

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.19

Работа с файловой системой в Python3 с использованием модуля pathlib

По дисциплине «Теории программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

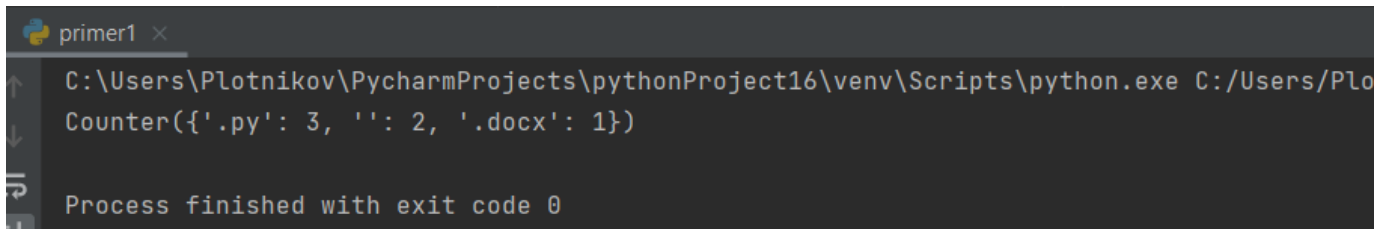
(подпись)

Ставрополь 2022

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

Ход работы

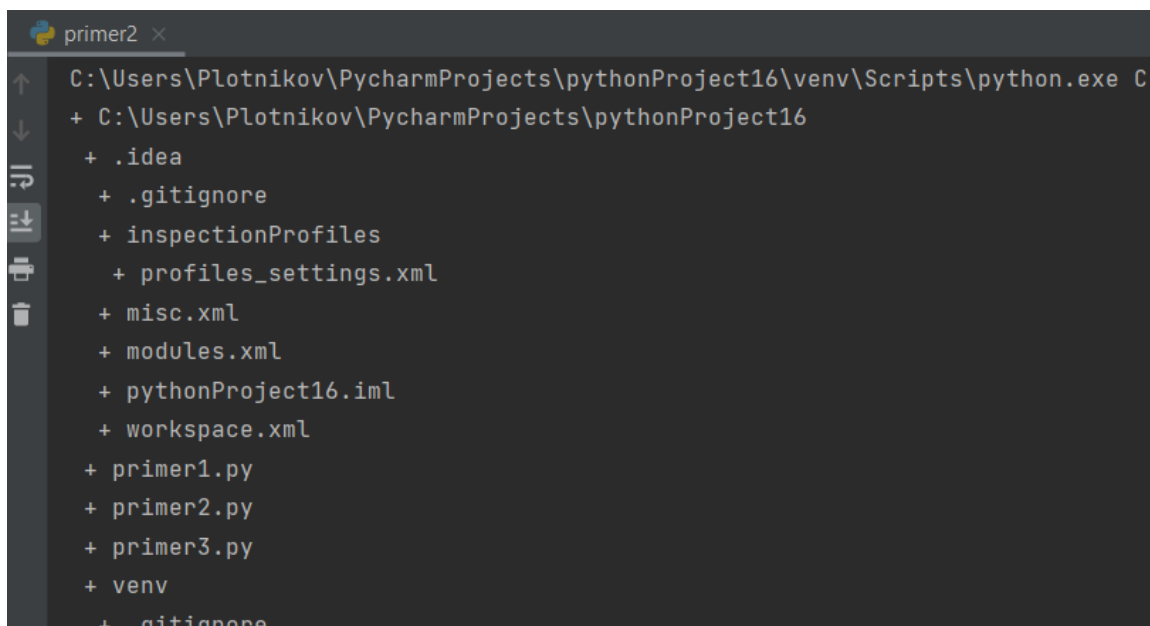
1. Создал новый собственный репозиторий. Ссылка на репозиторий: https://github.com/Dmitry-15/2.19_laba.
2. С помощью команды git clone клонировал удаленный репозиторий на свой ПК. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.
3. После ознакомления с теоретическим материалом приступил к выполнению примеров.



```
primer1 x
C:\Users\Plotnikov\PycharmProjects\pythonProject16\venv\Scripts\python.exe C:/Users/Pl...
Counter({'py': 3, '': 2, '.docx': 1})

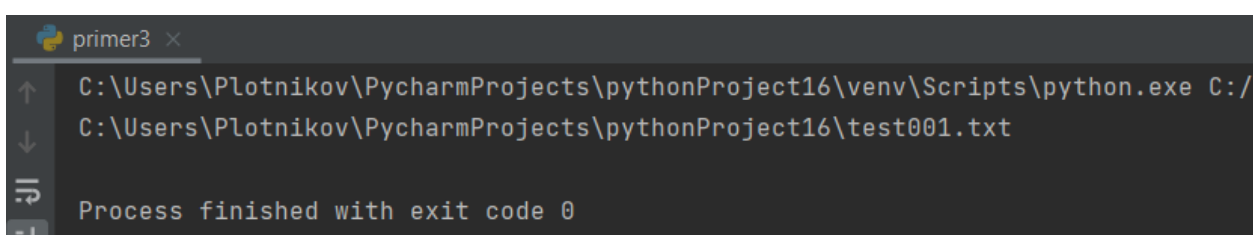
Process finished with exit code 0
```

