

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

Толубаев Рамиль

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.

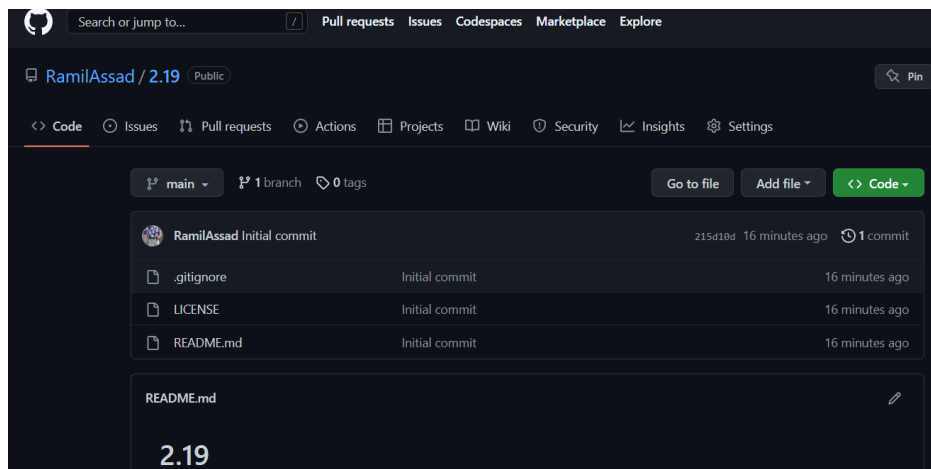


Рисунок 1 - Создание репозитория

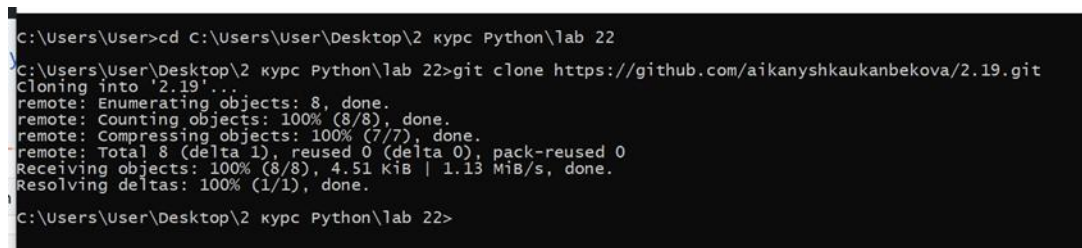


Рисунок 2 - Клонирование репозитория

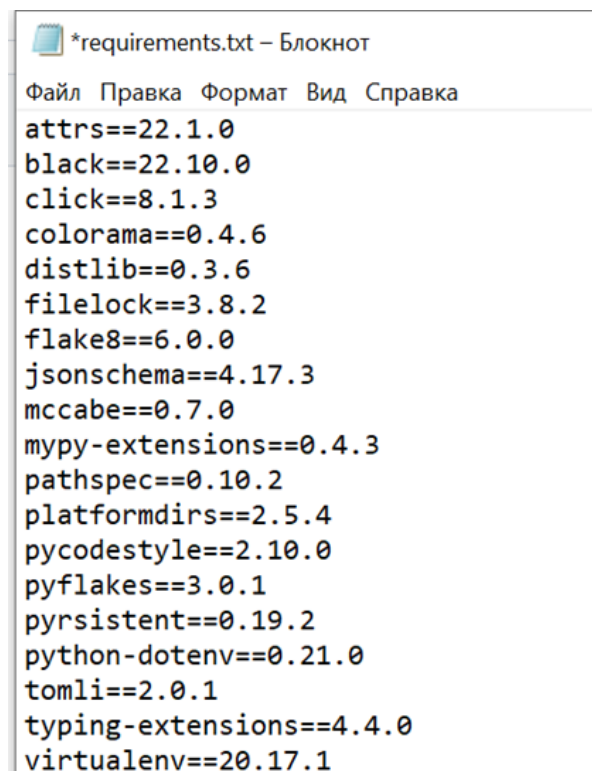
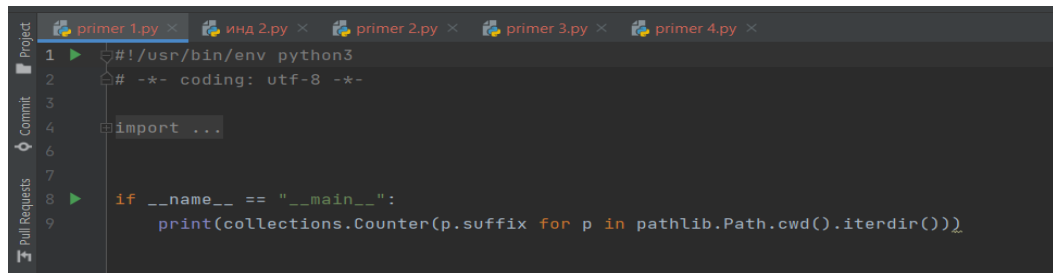



Рисунок 3 - Файл requirements.txt



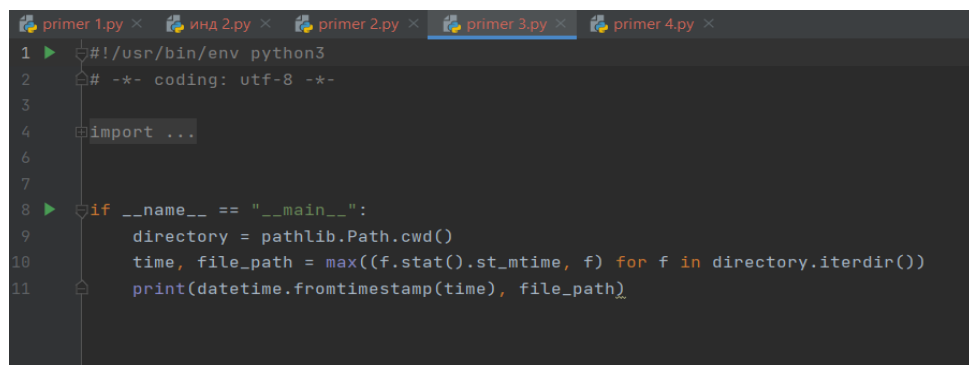
```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import ...
5
6
7
8 if __name__ == "__main__":
9     print(collections.Counter(p.suffix for p in pathlib.Path.cwd().iterdir()))
```

Рисунок 4 – Результат выполнения примера 1



```
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())
```

Рисунок 5 – Результат выполнения примера 2



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

Рисунок 6 – Результат выполнения примера 3


```

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

import ...

def tree(directory):
    print(Fore.BLUE + f'>>> {directory}')
    for path in sorted(directory.rglob('*')):
        print(Fore.YELLOW + f' >> {path.name}')
        for new_path in sorted(directory.joinpath(path).glob('*')):
            print(Fore.GREEN + f' > {new_path.name}')

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

    # Создаем основной парсер командной строки
    parser = argparse.ArgumentParser("tree")
    parser.add_argument(
        "--version",
        action="version",
        help="The main parser",
        version=f"%(prog)s 0.1.0"
    )

    subparsers = parser.add_subparsers(dest="command")

```

Рисунок 9 - Результат выполнения задания 2

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