

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Работа с файлами в языке Python»

Отчет по лабораторной работе № 2.15
по дисциплине «Программирование на Python»

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«21» ноября 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

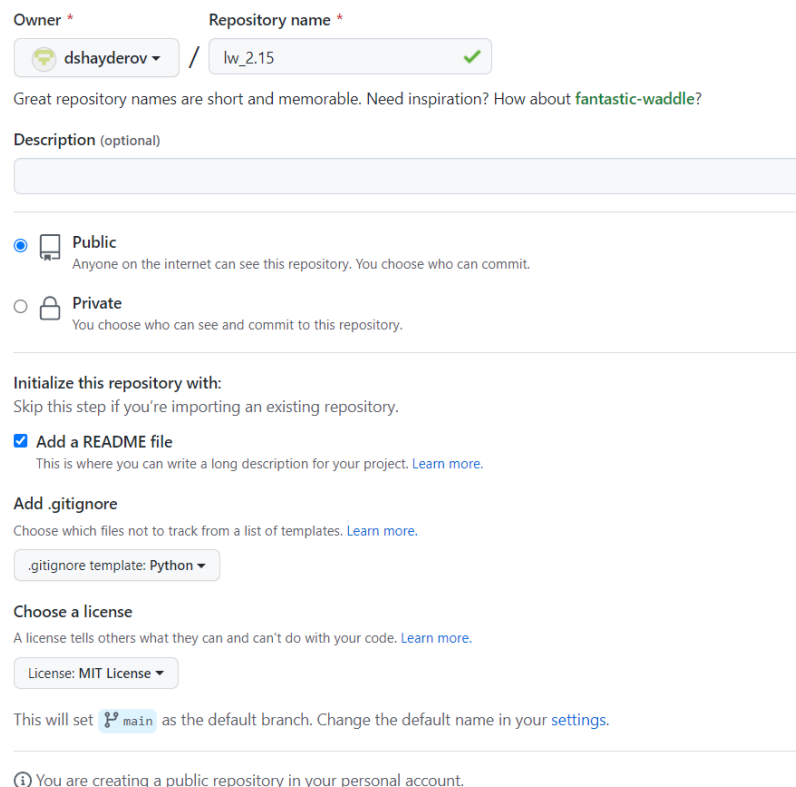
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022


Цель работы: приобретение навыков по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

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





Owner * Repository name *

 dshayderov / lw_2.15

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

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](#).


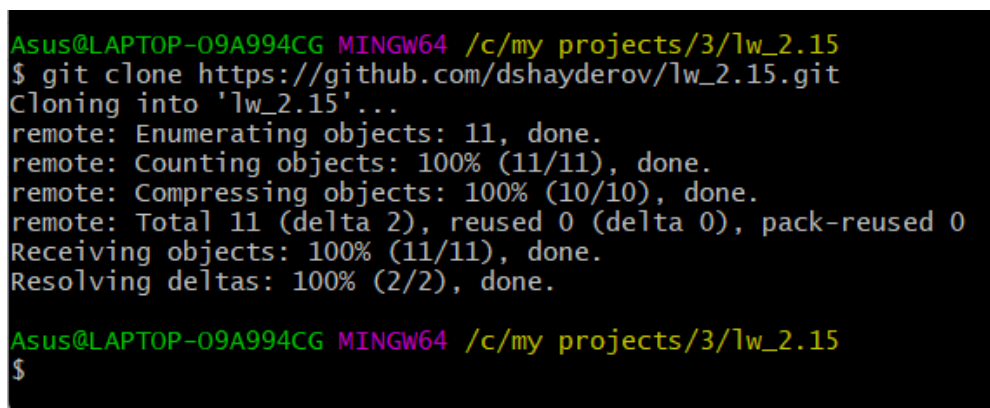
 You are creating a public repository in your personal account.

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.



```
Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3/lw_2.15
$ git clone https://github.com/dshayderov/lw_2.15.git
Cloning into 'lw_2.15'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (2/2), done.

Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3/lw_2.15
$
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```
Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3/lw_2.15 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

Asus@LAPTOP-09A994CG MINGW64 /c/my projects/3/lw_2.15 (develop)
$
```

Рисунок 3 - Ветвление по модели git-flow

4. Создайте проект PyCharm в папке репозитория.

