

**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 №2.15

Дисциплина: «Основы программной инженерии»

Тема: «Работа с файлами в языке Python»

Выполнила:

студентка 2 курса

группы Пиж-б-о-21-1

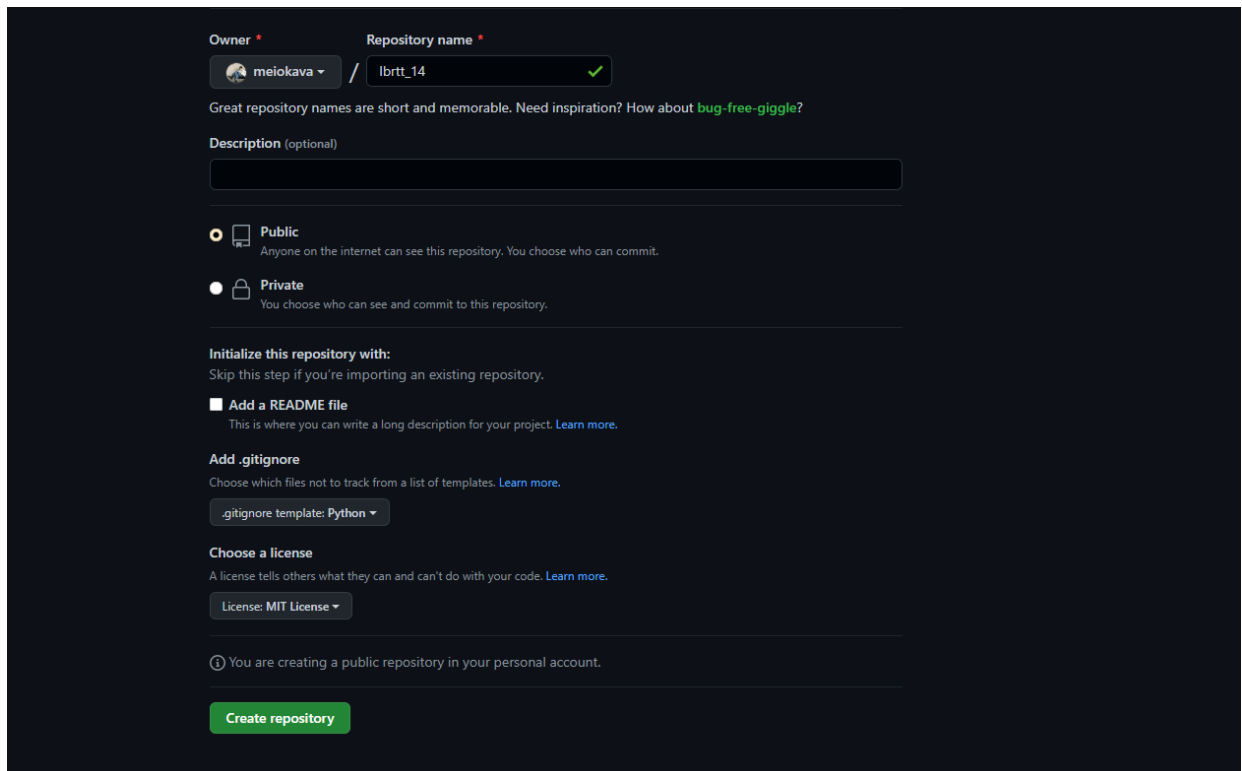
Джолдошова Мээрим

Бекболотовна

Ставрополь 2022

Цель работы: приобретение навыков по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучение основных методов модуля os для работы с файловой системой, получение аргументов командной строки.

1. Был создан репозиторий в Github в который были добавлены правила gitignore для работы IDE PyCharm, была выбрана лицензия MIT, сам репозиторий был клонирован на локальный сервер и был организован в соответствии с моделью ветвления git-flow.



