

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе № 2
«Работа с файлами в языке Python»**

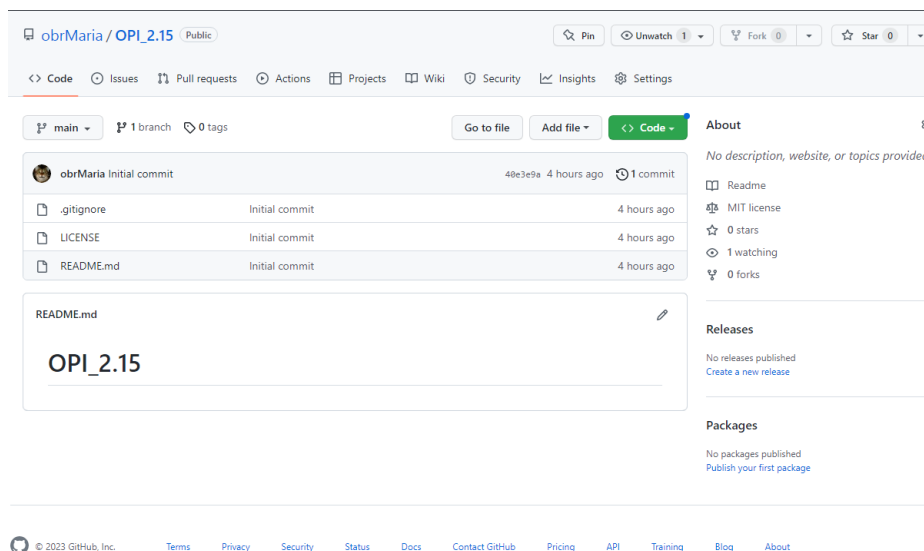
по дисциплине «Основы программной инженерии»

Выполнила:
Образцова Мария Дмитриевна,
2 курс, группа ПИЖ-б-о-21-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2023 г.

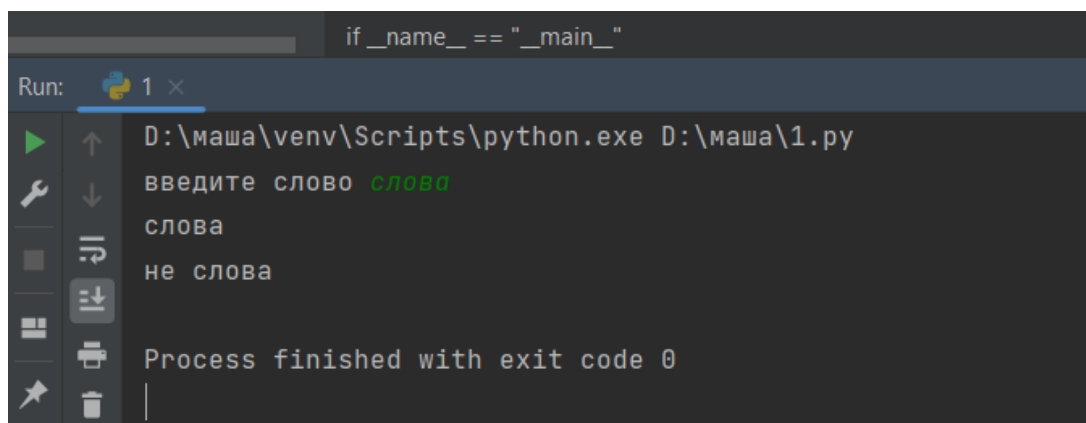
Методика и порядок выполнения работы

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python. Выполните клонирование созданного репозитория. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.



2. Создайте проект PyCharm в папке репозитория.

Проработать примеры лабораторной работы. Выполнить индивидуальные задания. Самостоятельно подберите или придумайте задачу для работы с изученными функциями модуля os. Приведите решение этой задачи.