Этот компьютер > OS (C:) > My projects > 3 > lw_2.15 >				
Имя	Дата изменения	Тип	Размер	
Project	21.11.2022 10:10	Папка с файлами		
.gitignore	21.11.2022 10:08	Файл "GITIGNORE"	2 КБ	
LICENSE	21.11.2022 10:08	Файл	2 КБ	
README.md	21.11.2022 10:08	Файл "MD"	1 КБ	

Рисунок 4 - Создание проекта

5. Проработать примеры лабораторной работы.

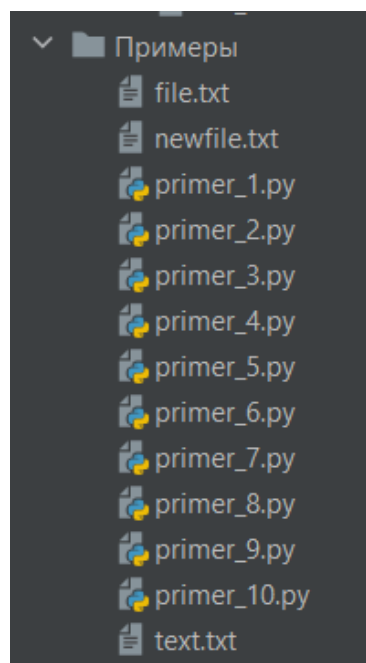
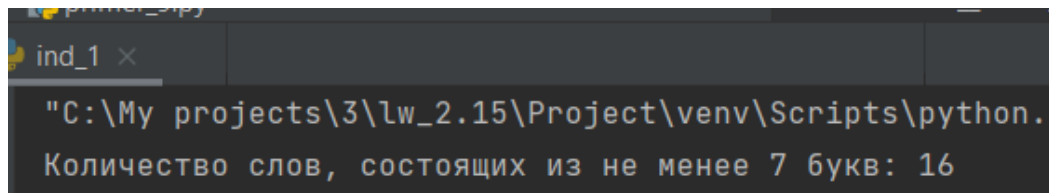


Рисунок 5 - Выполненные примеры

6. Выполнить индивидуальные задания.

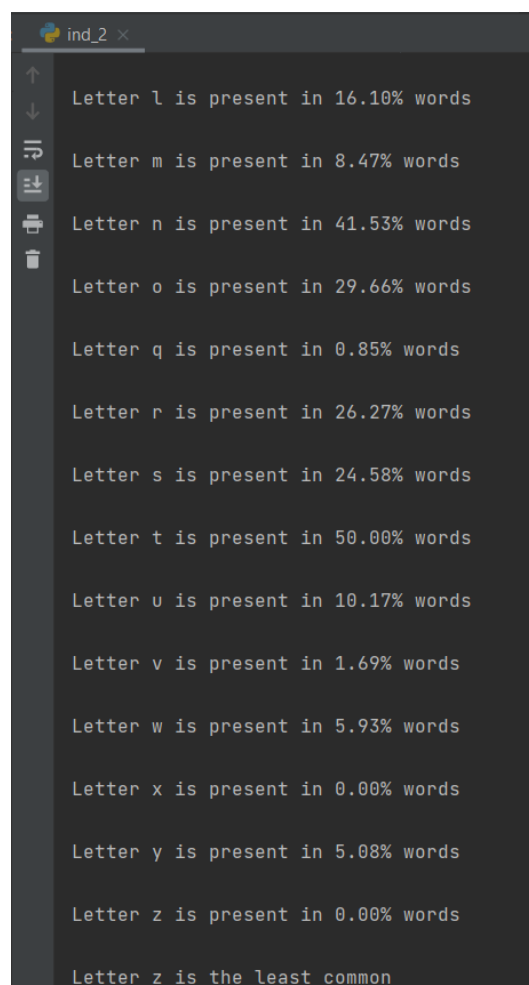
Индивидуальное задание 1 (Вариант 26/7). Написать программу, которая считывает текст из файла и определяет, сколько в нем слов, состоящих из не менее чем семи букв.



```
ind_1 x
"C:\My projects\3\lw_2.15\Project\venv\Scripts\python.
Количество слов, состоящих из не менее 7 букв: 16
```