Owner * meiokava / Repository name * lbrtt_14 ✓

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

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

① You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

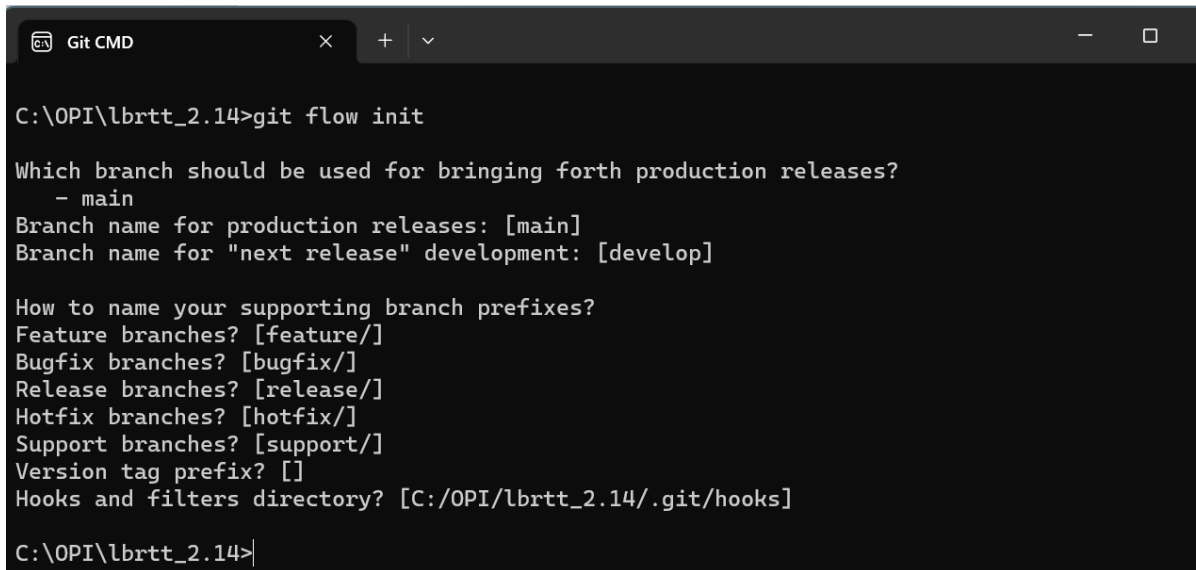
```
C:\Users\mвидео>cd/d C:\OPI

C:\OPI>git clone https://github.com/meiokava/lbrtt_2.14.git
Cloning into 'lbrtt_2.14'...
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:\OPI>cd/d C:\OPI\lbrtt_2.14

C:\OPI\lbrtt_2.14>
```

Рисунок 2 – Клонирование репозитория



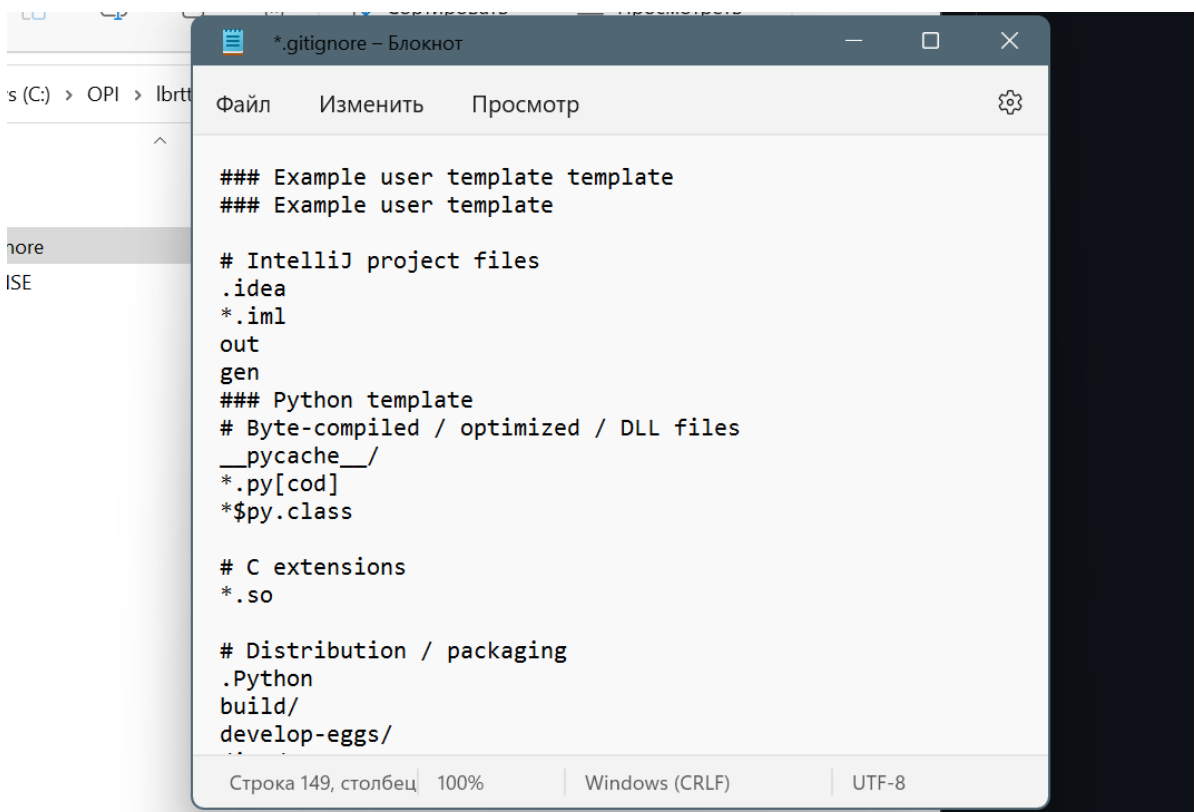
```
C:\OPI\lbrtt_2.14>git flow init

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:/OPI/lbrtt_2.14/.git/hooks]

C:\OPI\lbrtt_2.14>
```

