

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
дисциплины «Анализ данных»

Выполнил:
Говоров Егор Юрьевич
2 курс, группа ИВТ-б-о-22-1,
01.05.2024 «Информатика
и вычислительная
техника», направленность
(профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Работа с файловой системе в Python3 с использованием модуля pathlib

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

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствие с необходимыми требованиями.
4. Выполнил индивидуальные задания, согласно варианту 5. Привёл в отчете скриншоты работы программ.

Задание. Для своего варианта лабораторной работы 2.17 добавьте возможность хранения файла данных в домашнем каталоге пользователя. Для выполнения операций с файлами необходимо использовать модуль pathlib .

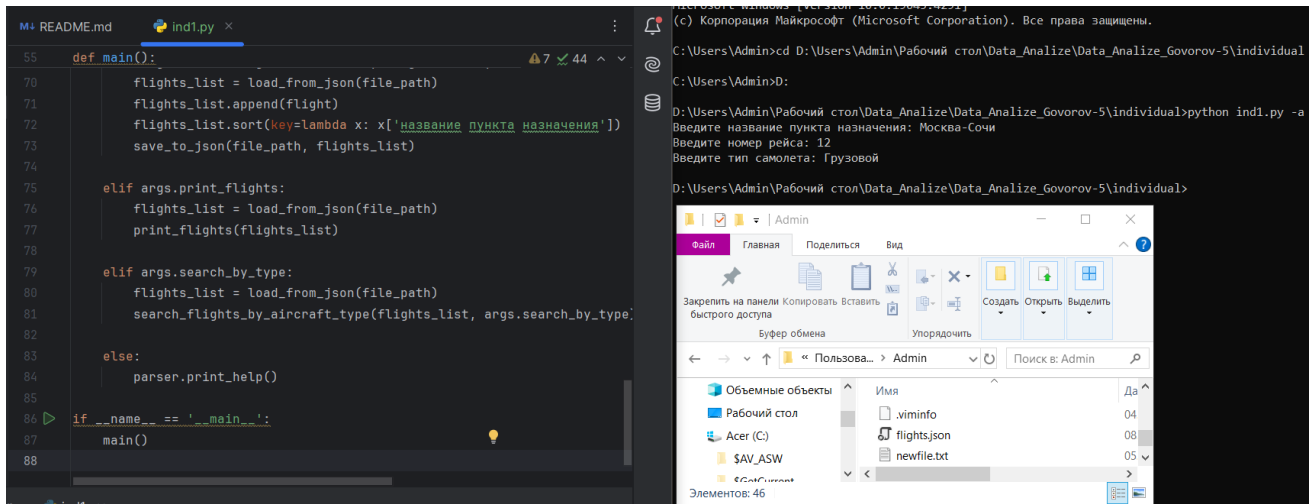


Рисунок 1. .json файл в домашнем каталоге

Задание. Разработайте аналог утилиты `tree` в Linux. Используйте возможности модуля `argparse` для управления отображением дерева каталогов файловой системы. Добавьте дополнительные уникальные возможности в данный программный продукт.

Рисунок 3. Результат работы программы задания 2

1. Средства для работы с файловой системой до Python 3.4

2. PEP 428

3. Создание путей средствами модуля pathlib

для создания пути к файлу можно использовать следующий синтаксис:

```
python
from pathlib import Path
path = Path('каталог/файл.txt')
```

4. Получение пути дочернего элемента файловой системы с помощью модуля `pathlib`

Для получения пути дочернего элемента файловой системы с помощью модуля `pathlib` можно использовать метод `joinpath()`. Например:

```
python
from pathlib import Path
path = Path('родительский_путь')
child_path = path.joinpath('дочерний_элемент')
```

5. Получение пути к родительским элементам файловой системы с помощью модуля `pathlib`

Для получения пути к родительским элементам файловой системы с помощью модуля `pathlib` можно использовать метод `parent`. Например:

```
python
from pathlib import Path
path = Path('путь/к/файлу')
parent_path = path.parent
```

6. Операции с файлами с помощью модуля `pathlib`

Модуль `pathlib` предоставляет удобные методы для выполнения операций с файлами, такие как проверка существования, чтение, запись, удаление и многое другое. Например, для проверки существования файла можно использовать метод `exists()`, а для удаления файла – метод `unlink()`.

7. Выделение компонентов пути файловой системы с помощью модуля `pathlib`

С помощью модуля `pathlib` можно выделить различные компоненты пути, такие как имя файла, расширение файла, имя каталога и другие. Например, для получения имени файла можно использовать метод `name`, а для получения расширения файла – метод `suffix`.

8. Отличия в использовании модуля `pathlib` для различных

операционных систем

Модуль `pathlib` предоставляет абстракцию от различий между операционными системами, что позволяет писать переносимый код для работы с файловой системой. Например, он автоматически учитывает различия в разделителях пути между Windows, Unix и другими операционными системами, что делает его использование удобным

9. Подсчет файлов в файловой системе

Для выполнения подсчета файлов в файловой системе с помощью модуля `pathlib` можно использовать метод `glob()`, который позволяет получить список файлов, соответствующих заданному шаблону. Например:

```
from pathlib import Path
path = Path('путь/к/каталогу')
file_count = sum(1 for _ in path.glob('**/*') if _.is_file())
```

10. Отображение дерева каталогов файловой системы

Для отображения дерева каталогов файловой системы с помощью модуля `pathlib` можно использовать метод `iterdir()`, который возвращает итератор по содержимому каталога. Например:

```
from pathlib import Path
def print_tree(path, indent=""):
    print(f'{indent}{path.name}/')
    for child in path.iterdir():
        if child.is_dir():
            print_tree(child, indent + ' ')
        else:
            print(f'{indent} {child.name}')
    print_tree(Path('путь/к/каталогу'))
```

11. Создание уникального имени файла

Для создания уникального имени файла с помощью модуля `pathlib` можно использовать метод `Path.with_name()`, который возвращает новый путь с указанным именем файла. Например:

```
from pathlib import Path
path = Path('путь/к/каталогу/файл.txt')
```

```
unique_name = path.with_name('уникальное_имя.txt')
```

12. Отличия в использовании модуля pathlib для различных операционных систем

Модуль `pathlib` предоставляет абстракцию от различий между операционными системами. Например, он автоматически учитывает различия в разделителях пути между Windows и Unix. Кроме того, он предоставляет классы, такие как `WindowsPath` и `PosixPath`, которые представляют пути в зависимости от операционной системы, что делает его использование удобным и универсальным

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