

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.19

Дисциплина: «Программирование на Python»

Тема: «Работа в файловой системе в Python3 с использованием
модуля pathlib»

Выполнил: студент 2 курса

группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович

Ставрополь 2022


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

Ход работы:

1. Создать общедоступный репозиторий с лицензией MIT и языком Python.

Owner *


Repository name *

 umarkochkarov ▾


/ lb2.19 ✓

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

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](#).

 You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Клонировать репозиторий на ПК:

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.19
$ git clone https://github.com/umarkochkarov/lb2.19.git
Cloning into 'lb2.19'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2. Клонирование репозитория

3. Организовать репозиторий в соответствии с моделью ветвления git-flow.

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.19/lb2.19 (main)
$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/erken/Desktop/python/Ла62.19/lb2.19/.git/hooks]
```

Рисунок 3. Организация репозитория в соответствии с моделью git-flow

4. Создание виртуального окружения.

```
C:\Users\erken\Desktop\python\Ла62.19\lb2.19>python -m venv env
C:\Users\erken\Desktop\python\Ла62.19\lb2.19>env\Scripts\activate
(env) C:\Users\erken\Desktop\python\Ла62.19\lb2.19>pip install black flake8
Collecting black
  Using cached black-23.1.0-cp310-cp310-win_amd64.whl (1.2 MB)
Collecting flake8
  Using cached flake8-6.0.0-py2.py3-none-any.whl (57 kB)
Collecting packaging>=22.0
  Using cached packaging-23.0-py3-none-any.whl (42 kB)
Collecting mypy_extensions>=0.4.3
  Using cached mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
```

Рисунок 4. Создание виртуального окружения.

5. Проработка примеров из лабораторной работы:

```
(env) C:\Users\erken\Desktop\python\Ла62.19\lb2.19\prim>python prim1.py
Counter({'py': 3, '': 1})

(env) C:\Users\erken\Desktop\python\Ла62.19\lb2.19\prim>python prim2.py
+ C:\Users\erken\Desktop\python\Ла62.19\lb2.19\prim
+ .idea
+ .gitignore
+ inspectionProfiles
+ profiles_settings.xml
+ misc.xml
+ modules.xml
+ prim.iml
+ vcs.xml
+ workspace.xml
+ prim1.py
+ prim2.py
+ prim3.py

(env) C:\Users\erken\Desktop\python\Ла62.19\lb2.19\prim>python prim3.py
C:\Users\erken\Desktop\python\Ла62.19\lb2.19\prim\test001.txt
```

Рисунок 5. Проработка примеров

6. Выполнение индивидуальных заданий:

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`.

```
(env) C:\Users\erken\Desktop\python\Ла62.19\lb2.19\ind>python ind1.py display ind1.json
+-----+-----+-----+
|      No      |      Название      |      Время      |
+-----+-----+-----+
|          21  |      Terrain       |      15:00:00   |
+-----+-----+-----+
```

Рисунок 6. Индивидуальное задание 1

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.

```

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py
Введите"help" для вывода списка команд

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py help
all - просмотр полного пути файла
files - просмотр всех файлов в директории
seze - просмотр размера файла
save - сохранение данных в json-файл
mkdir - создание директории
rmdir - удаление директории
mk - создание файла
rm - удаление файла

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py all
C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py files
>> .idea
> .gitignore
> ind.iml
> inspectionProfiles
> misc.xml
> modules.xml
> vcs.xml
> workspace.xml
>> .gitignore
>> ind.iml
>> inspectionProfiles
> profiles_settings.xml
>> profiles_settings.xml
>> misc.xml
>> modules.xml
>> vcs.xml
>> workspace.xml
>> ind1.json
>> ind1.py
>> ind2.py

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py size ind1.json
>> ind1.json - 92 B

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py save ind2.json

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py mkdir test

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py mk test.txt

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py files
>> .idea
> .gitignore
> ind.iml
> inspectionProfiles
> misc.xml
> modules.xml
> vcs.xml
> workspace.xml
>> .gitignore
>> ind.iml
>> inspectionProfiles
> profiles_settings.xml
>> profiles_settings.xml
>> misc.xml
>> modules.xml

```

Рисунок 7. Результат работы программы

```

(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py rmdir test
(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py rm test.txt
(env) C:\Users\erken\Desktop\python\ла62.19\lb2.19\ind>python ind2.py files
>> .idea
    > .gitignore
    > ind.iml
    > inspectionProfiles
    > misc.xml
    > modules.xml
    > vcs.xml
    > workspace.xml
>> .gitignore
>> ind.iml
>> inspectionProfiles
    > profiles_settings.xml
>> profiles_settings.xml
>> misc.xml
>> modules.xml
>> vcs.xml
>> workspace.xml
>> files
>> ind1.json
>> ind1.py
>> ind2.json
>> ind2.py

```

Рисунок 8. Результат работы программы

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 `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 теоретические сведения и практические навыки для работы с файловой системой с помощью библиотек `pathlib` и `salorama` языка программирования Python версии 3.x.