

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**  
**дисциплины «Анализ данных»**  
**Вариант 23**

Выполнил:  
Мотовилов Вадим Борисович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Информатика и вычислительная  
техника», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** работа с файлами в языке Python

**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.

Порядок выполнения работы:

1. Создал репозиторий и скопировал его

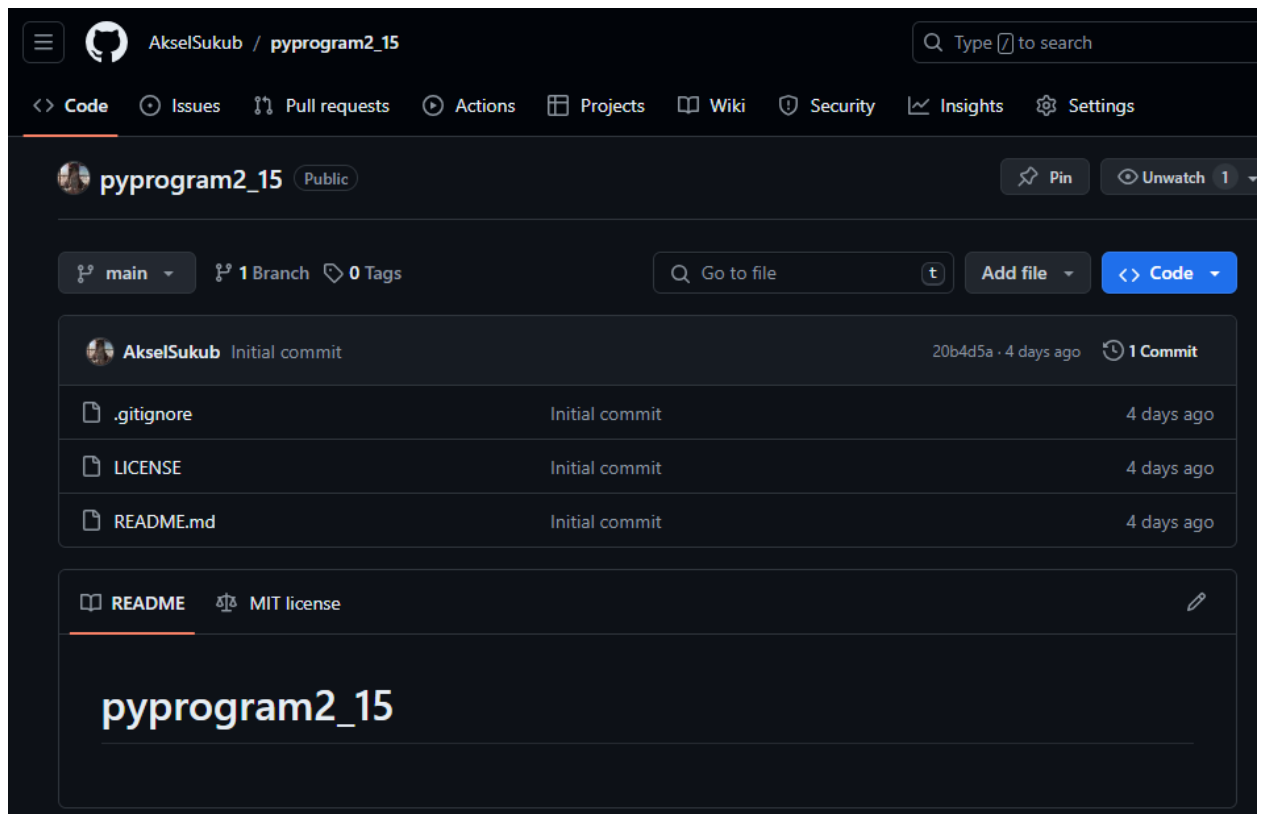


Рисунок 1. Созданный репозиторий

2. Изменил файл .gitignore и README.rm и добавил git flow

```
/site

# mypy
.mypy_cache/
.dmypy.json
dmypy.json

# Pyre type checker
.pyre/

# pytype static type analyzer
.pytype/

# Cython debug symbols
cython_debug/

# PyCharm
# JetBrains specific template is maintained in a separate JetBrains.gitignore that can
# be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore
# and can be added to the global gitignore or merged into this file. For a more nuclear
# option (not recommended) you can uncomment the following to ignore the entire idea folder.
.idea/
**/.DS_Store
.vscode
*.dSYM
```

Рисунок 2. Измененный файл .gitignore

### 3. Выполнил примеры и задания

4. Написать программу, которая считывает английский текст из файла и выводит на экран слова, начинающиеся с гласных букв.

