

**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

Диченко Дина Алексеевна

« » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

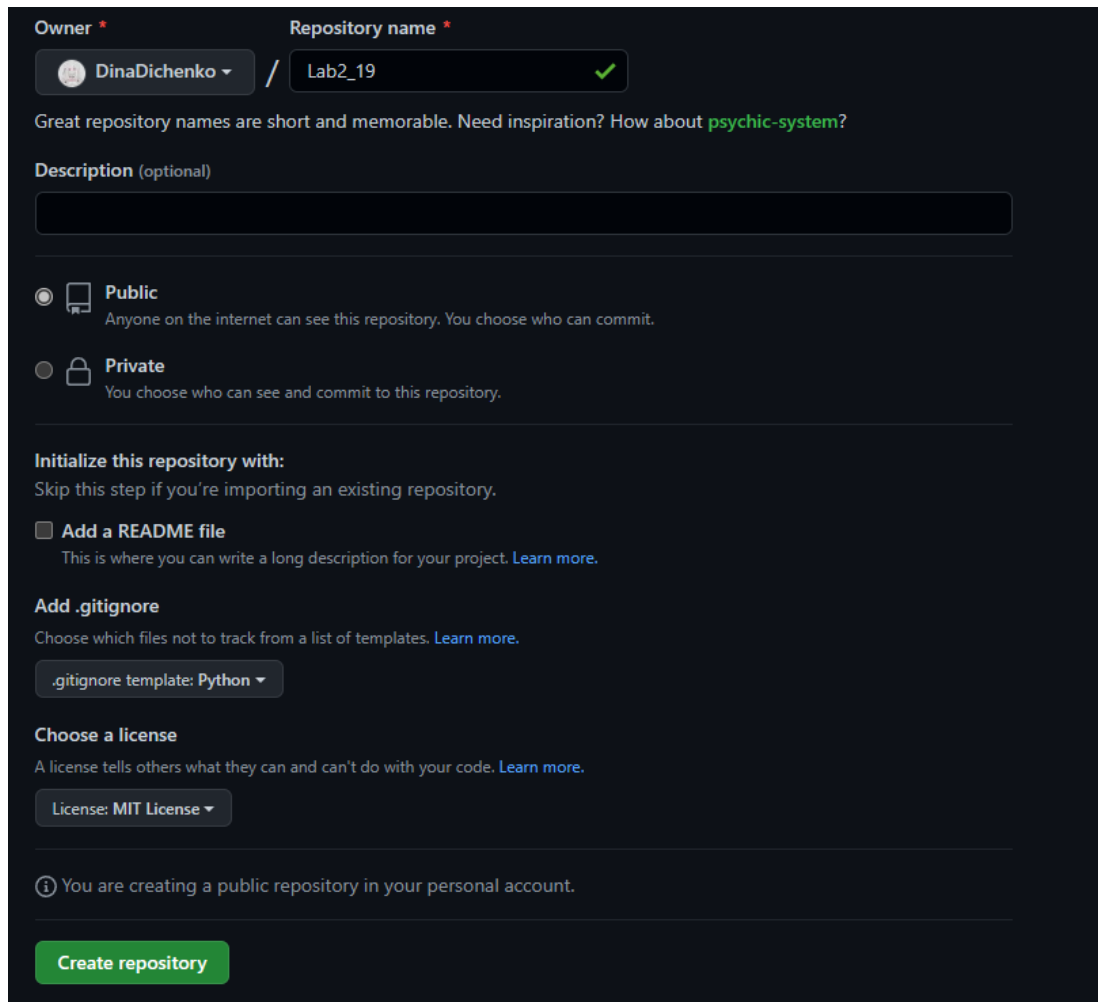
Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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

Практическая часть:

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



Owner * / Repository name *

DinaDichenko / Lab2_19 ✓

Great repository names are short and memorable. Need inspiration? How about [psychic-system?](#)

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 ▼

i You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Выполнила 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. Дополнила 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. Организовала свой репозиторий в соответствии с моделью ветвления git-flow.

```
C:\Users\Dina\Desktop\Вуз\Lab2_19>git branch develop
C:\Users\Dina\Desktop\Вуз\Lab2_19>git checkout develop
Switched to branch 'develop'

C:\Users\Dina\Desktop\Вуз\Lab2_19>git flow init

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

Which branch should be used for integration of the "next release"?
  - develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] fea
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] sup
Version tag prefix? [] ver
Hooks and filters directory? [C:/Users/Dina/Desktop/Вуз/Lab2_19/.git/hooks] hook
```

Рисунок 4. Организация репозитория в соответствии с git-flow

5. Проработала примеры лабораторной работы. Создала для них отдельные модули языка Python. Зафиксировала изменения в репозитории.

