

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

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №2.19**

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

Выполнил студент группы ИВТ-б-о-21-1

Горшков Виталий Игоревич

« » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

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..

### Практическая часть:

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

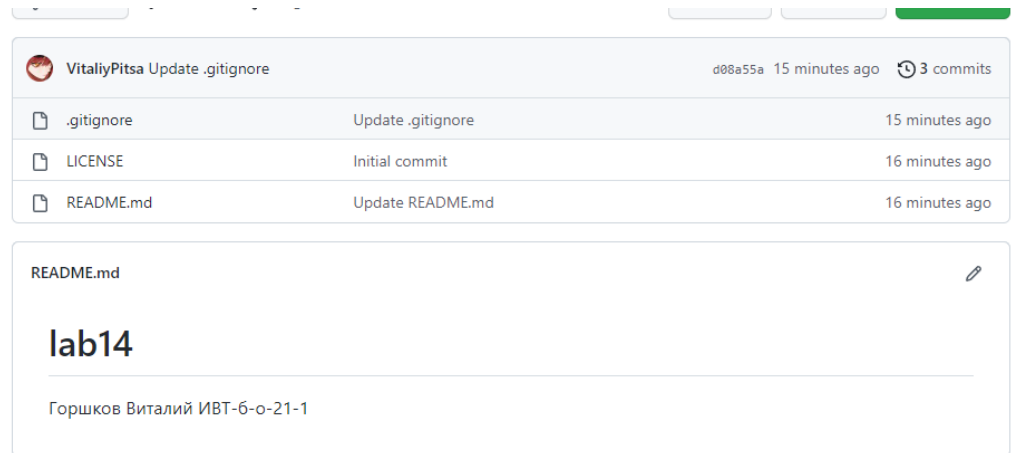


Рисунок 1. Создание репозитория

2. Выполнил клонирование созданного репозитория.

```
C:\Users\Dina\Desktop\Вуз>git clone https://github.com/DinaDichenko/Lab2_19.git
Cloning into 'Lab2_19'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
C:\Users\Dina\Desktop\Вуз>
```

Рисунок 2. Клонирование репозитория

### 3. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.

```
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839
.idea/
.idea
*.json
# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf
/Lab19

# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml

# Gradle
```

Рисунок 3. Изменение файла .gitignore

4. Проработал примеры лабораторной работы. Создала для них отдельные модули языка Python. Зафиксировала изменения в репозитории.

```
Desktop\Быз\Lab2_19\prog\prim1.py'  
Counter({' .py': 1})  
PS C:\Users\Dina\Desktop\Быз\Lab2_19\prog> █
```

Рисунок 5. Результат работы примера 1

```
+ C:\Users\Dina\Desktop\Быз\Lab2_19\prog  
+ prim1.py  
+ prim2.py  
PS C:\Users\Dina\Desktop\Быз\Lab2_19\prog> █
```

Рисунок 6. Результат работы примера 2

```
honFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher"  
C:\Users\Dina\Desktop\Быз\Lab2_19\prog\test001.txt  
PS C:\Users\Dina\Desktop\Быз\Lab2_19\prog> █
```

Рисунок 7. Результат работы примера 3

5. Выполнила индивидуальное задания.

#### 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 .

```
C:\Users\Dina\Desktop\Быз\Lab2_19\prog>python ind1.py display i.json  
+-----+-----+-----+  
|      No      |      Название      |      Время      |  
+-----+-----+-----+  
|      11      |      One      |      11:11:00      |  
+-----+-----+-----+
```

Рисунок 8. Результат работы программы

## 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.

```
(venv) C:\Users\Виталий\PycharmProjects\pythonProject9>py indiviudal2.py size primer2.py  
>> primer2.py - 356 B
```

Рисунок 10. 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..