Рисунок 3 – Организация модели ветвления git-flow



```
*.gitignore – Блокнот
Файл  Изменить  Просмотр

### Example user template template
### Example user template

# IntelliJ project files
.idea
*.iml
out
gen
### Python template
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/

Строка 149, столбец 100%  Windows (CRLF)  UTF-8
```

Рисунок 4 – Дополнение файла .gitignore

```
C:\OPI>python -m venv env
```

```
C:\OPI>.\env\Scripts\activate  
  
(env) C:\OPI>|
```

Рисунок 5 – Установка и активация виртуального окружения

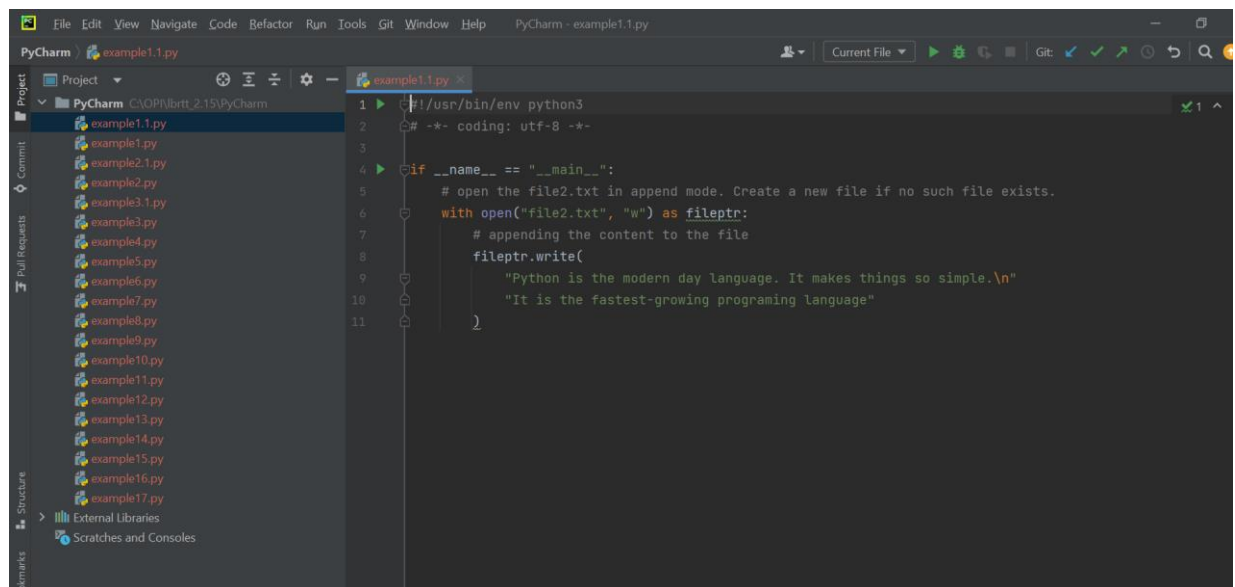


Рисунок 6 – Проработанные примеры лабораторной работы

Индивидуальное задание

Вариант 19

Задание 1

Составить программу с использованием списков и словарей для решения задачи. Номер варианта определяется по согласованию с преподавателем. Исходный файл, из которого выполняется чтение, необходимо также добавить в репозиторий, каждое предложение в файле должно находиться на отдельной строке.

Написать программу, которая считывает текст из файла и выводит на экран предложения, содержащие минимальное количество знаков пунктуации.

