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 №6
## дисциплины «Программирование на Python»
## 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.

Порядок выполнения работы:

1. Создал репозиторий и скопировал его

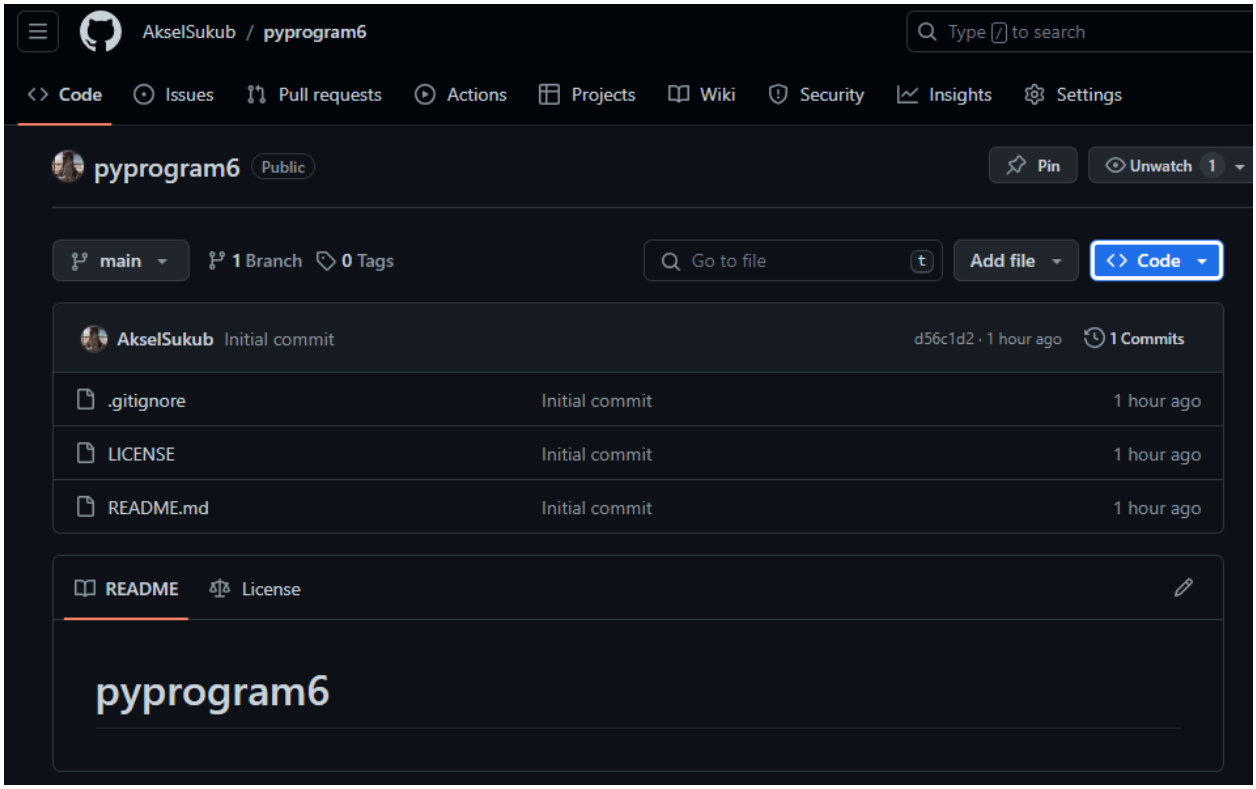


Рисунок 1. Созданный репозиторий

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

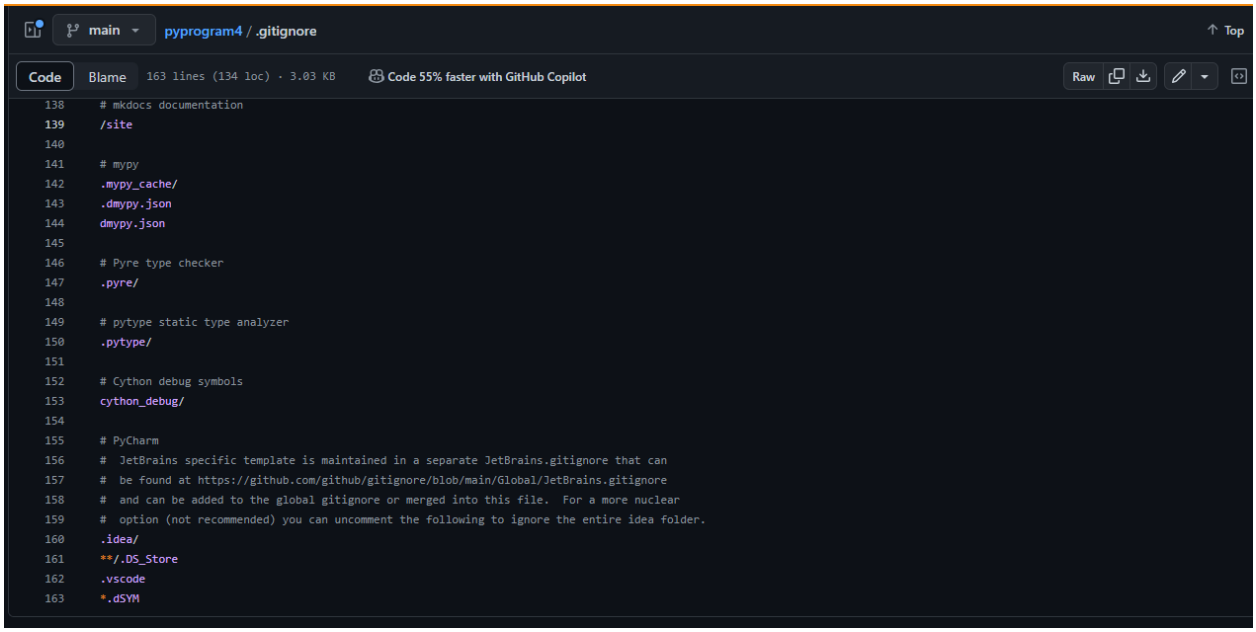


Рисунок 2. Измененный файл .gitignore

3. Выполнил задания

