

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Работа с файловой системой в Python3 с использованием модуля
pathlib»

Отчет по лабораторной работе № 2.19
по дисциплине «Программирование на Python»

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«19» декабря 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

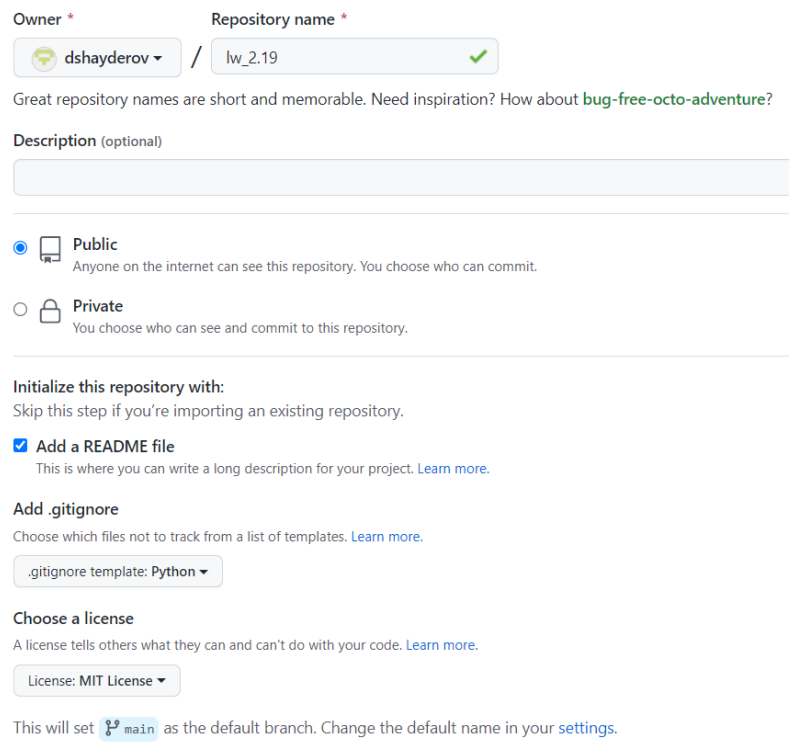
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022


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

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.



Owner * Repository name *

 dshayderov / lw_2.19

Great repository names are short and memorable. Need inspiration? How about [bug-free-octo-adventure?](#)

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

This will set `main` as the default branch. Change the default name in your [settings](#).

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

```
Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3
$ git clone https://github.com/dshayderov/lw_2.19.git
Cloning into 'lw_2.19'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```

Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3
$ cd lw_2.19

Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3/lw_2.19 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3/lw_2.19 (develop)
$

```

Рисунок 3 - Ветвление по модели git-flow

4. Создание виртуального окружения.

```

C:\My projects\3\lw_2.19>python -m venv .venv

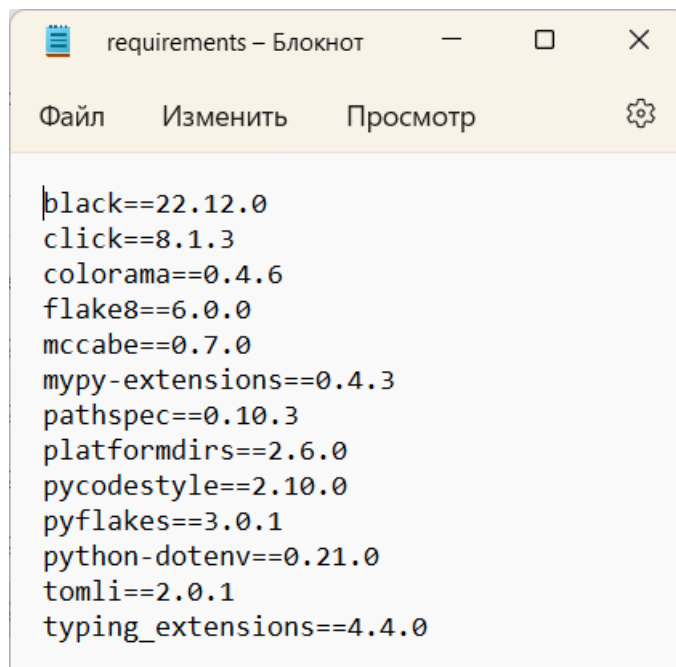
C:\My projects\3\lw_2.19>.venv\Scripts\activate

(.venv) C:\My projects\3\lw_2.19>pip install black flake8
Collecting black
  Using cached black-22.12.0-cp39-cp39-win_amd64.whl (1.2 MB)
Collecting flake8
  Using cached flake8-6.0.0-py2.py3-none-any.whl (57 kB)
Collecting typing-extensions>=3.10.0.0