Рисунок 1 – Результат выполнения первого примера



```
primer2 x
C:\Users\Plotnikov\PycharmProjects\pythonProject16\venv\Scripts\python.exe C:/...
+ C:\Users\Plotnikov\PycharmProjects\pythonProject16
+ .idea
+ .gitignore
+ inspectionProfiles
+ profiles_settings.xml
+ misc.xml
+ modules.xml
+ pythonProject16.iml
+ workspace.xml
+ primer1.py
+ primer2.py
+ primer3.py
+ venv
+ .gitignore
```

Рисунок 2 – Результат выполнения второго примера



```
primer3 x
C:\Users\Plotnikov\PycharmProjects\pythonProject16\venv\Scripts\python.exe C:/...
C:\Users\Plotnikov\PycharmProjects\pythonProject16\test001.txt

Process finished with exit code 0
```

Рисунок 3 – Результат выполнения третьего примера

Индивидуальные задания

Задание 1

1. Приступил к выполнению первого индивидуального задания.
2. Условие: 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`.

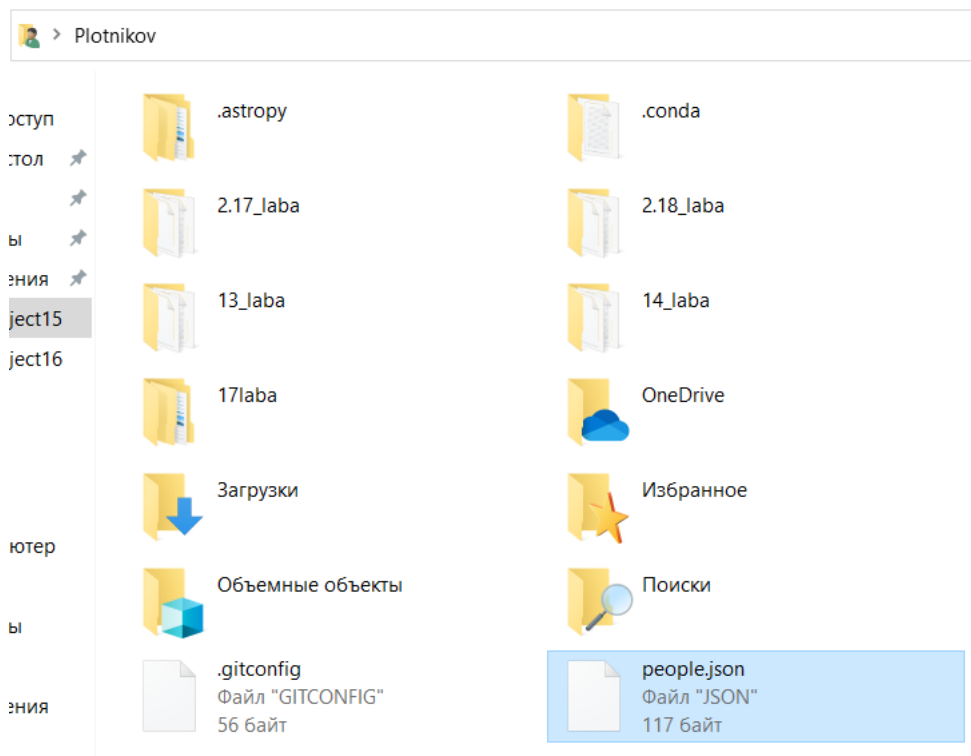


Рисунок 4 – Результат выполнения первого индивид. задания

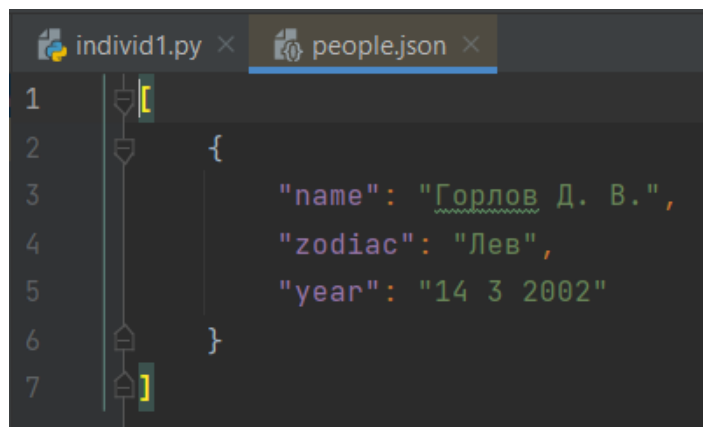


Рисунок 5 – Содержимое файла

Задание 2

1. Затем перешел к выполнению второго индивидуального задания.

2. Условие: 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.

```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\Plotnikov\PycharmProjects\pythonProject16> python individ2.py
>>> C:\Users\Plotnikov\PycharmProjects\pythonProject16
>> .idea
    > .gitignore
    > .name
    > inspectionProfiles
    > misc.xml
    > modules.xml
    > pythonProject16.iml
    > workspace.xml
>> .gitignore
>> .name
>> inspectionProfiles
    > profiles_settings.xml
>> profiles_settings.xml
>> misc.xml
>> modules.xml
>> pythonProject16.iml
>> workspace.xml
>> 123.txt
>> individ1.py
>> individ2.py
>> primer1.py
>> primer2.py
>> primer3.py
>> venv
    > .gitignore
```

Рисунок 6 – Результат выполнения второго индивид. Задания

```

Anaconda Powershell Prompt (Anaconda3)

(base) PS C:\Users\Plotnikov\PycharmProjects> cd pythonProject16
