

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 №2.19
дисциплины «Анализ данных»

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

(подпись)

Руководитель практики:
Воронкин Р.А., 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: приобретение навыков по работе с файловой системой с помощью библиотеки pathlib языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.
2. Выполнил индивидуальное задание 1: 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.

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19> python ind.py -h
usage: students [-h] [--version] {add,display,select} ...

positional arguments:
  add                Add a new student
  display            Display all students
  select            Select the students

options:
  -h, --help            show this help message and exit
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19> python ind.py display --home ind.json
+-----+-----+-----+-----+
| № |      Ф.И.О.      | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Andrey A.A.      | 3      | 4 4 3 5 4 |
+-----+-----+-----+-----+
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19> python ind.py add -n "Shams V." -g 8 -gr "5 5 5 5 5" ind.json
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19> python ind.py display --home ind.json
+-----+-----+-----+-----+
| № |      Ф.И.О.      | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Andrey A.A.      | 3      | 4 4 3 5 4 |
+-----+-----+-----+-----+
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19> python ind.py display ind.json
+-----+-----+-----+-----+
| № |      Ф.И.О.      | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Magdaev D.M.     | 4      | 5 4 3 5 4 |
| 2 | Shams V.         | 8      | 5 5 5 5 5 |
+-----+-----+-----+-----+
```

Рисунок 1. Вывод программы индивидуального задания 1

3. Выполнил индивидуальное задание 2: 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.

```

(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19> python ind_hard.py --help
usage: ind_hard.py [-h] [-l LEVEL] [-d DIR] [-s SHOWSIZE] [-t TIME] [path]

Утилита для отображения дерева каталогов и файлов

positional arguments:
  path                  Путь к каталогу

options:
  -h, --help            show this help message and exit
  -l LEVEL, --level LEVEL
                        Уровень вложенности
  -d DIR, --dir DIR     Показывать только директории
  -s SHOWSIZE, --showsize SHOWSIZE
                        Показать размер файлов
  -t TIME, --time TIME  Показать время изменения
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19> python ind_hard.py
11.py
ind.json
ind.py
ind_hard.py
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.19>

```

Рисунок 2. Вывод программы аналога tree

Ответы на контрольные вопросы:

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

- Методы строк, например `path.rsplitle('\'', 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). Также можно создать путь при помощи: `pathlib.Path(«путь»)`.

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?

`source.replace(destination)`

– перемещение; `file_to_delete.unlink()`

– удаление файлов; `file_to_write.write_text('Привет, мир!')` `content = file_to_read.read_text()`

– чтение и запись файлов.

7. Как можно выделить компоненты пути файловой системы с помощью модуля pathlib?

`.name` : имя файла без какого-либо каталога

`.parent` : каталог, содержащий файл, или родительский каталог, если путь является каталогом

`.stem` : имя файла без суффикса `.suffix` : расширение файла

`.anchor` : часть пути перед каталогами

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?

`import collections`

`collections.Counter(p.suffix for p in pathlib.Path.cwd().iterdir())`

`Counter({'md': 2, 'txt': 4, 'pdf': 2, 'py': 1})`

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 результате выполнения лабораторной работы были приобретены навыки по работе с файловой системой с помощью библиотеки `pathlib` языка программирования Python версии 3.x.