```

Рисунок 4 - Виртуальное окружение

5. Формирование файла requirements.txt.



```

black==22.12.0
click==8.1.3
colorama==0.4.6
flake8==6.0.0
mccabe==0.7.0
mypy_extensions==0.4.3
pathspec==0.10.3
platformdirs==2.6.0
pycodestyle==2.10.0
pyflakes==3.0.1
python-dotenv==0.21.0
tomli==2.0.1
typing_extensions==4.4.0

```

Рисунок 5 - Файл requirements.txt

6. Создайте проект PyCharm в папке репозитория.

Этот компьютер > OS (C:) > My projects > 3 > lw_2.19 >				
Имя	Дата изменения	Тип	Размер	
.env	25.12.2022 20:51	Папка с файлами		
Project	25.12.2022 20:56	Папка с файлами		
.gitignore	25.12.2022 20:48	Файл "GITIGNORE"	2 КБ	
LICENSE	25.12.2022 20:48	Файл	2 КБ	
README.md	25.12.2022 20:48	Файл "MD"	1 КБ	
requirements	25.12.2022 20:55	Текстовый докум...	1 КБ	

Рисунок 6 - Создание проекта

7. Проработать примеры лабораторной работы.

```
primer_1 x
"C:\My projects\3\lw_2.19\.venv\Scripts\
Counter({'py': 1})
```

Рисунок 7 – Результат выполнения примера 1

```
primer_2 x
"C:\My projects\3\lw_2.19\.venv\Scripts\python.
+ C:\My projects\3\lw_2.19\Project\Примеры
+ primer_1.py
+ primer_2.py
```

Рисунок 8 - Результат выполнения примера 2

```
primer_3 x
"C:\My projects\3\lw_2.19\.venv\Scripts\python.exe" "C:/My projects/3/lw_2.19/Pr
2022-12-25 21:20:00.037258 C:\My projects\3\lw_2.19\Project\Примеры\primer_3.py
```

Рисунок 9 - Результат выполнения примера 3

```
primer_4 x
"C:\My projects\3\lw_2.19\.venv\Scripts\python.exe"
C:\My projects\3\lw_2.19\Project\Примеры\test001.txt
```

Рисунок 10 - Результат выполнения примера 4

8. Выполнить индивидуальные задания.

Задание 1

Для своего варианта лабораторной работы 2.17 добавьте возможность получения имени файла данных, используя соответствующую переменную окружения.

```
(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_1.py add planes.json --destination="Новгород" --num=900 --typ="грузовой"

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_1.py display planes.json
```

No	Пункт назначения	Номер рейса	Тип самолета
1	Пермь	123	пассажирский
2	Новгород	900	грузовой

Рисунок 11 - Результат выполнения задания 1

Задание 2

Самостоятельно изучите работу с пакетом `python-dotenv`. Модифицируйте программу задания 1 таким образом, чтобы значения необходимых переменных окружения считывались из файла `.env`.

```
(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_2.py
+ C:\My projects\3\lw_2.19\Project\Индивидуальные задания
+ ind_1.py
+ ind_2.py

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_2.py mkfile file.txt
+ C:\My projects\3\lw_2.19\Project\Индивидуальные задания
+ file.txt
+ ind_1.py
+ ind_2.py

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_2.py rmfile file.txt
+ C:\My projects\3\lw_2.19\Project\Индивидуальные задания
+ ind_1.py
+ ind_2.py

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_2.py mkdir path
+ C:\My projects\3\lw_2.19\Project\Индивидуальные задания
+ ind_1.py
+ ind_2.py
+ path

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_2.py rmdir path
+ C:\My projects\3\lw_2.19\Project\Индивидуальные задания
+ ind_1.py
+ ind_2.py

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_2.py count
Counter({'py': 2})

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>python ind_2.py last
2022-12-26 00:05:52.565114 ind_2.py

(.venv) C:\My projects\3\lw_2.19\Project\Индивидуальные задания>
```

Рисунок 11 - Результат выполнения задания 2

Контрольные вопросы:

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

- Методы строк, например `path.rsplit('\\', 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 `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: были приобретены навыки по работе с файловой системой с помощью библиотеки `pathlib` языка программирования Python версии 3.x.