

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №1

Работа с файловой системой в Python3 с использованием модуля pathlib

По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « » _____ 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.

Ссылка на репозиторий: <https://github.com/DIMITRY-GALYAS1/Laba-2.19.git>

Ход работы:

1. Создал новый репозиторий на github, после клонировал его и создал в папке репозитория новый проект PyCharm.
2. Выполнил первый пример.



The screenshot shows a PyCharm IDE with a Python script in the editor and its execution output in the console. The script is as follows:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import pathlib
5 import collections
6
7
8 if __name__ == '__main__':
9     print(collections.Counter(p.suffix for p in pathlib.Path.cwd().iterdir()))
10
```

The console output shows the execution of the script, displaying the Counter object and the exit code:

```
primer_1 x
C:\Users\lizeq\anaconda3\envs\pythonProject14\python.exe C:/Users/lizeq/PycharmProjects/pythonProject14/primer_1.py
Counter({'py': 5, '': 1, '.json': 1})
Process finished with exit code 0
```

Рисунок 1. Пример 1

3. Выполнил второй пример.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import pathlib
5
6
7 def tree(directory):
8     print(f'+ {directory}')
9     for path in sorted(directory.rglob('*')):
10         depth = len(path.relative_to(directory).parts)
11         spacer = ' ' * depth
12         print(f'{spacer}+ {path.name}')
13
14
15 if __name__ == '__main__':
16     tree(pathlib.Path.cwd())
17
```

primer_2

C:\Users\lizeq\anaconda3\envs\pythonProject14\python.exe C:/Users/lizeq/PycharmProjects/pythonProject14/primer_2.py
+ C:\Users\lizeq\PycharmProjects\pythonProject14
+ .idea
+ .gitignore
+ inspectionProfiles
+ profiles_settings.xml
+ misc.xml
+ modules.xml
+ pythonProject14.iml
+ workspace.xml
+ data.json

Рисунок 2. Пример 2

4. Выполнил третий пример.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from datetime import datetime
5 import pathlib
6
7
8 def tree(directory):
9     time, file_path = max((f.stat().st_mtime, f) for f in directory.iterdir())
10    print(datetime.fromtimestamp(time), file_path)
11
12
13 if __name__ == '__main__':
14     tree(pathlib.Path.cwd())
15     print(pathlib.Path.home())
16
```

primer_3

C:\Users\lizeq\anaconda3\envs\pythonProject14\python.exe C:/Users/lizeq/PycharmProjects/pythonProject14/primer_3.py
2022-02-13 16:42:08.854453 C:\Users\lizeq\PycharmProjects\pythonProject14\idea
C:\Users\lizeq
Process finished with exit code 0

Рисунок 3. Пример 3

5. Выполнил первое индивидуальное задание

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import json
import click
import pathlib

@click.group()
def cli():
    pass

@cli.command()
@click.argument('filename')
@click.option("-n", "--name")
@click.option("-g", "--group")
@click.option("-p", "--progress")
def add(filename, name, group, progress):
    """
    Добавление нового студента
    """
    students = open_file(filename)
    students.append(
        {
            'name': name,
            'group': group,
            'progress': progress
        }
    )

```

Рисунок 4. Код задания

```

{
    "name": "Nikolay",
    "group": "2",
    "progress": "2444"
}

```

Рисунок 5. Результат работы программы

6. Выполнил второе индивидуальное задание.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import argparse
import pathlib
import colorama
import collections

def tree(directory):
    print('\033[1;31m' + f'>>> {directory}')
    for path in sorted(directory.rglob('*')):
        depth = len(path.relative_to(directory).parts)
        spacer = '\t' * depth
        print('\033[1;32m' + f'{spacer} >> {path.name}')
        for new_path in sorted(directory.joinpath(path).rglob('*')):
            depth = len(new_path.relative_to(directory.joinpath(path)).parts)
            spacer = '\t\t' * depth
            print('\033[1;33m' + f'{spacer} > {new_path.name}')

def main(command_line=None):
    colorama.init()
    path = pathlib.Path.cwd()
    file_parser = argparse.ArgumentParser(add_help=False)

    parser = argparse.ArgumentParser("tree")
    parser.add_argument(
        "--version",
        action="version",
    )

```

Рисунок 6. Второе индивидуальное задание

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?

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 приобрел навыки по работе с файловой системой с помощью библиотеки `pathlib` языка программирования Python версии 3.x.