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»
Вариант 9

Выполнил:
Дудкин Константин Александрович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»
направление «Программное
обеспечение средств вычислительной
техники и автоматизированных
систем»,
очная форма обучения

_____
(подпись)

Руководитель практики:
Кандидат технических наук, доцент
кафедры инфокомуникаций, доцент
Воронкин Роман Александрович

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2024 г.

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

Цель: Приобрести навыки по работе с файловой системой с помощью библиотеки pathlib языка программирования Python версии 3.x.

Порядок выполнения работы

Выполнил задание 1: Нужно для программы из лабораторной работы 2.17 добавить возможность хранения файла данных в домашнем каталоге пользователя с помощью модуля pathlib:

```
# Найти файл по переменной окружения
data_file = args.f__filename
if not data_file:
    data_file = os.environ.get("ROUTES_FILE")
if not data_file:
    data_file = home() / 'routes.json'
```

Рисунок 1. Измененная часть кода (часть 1)

```
def import_json(file):
    file_path = Path(file)
    if not file_path.exists():
        print(f"Файла {file} не существует")
        return []
    else:
        with open(file, 'r', encoding='utf-8') as filein:
            return json.load(filein)
```

Рисунок 2. Измененная часть кода (часть 2)

```
(base) [code_ra1der@archlinux Progs]$ python individual.py list
Файла routes.json не существует
Список маршрутов пуст
(base) [code_ra1der@archlinux Progs]$ python individual.py add --first="Ставрополь" --second="Ессентуки"
(base) [code_ra1der@archlinux Progs]$ python individual.py list
+----------------+----------------------+----------------------+
| Номер маршрута |    Место отправки    |    Место прибытия    |
+----------------+----------------------+----------------------+
| 1              | Ставрополь           | Ессентуки            |
+----------------+----------------------+----------------------+
(base) [code_ra1der@archlinux Progs]$
```

Рисунок 3. Результат

Выполнил задание 2: Нужно разработать аналог утилиты tree в Linux, используя argparse для управления отображением дерева каталогов файловой системы:

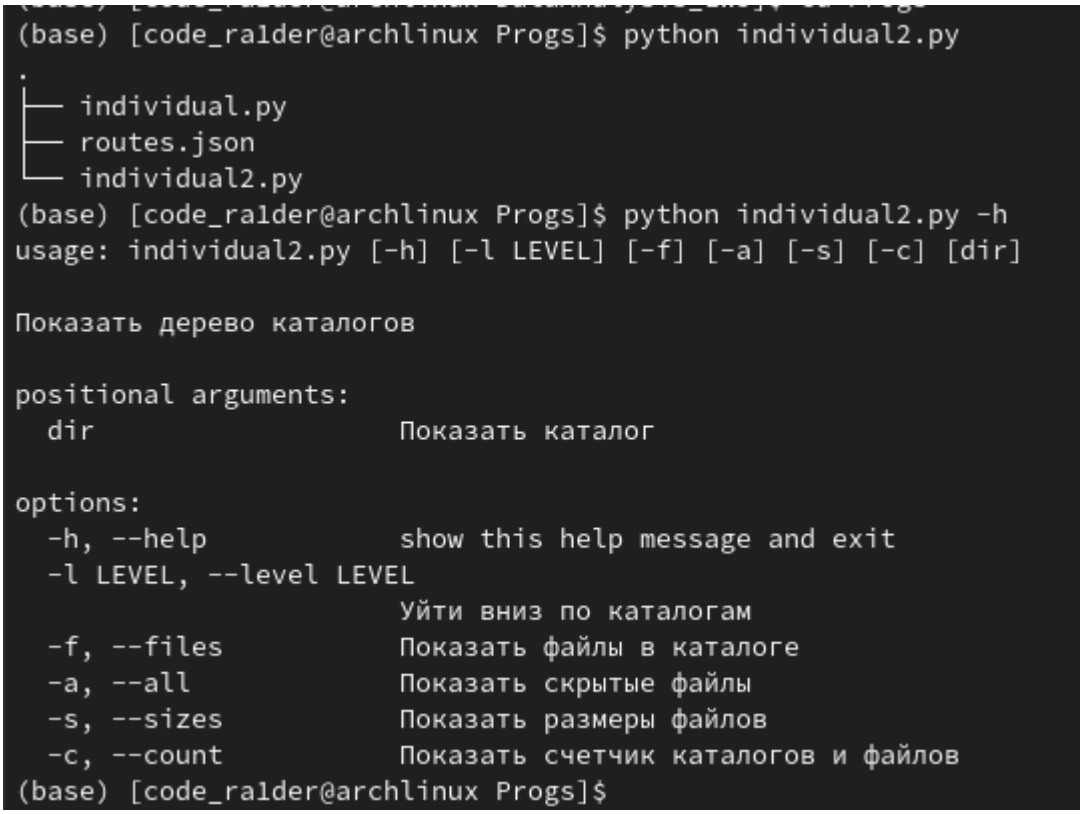

Рисунок 4. Результат

Ответы на вопросы:

1. Так как pathlib появился как раз в Python 3.4, то для работы с файловой системой можно было с помощью библиотек os, os.path, shutil и glob

2. PEP 428 регламентирует модуль pathlib, предоставляющий функции и интерфейс работы с файловой системой. Он способен манипулировать ею вне зависимости от типа операционной системы, а также предоставляет удобный и читаемый инструментарий для работы с данным модулем

3. Для этого вместе с данным модулем назначается следующее значение переменной:

path = Path(„path/to/file“)

4. Для получения пути дочернего элементы используется метод path.joinpath, но вместо него можно также использовать оператор /

5. Для получения пути к родителям можно испрользовать атрибут path.parent

6. Для выполнения операций над файлами можно использовать следующие методы модуля pathlib:

Для чтения: path.read_text()

Для записи: path.write_text()

Для проверки наличия файла: path.exists()

Для удаления: path.unlink()

7. Для того, чтобы выделить компоненты пути файловой системы, можно использовать следующие методы:

По имени: path.name

По расширению: path.suffix

По имени без расширения: path.stem

По всем частям пути: path.parts

8. Для перемещения и удаления используются методы rename и unlink

9. Для подсчета файлов в файловой системе можно использовать метод rglob

10. Для отображения дерева каталогов файловой системы можно использовать рекурсивный обход директорий и файлов:

```
from pathlib import Path
def show_tree(dir, prefix=“):
    print(prefix + dir.name)
      prefix += „| “ for path int sorted(dir.iterdir(), key=lambda p:(p.is file(), p.name.lower())):
        if path.is_dir(): show_tree(path, prefix)
        else:
          print(prefix + path.name)
start_dir = path(„путь/к каталогу“)
```

show_tree(start_dir)

11. Для этого можно воспользоваться модулем uuid

12. Между разными операционными системами можно наблюдать различие в использовании pathlib

Как пример, UNIX-системы и Windows по разному назначают пути к файлам и каталогам («/путь/к файлу» у UNIX и «C:\\путь\\к файлу» у Windows)

Вывод: В ходе лабораторной работы были приобретены навыки работы с файловой системов с помощью модуля pathlib в Python 3.x.