# 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 |
| Урусов Максим Андреевич.  «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.

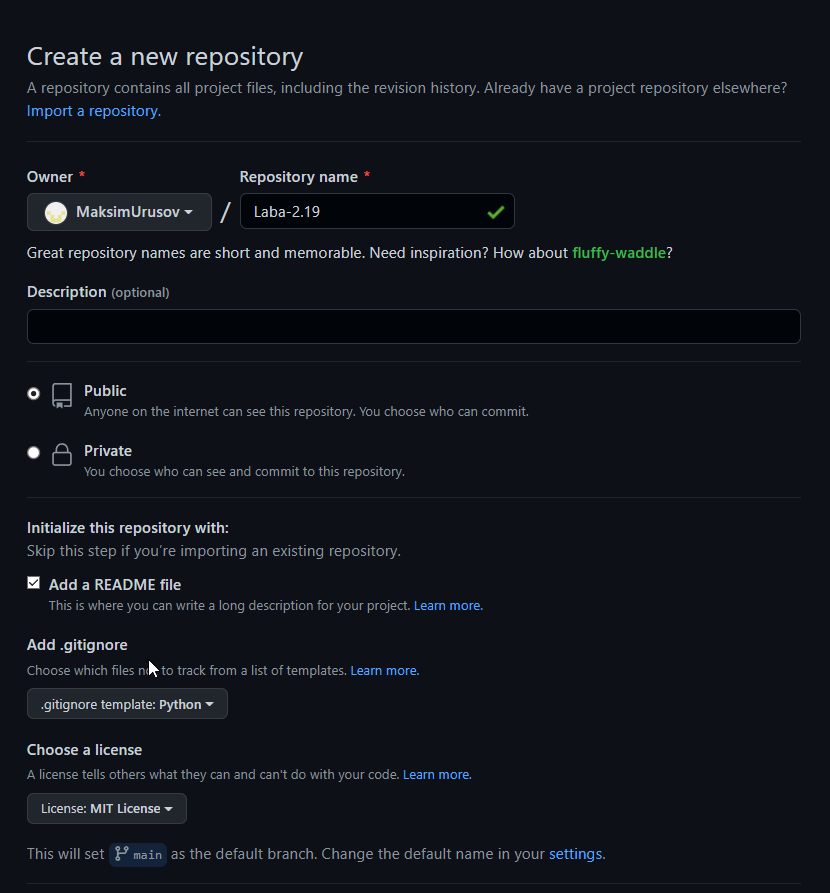


Рисунок 1 - Создание репозитория

1. Выполните клонирование созданного репозитория.

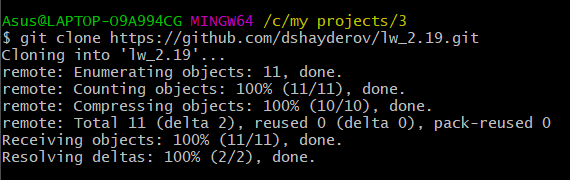


Рисунок 2 - Клонирование репозитория

1. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

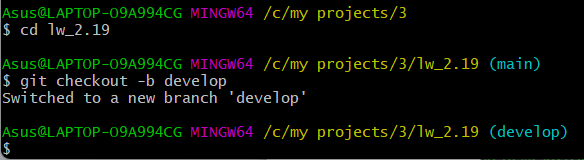


Рисунок 3 - Ветвление по модели git-flow

1. Создание виртуального окружения.

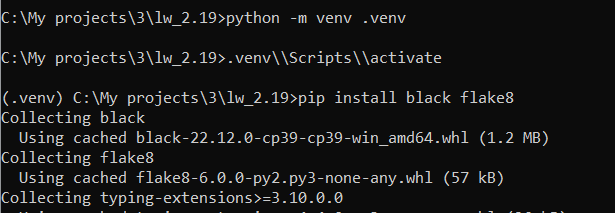


Рисунок 4 - Виртуальное окружение

1. Формирование файла requirements.txt.

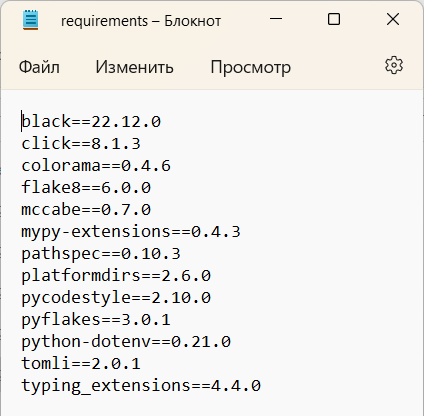


Рисунок 5 - Файл requirements.txt

1. Создайте проект PyCharm в папке репозитория.

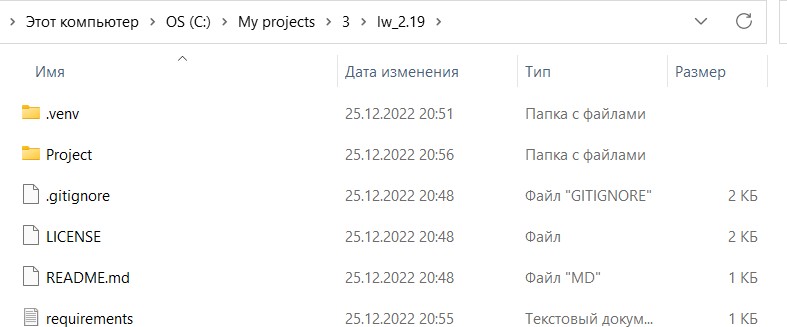


Рисунок 6 - Создание проекта

1. Проработать примеры лабораторной работы.