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

def read_text_from_file(file_path):
    with open(file_path, 'r') as file:
        return file.read()

def extract_vowel_starting_words(text):
    vowels = 'aeiouAEIOU'
    words = text.split()
    vowel_words = [word for word in words if word and word[0] in vowels]
    return vowel_words

def main():
    file_path = 'C:/otkat/progr5/english_text.txt'
    english_text = read_text_from_file(file_path)
    vowel_starting_words = extract_vowel_starting_words(english_text)

    for word in vowel_starting_words:
        print(word)

if __name__ == "__main__":
    main()
```

```
progr5 > ≡ english_text.txt
1 I am Malenia blade of Mickela; And
2 Aksel
```

```
I
am
of
And
Aksel
```

Рисунок 3 . Выполнение 1 задания

8. Как вы знаете, в языке Python для создания комментариев в коде используется символ #. Комментарий начинается с этого символа и продолжается до конца строки – без возможности остановить его раньше. В данном упражнении вам предстоит написать программу, которая будет удалять все комментарии из исходного файла с кодом на языке Python. Пройдите по всем строкам в файле на предмет поиска символа #. Обнаружив его, программа должна удалить все содержимое, начиная с этого символа и до конца строки. Для простоты не будем рассматривать ситуации, когда знак решетки встречается в середине строки. Сохраните новое содержимое в созданном файле. Имена файла источника и файла назначения должны быть запрошены у пользователя. Удостоверьтесь в том, что программа корректно обрабатывает возможные ошибки при работе с обоими файлами.

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

def remove_comments(source_file, destination_file):
    try:
        with open(source_file, 'r') as file:
            lines = file.readlines()

        with open(destination_file, 'w') as file:
            for line in lines:
                idx = line.find('#')
                if idx != -1:
                    file.write(line[:idx] + '\n')
                else:
                    file.write(line)

        with open(destination_file, 'r') as file:
            modified_content = file.read()
            print("Modified content of the file: ", modified_content)
    except FileNotFoundError:
        print("File not found")
    except Exception as e:
        print("An error occurred:", e)

def main():
    source_file = input("Enter the source file name: ")
    destination_file = input("Enter the destination file name: ")
    remove_comments(source_file, destination_file)

if __name__ == "__main__":
    main()
```

```
progr5 > input.txt
1  #Это исходный файл с комментариями
2  print("Hello World!")
3  #Эта строка содержит комментарий
```

```
progr5 > output.txt
1
2  print("Hello World!")
3
```

Рисунок 4 . Выполнение 2 задания

### Ответы на контрольные вопросы:

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

Ответ: Чтобы открыть файл для чтения, мы используем режим r или rb.  
`file = open("text.txt", 'r').`

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

Ответ: Чтобы открыть файл для записи, мы используем режим w или wb. `file = open("text.txt", 'w').` Также можно использовать режим a или ab, чтобы не перезаписывать файл, а дозаписывать в него.

### 3. Как прочитать данные из файла в языке Python?

Ответ: 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. Также можно использовать цикл `for`: `for i in fileptr: print(i)`; также можно использовать метод `readline()`, который читает строки файла с самого начала, т. е. если мы используем его два раза, мы можем получить первые две строки файла. Python также предоставляет метод `readlines()`, который используется для чтения строк. Возвращает список строк до конца файла (EOF).

### 4. Как записать данные в файл в языке Python?

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

### 5. Как закрыть файл в языке Python?

Ответ: 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?

Ответ: Конструкция `with ... as` в языке Python предназначена для обеспечения управления ресурсами с автоматическим освобождением этих ресурсов после завершения блока кода. Она часто используется с объектами, которые поддерживают протокол менеджеров контекста.

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

Ответ: Метод `writelines()`: Данный метод принимает список строк в качестве аргумента и записывает каждую строку списка в файл. В Python также существует метод `readinto()`, который читает данные из файла и записывает их в предварительно выделенный буфер. Это может быть

полезно, если нужно работать с данными напрямую в буфере, минуя дополнительные копирования в памяти.

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

Ответ: `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.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.makedirs (path, mode=0o777, exist_ok=False)` - создаёт директорию, создавая при этом промежуточные директории. Вывод: в результате выполнения программы были приобретены навыки по работе с текстовыми файлами при написании программ с помощью языка программирования Python версии 3.x, изучены основные методы модуля `os` для работы с файловой системой, изучено получение аргументов командной строки.