF-86123",
        "value": "Airbus"
    }
]
```

Рисунок 21.3 – Проверка файла

3. После чего приступил к выполнению второго индивидуального задания.
4. Для этого самостоятельно изучил и подключил новую библиотеку colorama.

```
(lab2.19) C:\Users\zligo\Documents\GitHub\demo-2.19>python zadanie2.py
|-- C:\Users\zligo\Documents\GitHub\demo-2.19
|-- ← .git
|   |-- ← config
|   |-- ← description
|   |-- ← HEAD
|   |-- ← hooks
|   |-- ← index
|   |-- ← info
|   |-- ← logs
|   |-- ← objects
|   |-- ← packed-refs
|   |-- ← refs
|-- ← .gitignore
|-- ← Doc
|   |-- ← ~$Отчёт.DOCX
|   |-- ← Отчёт.DOCX
|-- ← LICENSE
|-- ← README.md
|-- ← zadanie.py
|-- ← zadanie2.py
```

Рисунок 21.4 – Проверка кода без указания параметров

```
(lab2.19) C:\Users\zligo\Documents\GitHub\demo-2.19>python zadanie2.py cd Doc
|-- C:\Users\zligo\Documents\GitHub\demo-2.19\Doc
|   |-- ← ~$Отчёт.DOCX
|   |-- ← Отчёт.DOCX
```

Рисунок 21.5 – Проверка кода с указанием дочернего каталога

```

(lab2.19) C:\Users\zligo\Documents\GitHub\demo-2.19>python zadanie2.py back demo-2.19
|-- C:\Users\zligo\Documents\GitHub
    |-- ← %SystemDrive%
        |-- ← ProgramData
    |-- ← demo-2.17
        |-- ← .git
        |-- ← .gitignore
        |-- ← Doc
        |-- ← environment.yml
        |-- ← LICENSE
        |-- ← README.md
        |-- ← requirements.txt
        |-- ← Zadanie
    |-- ← demo-2.19
        |-- ← .git
        |-- ← .gitignore
        |-- ← Doc
        |-- ← LICENSE
        |-- ← README.md
        |-- ← zadanie.py
        |-- ← zadanie2.py
    |-- ← TodoTasksList
        |-- ← .git
        |-- ← .gitignore
        |-- ← LICENSE
        |-- ← README.md
    |-- ← проекты
        |-- ← %SystemDrive%
        |-- ← ООП
        |-- ← проекты
        |-- ← ТПА

```

Рисунок 21.6 – Проверка кода с переходом в родительский каталог

## Контрольные вопросы

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

- Методы строк;
- модуль `os.path`.

2. Что регламентирует PEP 428?

Модуль `pathlib` -- пути объектно-ориентированной файловой системы 3.

Как осуществляется создание путей средствами модуля `pathlib`?

Есть несколько разных способов создания пути. Прежде всего, существуют `classmethods` класса `pathlib.Path`.

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: в ходе выполнения лабораторной работы были приобретены навыки по работе с библиотеками `pathlib` и `colorama`, а также закреплены знания по использованию библиотеки `argparse`.