Код программы:

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

def punk(f):
    if ((f == ",") or (f == ".") or (f == "!") or (f == ";") or
        (f == ":") or (f == "?")):
        return 1

if __name__ == "__main__":
    file = open("my_file.txt", encoding='UTF')
    f = file.readlines()
    c = 0
    h = 0
    m_punk = 0
    for sim in f:
        c = 0
        for i in sim:
            if punk(i) == 1:
                c += 1
        if c < h:
            h = c
            m_punk = sim
    for sim in f:
        c = 0
        for i in sim:
            if punk(i) == 1:
                c += 1
        if c == h:
            print(sim)
```

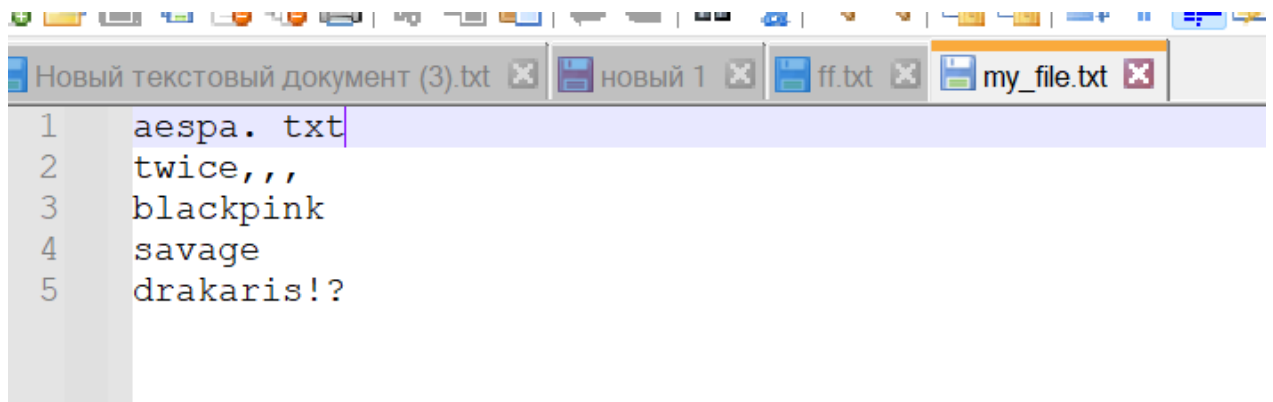


Рисунок 7 – Текстовый файл



Рисунок 8 – Результат работы программы

Задание 2

Составить программу с использованием текстовых файлов. Номер вариант необходимо получить у преподавателя.

Напишите программу, которая будет считывать содержимое файла, добавлять к считанным строкам порядковый номер и сохранять их в таком виде в новом файле. Имя исходного файла необходимо запросить у пользователя, так же, как и имя целевого файла. Каждая строка в созданном файле должна начинаться с ее номера, двоеточия и пробела, после чего должен идти текст строки из исходного файла.

Код программы:

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

if __name__ == "__main__":
    rd = str(input("source - "))
    wrt = str(input("where to write - "))
    with open(rd, "r", encoding="utf-8") as file:
        tw = open(wrt, "a")
        s_ind = 1
        q = ""
        for i in file:
            q = str(s_ind) + "; " + i
            s_ind += 1
            tw.write(q+"\n")
```