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

Рисунок 5. Результат работы примера 1

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

Рисунок 6. Результат работы примера 2

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

Рисунок 7. Результат работы примера 3

6. 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\Byz\Lab2_19\prog>python ind1.py display i.json  
+-----+-----+-----+  
|      No      |      Название      |      Время      |  
+-----+-----+-----+  
|           11 |      One           |      11:11:00   |  
+-----+-----+-----+
```

Рисунок 8. Результат работы программы

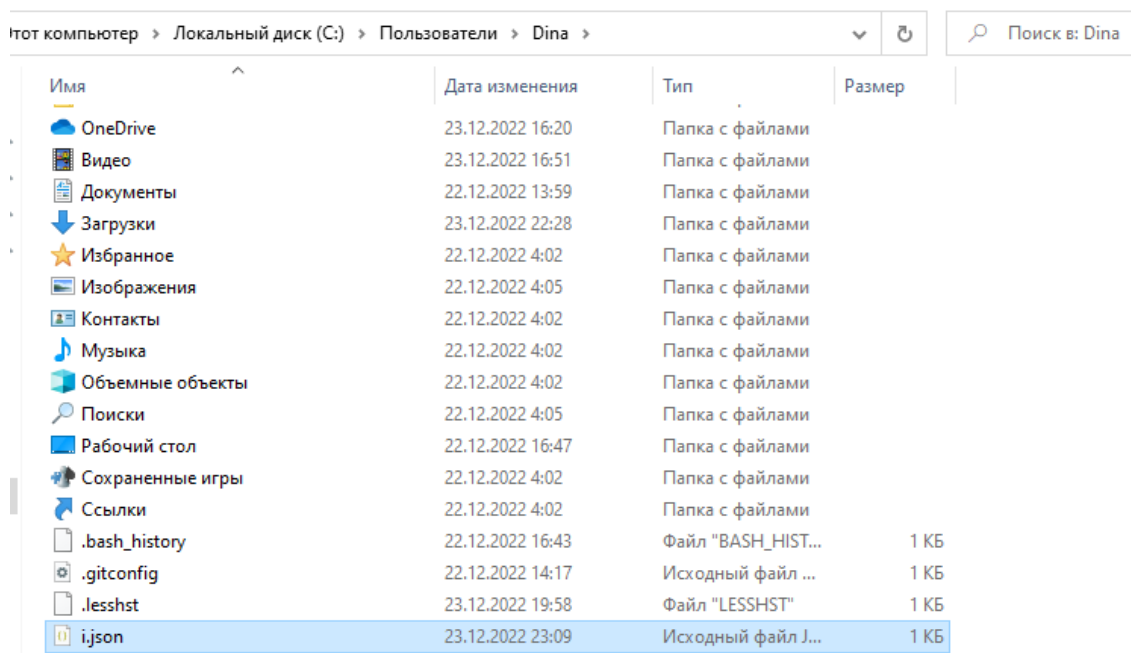


Рисунок 9. JSON-файл в домашнем каталоге

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.

```
C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py
Введите "help" для вывода списка команд

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py 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 дириктории
rmdir - удаление дириктории
mk - создание файла
rm - удаление файла

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py all
C:\Users\Dina\Desktop\Вуз\Lab2_19\prog

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py files
>> ind.json
>> ind1.py
>> ind2.py
>> load.json
>> prim1.py
>> prim2.py
>> prim3.py

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py size ind.json
>> ind.json - 88 B

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py save ind2.json

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py mkdir test

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py mk test.txt

C:\Users\Dina\Desktop\Вуз\Lab2_19\prog>py ind2.py files
>> ind.json
>> ind1.py
>> ind2.json
>> ind2.py
>> load.json
>> prim1.py
>> prim2.py
>> prim3.py
>> test
>> test.txt
```

Рисунок 10. Результат работы программы

```

C:\Users\Dina\Desktop\Быз\Lab2_19\prog>py ind2.py rmdir test
C:\Users\Dina\Desktop\Быз\Lab2_19\prog>py ind2.py rm test.json
Traceback (most recent call last):
  File "ind2.py", line 131, in <module>
    main()
  File "ind2.py", line 108, in main
    directory_path.unlink()
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.7_3.7.2544.0_x64__qbz
", line 1309, in unlink
    self._accessor.unlink(self)
FileNotFoundError: [WinError 2] Не удается найти указанный файл: 'C:\\Users\\Dina\\Desktop\\
.json'
C:\Users\Dina\Desktop\Быз\Lab2_19\prog>py ind2.py rm test.txt
C:\Users\Dina\Desktop\Быз\Lab2_19\prog>py ind2.py files
>> ind.json
>> ind1.py
>> ind2.json
>> ind2.py
>> load.json
>> prim1.py
>> prim2.py
>> prim3.py

```

Рисунок 11. 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 `colorama` языка программирования Python версии 3.x..