The screenshot shows a code editor with a file explorer on the left and a console at the bottom. The file explorer shows a directory structure: 'маша' (D:\маша) containing 'venv' (library root) and 'дир_маша'. Inside 'дир_маша' are files '1.py', '1.txt', '2.txt', '3.py', and 'my_file.txt'. The code editor displays a Python script with the following content:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import os
5
6  if __name__ == '__main__':
7      # Создание директории и запоминание пути
8      new_dir = input("Введите имя директории: ")
9      os.mkdir(new_dir)
10     print(os.getcwd())
11
12     # создание нового файла
13     with open("newfile.txt", "x") as f:
14         print(f)
15         if f:
16             print("File created successfully")
17
18     # Переименование файла
19     os.rename("newfile.txt", "my_file.txt")
20
```

The console at the bottom shows the execution of the script:

```
Python 3 x
D:\маша\venv\Scripts\python.exe D:\маша\3.py
Введите имя директории: дир_маша
D:\маша
<_io.TextIOWrapper name='newfile.txt' mode='x' encoding='cp1251'>
File created successfully
Process finished with exit code 0
```

3. Зафиксируйте изменения в репозитории. Зафиксируйте изменения в репозитории. Выполните слияние ветки для разработки с веткой master/main.

Контрольные вопросы.

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

Чтобы открыть файл для чтения, мы используем режим `r`. Для чтения мы воспользуемся функцией `read(size)`, если параметр `size` не указан, функция вернет нам всю строку. `file = open("text.txt", 'r', encoding = 'utf-8')`.

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

В Python открытие файлов выполняется с помощью функции `open()`, которой передается два аргумента - имя файла и режим. Файл может быть открыт в режиме чтения, записи, добавления.

3. Как прочитать данные из файла в языке Python?

Чтение данных из файла осуществляется с помощью методов `read(размер)` и `readline()`. Метод `read(размер)` считывает из файла определенное количество символов, переданное в качестве аргумента.

4. Как записать данные в файл в языке Python?

Запись данных в файл. Записать данные в файл можно с помощью метода `write()`

5. Как закрыть файл в языке Python?

После того, как мы открыли файл, и выполнили все нужные операции, нам необходимо его закрыть. Для закрытия файла используется функция `close()`.

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. ... Если в конструкции `with - as`

было несколько выражений, то это эквивалентно нескольким вложенным конструкциям

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

Один из самых распространенных способов вывести данные в Python – это напечатать их в консоли. Если вы находитесь на этапе изучения языка, такой способ является основным для того, чтобы быстро просмотреть результат своей работы

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

- `os.chdir(path)` - смена текущей директории.
- `os.chmod (path, mode, *, dir_fd=None, follow_symlinks=True)` - смена прав доступа к объекту (`mode` - восьмеричное число).
- `os.chown (path, uid, gid, *, dir_fd=None, follow_symlinks=True)` - меняет `id` владельца и группы (Unix).
- `os.getcwd()` - текущая рабочая директория.
- `os.link (src, dst, *, src_dir_fd=None, dst_dir_fd=None, follow_symlinks=True)` - создаёт жёсткую ссылку.
- `os.listdir (path=".")` - список файлов и директорий в папке.
- `os.mkdir (path, mode=0o777, *, dir_fd=None)` - создаёт директорию.
- `OSError`, если директория существует.
- `os.makedirs (path, mode=0o777, exist_ok=False)` - создаёт директорию, создавая при этом промежуточные директории.
- `os.remove (path, *, dir_fd=None)` - удаляет путь к файлу.
- `os.rename (src, dst, *, src_dir_fd=None, dst_dir_fd=None)` - переименовывает файл или директорию из `src` в `dst`.

- `os.rename (old, new)` - переименовывает `old` в `new`, создавая промежуточные директории.
- `os.replace (src, dst, *, src_dir_fd=None, dst_dir_fd=None)` - переименовывает из `src` в `dst` с принудительной заменой.
- `os.rmdir (path, *, dir_fd=None)` - удаляет пустую директорию.
- `os.removedirs (path)` - удаляет директорию, затем пытается удалить родительские директории, и удаляет их рекурсивно, пока они пусты.
- `os.sync()` - записывает все данные на диск (Unix).
- `os.truncate (path, length)` - обрезает файл до длины `length`.
- `os.utime (path, times=None, *, ns=None, dir_fd=None,`
- `follow_symlinks=True)` - модификация времени последнего доступа и изменения файла. Либо `times` - кортеж (время доступа в секундах, время изменения в секундах), либо `ns` - кортеж (время доступа в наносекундах, время изменения в наносекундах). `os.walk (top, topdown=True, onerror = None, followlinks=False)` – генерация имён файлов в дереве каталогов, сверху вниз (если `topdown` равен `True`), либо снизу вверх (если `False`). Для каждого каталога функция `walk` возвращает кортеж (путь к каталогу, список каталогов, список файлов).