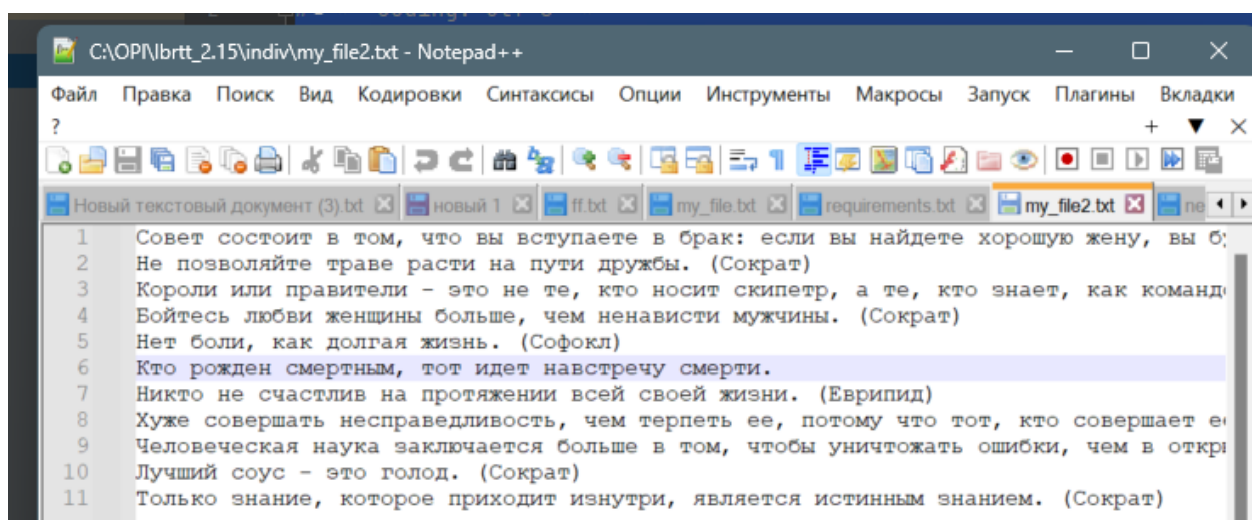


Рисунок 9 – Исходный текстовый файл

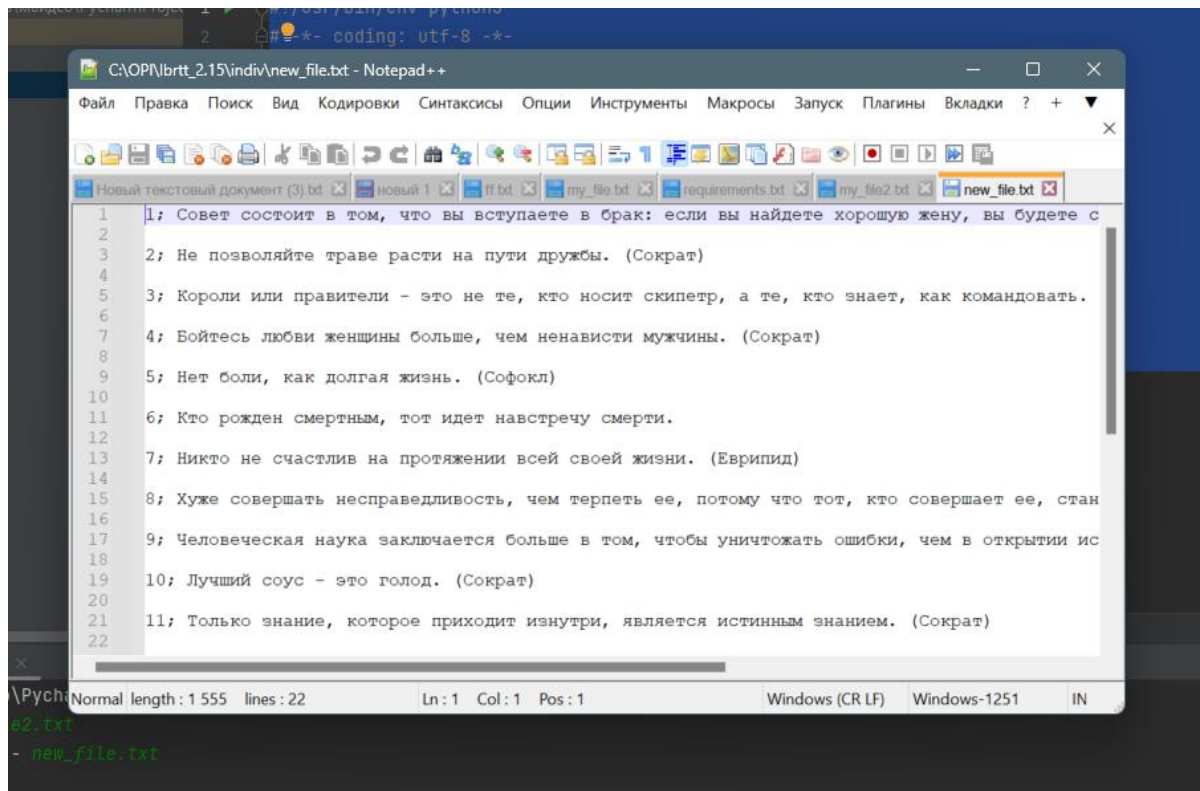


Рисунок 10 – Новый файл после работы программы

Задание 3:

Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля `os`. Приведите решение этой задачи.

Задание: создать новый каталог, затем создать в нем текстовый файл с именем этого каталога и переименовать его при помощи пользователя и после удалить.