(base) PS C:\Users\Plotnikov\PycharmProjects\pythonProject16> python individ2.py touch 123.txt
>>> C:\Users\Plotnikov\PycharmProjects\pythonProject16
>>> .idea
>>> .gitignore
>>> .name
>>> inspectionProfiles
>>> misc.xml
>>> modules.xml
>>> pythonProject16.iml
>>> workspace.xml
>>> .gitignore
>>> .name
>>> inspectionProfiles
>>> profiles_settings.xml
>>> profiles_settings.xml
>>> misc.xml
>>> modules.xml
>>> pythonProject16.iml
>>> workspace.xml
>>> 123.txt
>>> individ1.py
>>> individ2.py
>>> primer1.py
>>> primer2.py
>>> primer3.py
>>> venv
>>> .gitignore
>>> lib

```

Рисунок 7 – Создание файла

```

Anaconda Powershell Prompt (Anaconda3)

(base) PS C:\Users\Plotnikov\PycharmProjects\pythonProject16> python individ2.py unlink 123.txt
>>> C:\Users\Plotnikov\PycharmProjects\pythonProject16
>>> .idea
>>> .gitignore
>>> .name
>>> inspectionProfiles
>>> misc.xml
>>> modules.xml
>>> pythonProject16.iml
>>> workspace.xml
>>> .gitignore
>>> .name
>>> inspectionProfiles
>>> profiles_settings.xml
>>> profiles_settings.xml
>>> misc.xml
>>> modules.xml
>>> pythonProject16.iml
>>> workspace.xml
>>> individ1.py
>>> individ2.py
>>> primer1.py
>>> primer2.py
>>> primer3.py
>>> venv
>>> .gitignore

```

Рисунок 8 – Удаление файла

```

Anaconda Powershell Prompt (Anaconda3)

2022-02-27 20:04:54.665927 C:\Users\Plotnikov\PycharmProjects\pythonProject16\~$чет2.19.docx
(base) PS C:\Users\Plotnikov\PycharmProjects\pythonProject16>

```

Рисунок 9 – Нахождение последнего измененного файла

```
Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\Plotnikov\PycharmProjects\pythonProject16> python individ2.py count pythonProject16
Counter({'py': 5, '': 2, '.docx': 2})
(base) PS C:\Users\Plotnikov\PycharmProjects\pythonProject16>
```

Рисунок 10 – Подсчет файлов директории

Контрольные вопросы:

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

- Методы строк, например `path.rsplitt("\\', 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?

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