Рисунок 6 - Результат выполнения индивидуального задания 1

Индивидуальное задание 2 (Вариант 26/11). Напишите программу, которая будет считывать список слов из файла и собирать статистику о том, в каком проценте слов используется каждая буква алфавита. Выведите результат для всех 26 букв английского алфавита и отдельно отметьте букву, которая встречалась в словах наиболее редко. В вашей программе должны игнорироваться знаки препинания и регистр символов.



```
ind_2 x
Letter l is present in 16.10% words
Letter m is present in 8.47% words
Letter n is present in 41.53% words
Letter o is present in 29.66% words
Letter q is present in 0.85% words
Letter r is present in 26.27% words
Letter s is present in 24.58% words
Letter t is present in 50.00% words
Letter u is present in 10.17% words
Letter v is present in 1.69% words
Letter w is present in 5.93% words
Letter x is present in 0.00% words
Letter y is present in 5.08% words
Letter z is present in 0.00% words
Letter z is the least common
```

Рисунок 7 - Результат выполнения индивидуального задания 2

Контрольные вопросы:

1. Как открыть файл в языке Python только для чтения?

`file object = open(<file-name>, <access-mode>, <buffering>)`

`access-mode = r` - открывает файл в режиме только для чтения.

Указатель файла существует в начале.

Файл по умолчанию открывается в этом режиме, если не передан режим доступа.

2. Как открыть файл в языке Python только для записи?

`file object = open(<file-name>, <access-mode>, <buffering>)`

`access-mode = w` - только для записи. Он перезаписывает файл, если он существовал ранее, или создает новый, если файл с таким именем не существует. Указатель имеется в начале файла.

3. Как прочитать данные из файла в языке Python?

Чтобы прочитать файл с помощью сценария Python, Python предоставляет метод `read()`.

Метод `read()` считывает строку из файла. Он может читать данные как в текстовом, так и в двоичном формате.

Python упрощает чтение файла построчно с помощью метода `readline()`. Метод `readline()` читает строки файла с самого начала, т. е. если мы используем его два раза, мы можем получить первые две строки файла.

4. Как записать данные в файл в языке Python?

Чтобы записать текст в файл, нам нужно открыть файл с помощью метода `open` с одним из следующих режимов доступа.

`'w'`: он перезапишет файл, если какой-либо файл существует. Указатель файла находится в начале файла.

`'a'`: добавит существующий файл. Указатель файла находится в конце файла. Он создает новый файл, если файл не существует.

5. Как закрыть файл в языке Python?

После того, как все операции будут выполнены с файлом, мы должны закрыть его с помощью нашего скрипта Python, используя метод `close()`.

Любая незаписанная информация уничтожается после вызова метода `close()` для файлового объекта.

6. Изучите самостоятельно работу конструкции `with ... as`. Каково ее назначение в языке Python? Где она может быть использована еще, помимо работы с файлами?

Конструкция `with ... as` используется для оборачивания выполнения блока инструкций менеджером контекста. Используется в сценарии, когда пара операторов должна выполняться с блоком кода между ними.

Преимущество использования оператора `with` заключается в том, что он обеспечивает гарантию закрытия файла независимо от того, как закрывается вложенный блок.

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

Функция `write()`

Функция `write()` используется для записи в файлы Python, открытые в режиме записи.

С помощью метода `writelines()` можно записать в файл итерируемую последовательность.

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

Функцию `os.replace()` можно использовать для перемещения файлов или каталогов.

Функция `os.listdir()` возвращает список, который содержит имена файлов в папке.

`os.walk()` — это генератор дерева каталогов. Он будет перебирать все переданные составляющие.

Метод `os.path.join()` был использован для объединения текущего пути с именем файла/папки.

Для получения информации о файле в ОС используется функция `os.stat()`, которая выполняет системный вызов `stat()` по выбранному пути.

Вывод: были приобретены навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x.