Код программы:

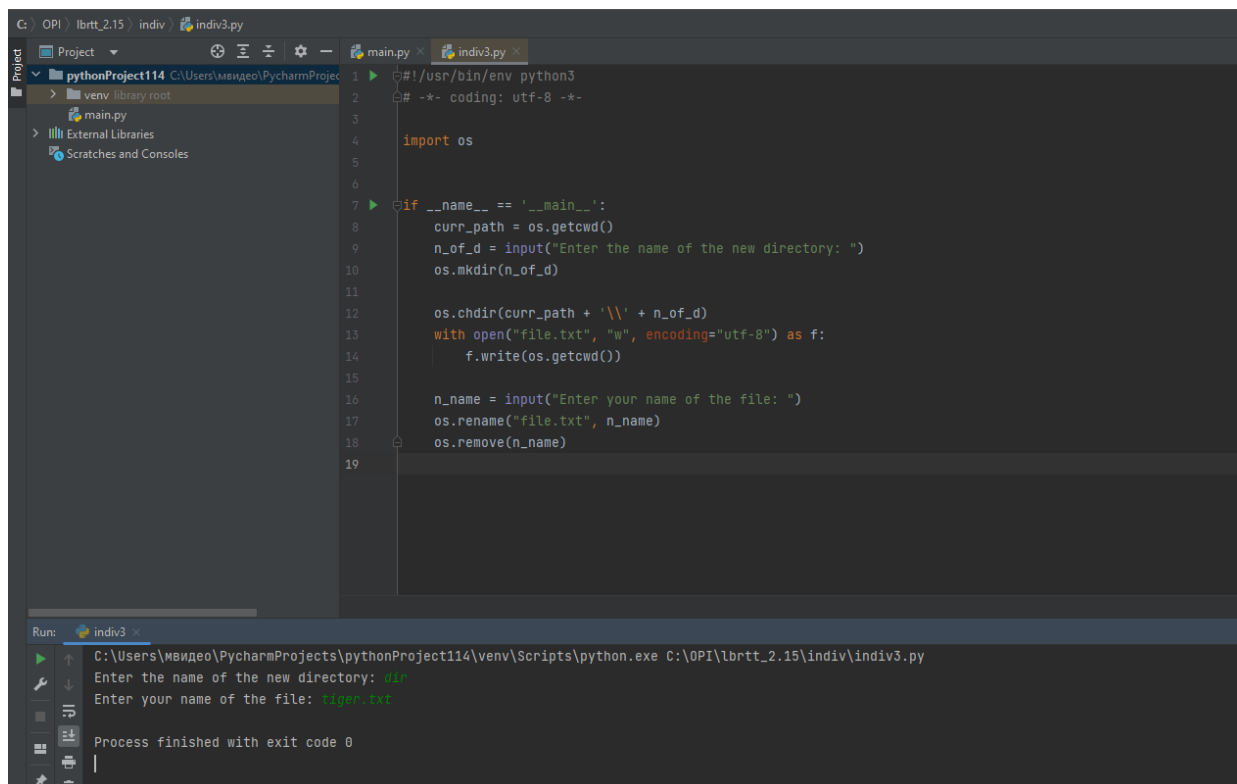
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os

if __name__ == '__main__':
    curr_path = os.getcwd()
    n_of_d = input("Enter the name of the new directory: ")
    os.mkdir(n_of_d)

    os.chdir(curr_path + '\\' + n_of_d)
    with open("file.txt", "w", encoding="utf-8") as f:
        f.write(os.getcwd())

    n_name = input("Enter your name of the file: ")
    os.rename("file.txt", n_name)
    os.remove(n_name)
```



Имя	Дата изменения	Тип	Рс
ddrrr	02.03.2023 16:53	Папка с файлами	
indiv1.py	02.03.2023 15:38	Python File	
indiv2.py	02.03.2023 16:05	Python File	
indiv3.py	02.03.2023 16:52	Python File	
my_file.txt	02.03.2023 15:39	Файл "TXT"	
my_file2.txt	02.03.2023 16:07	Файл "TXT"	
new_file.txt	02.03.2023 16:08	Файл "TXT"	

Рисунок 11 – Созданный каталог и удаленный в нем файл

Вопросы для защиты работы:

1. Как открыть файл в языке Python только для чтения?
2. Как открыть файл в языке Python только для записи?
3. Как прочитать данные из файла в языке Python?
4. Как записать данные в файл в языке Python?
5. Как закрыть файл в языке Python?
6. Изучите самостоятельно работу конструкции with ... as. Каково ее

назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?

7. Изучите самостоятельно документацию Python по работе с файлами. Какие помимо рассмотренных существуют методы записи/чтения информации из файла?

8. Какие существуют, помимо рассмотренных, функции модуля os для работы с файловой системой?