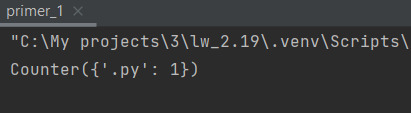


Рисунок 7 – Результат выполнения примера 1

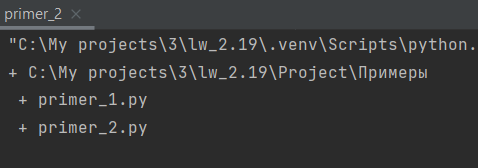


Рисунок 8 - Результат выполнения примера 2

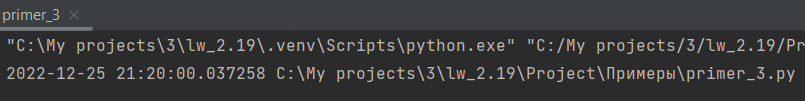


Рисунок 9 **-** Результат выполнения примера 3

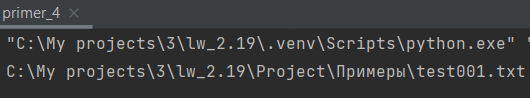


Рисунок 10 - Результат выполнения примера 4

1. Выполнить индивидуальные задания.

# Задание 1

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

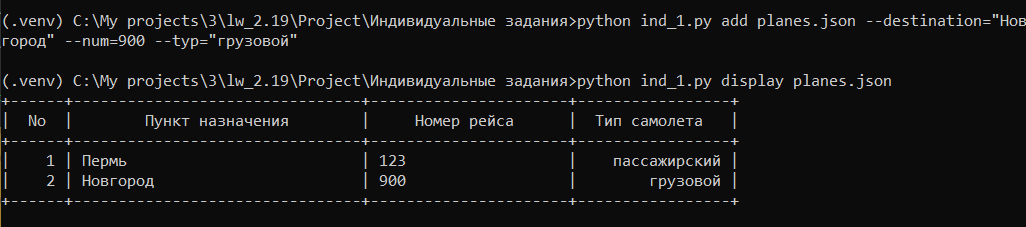


Рисунок 11 - Результат выполнения задания 1

# Задание 2

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

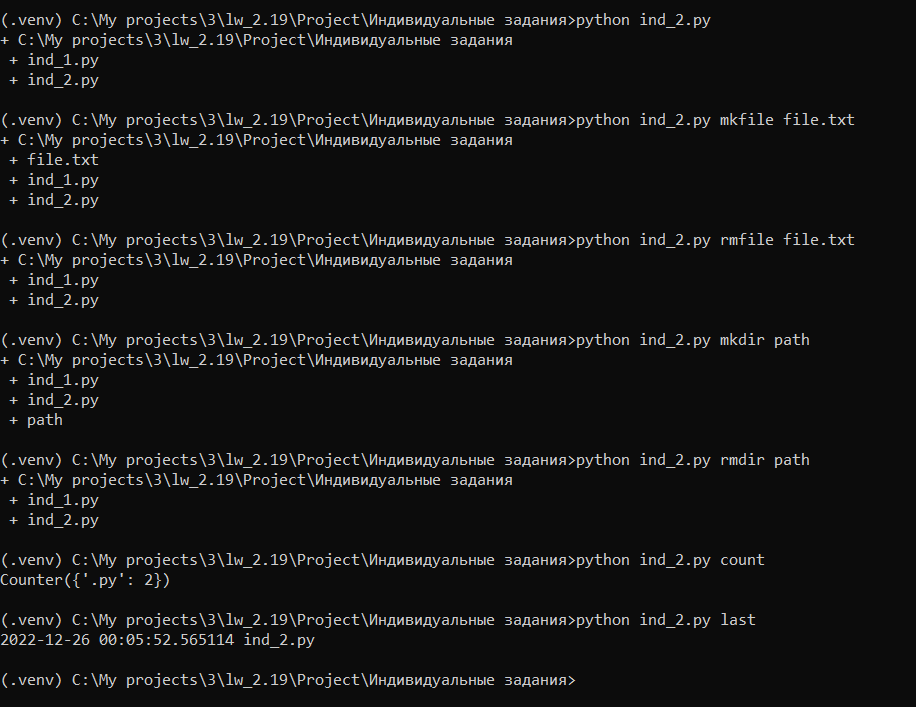


Рисунок 11 - Результат выполнения задания 2

# 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

# Что регламентирует PEP 428?

Модуль Pathlib – Объектно-ориентированные пути файловой системы

# 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)

# Как получить путь дочернего элемента файловой системы с помощью модуля pathlib?

При помощи метода resolve().

# Как получить путь к родительским элементам файловой системы с помощью модуля pathlib?

При помощи свойства parent.

# 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.

# Как можно выделить компоненты пути файловой системы с помощью модуля pathlib?

.name

.parent

.stem

.suffix

.anchor

# Как выполнить перемещение и удаление файлов с помощью модуля pathlib?

.replace() – метод перемещения файлов

.unlink() – метод удаления файлов

# Как выполнить подсчет файлов в файловой системе?

Метод .iterdir()

# 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}')

# 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')

# 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.