23. Дано предложение. В нем слова разделены одним или несколькими пробелами (символ «-» в предложении отсутствует). Определить количество слов в предложении. Рассмотреть два случая:

- начальные и конечные пробелы в предложении отсутствуют;
- начальные и конечные пробелы в предложении имеются.

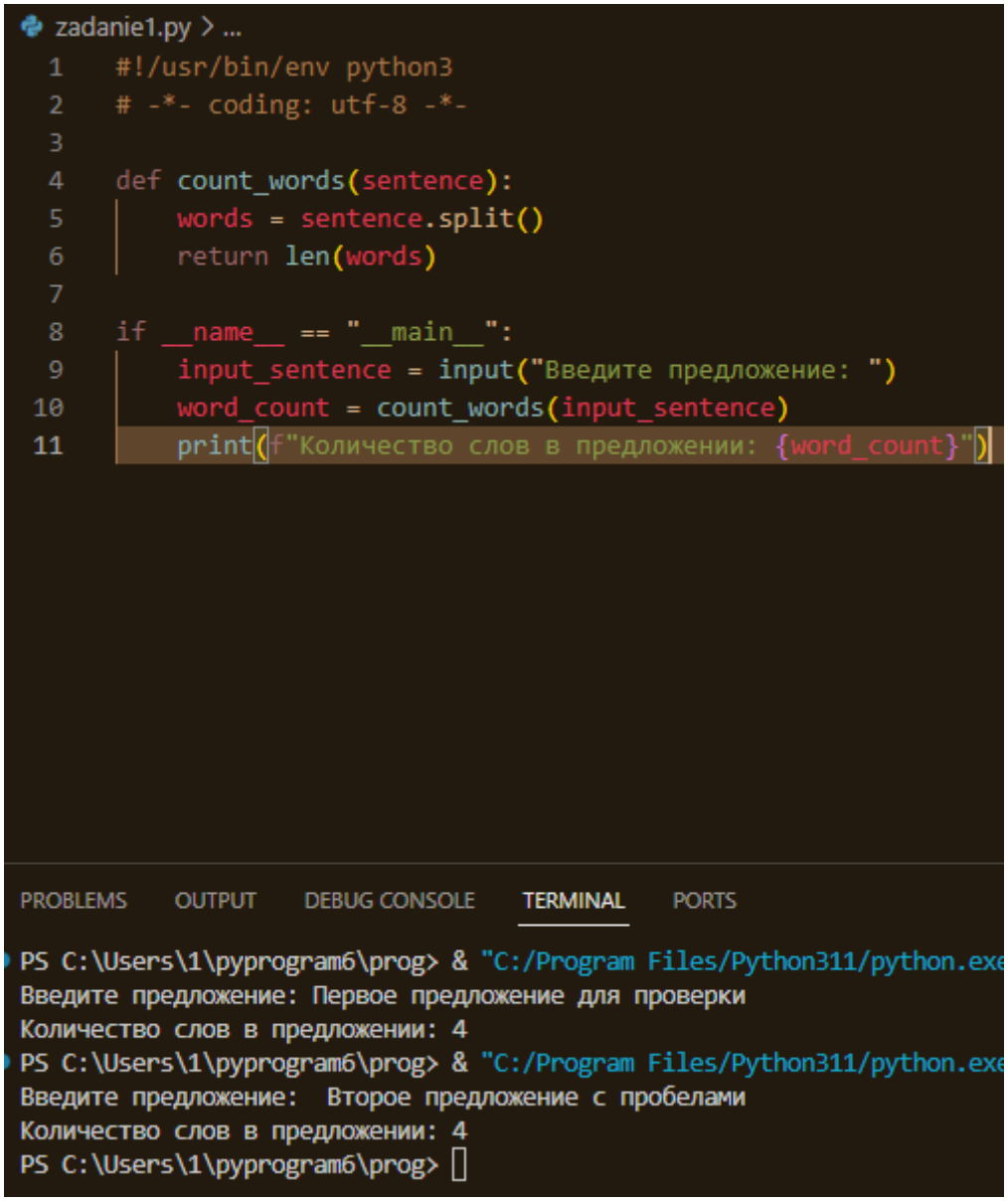


Рисунок 3 . Выполнение 1 задания

23. Дано слово. Поменять местами его m-ю и n-ю буквы.


Рисунок 4 . Выполнение 2 задания

Дано ошибочно написанное слово килбайот. Путем перемещения его букв получить слово килобайт.

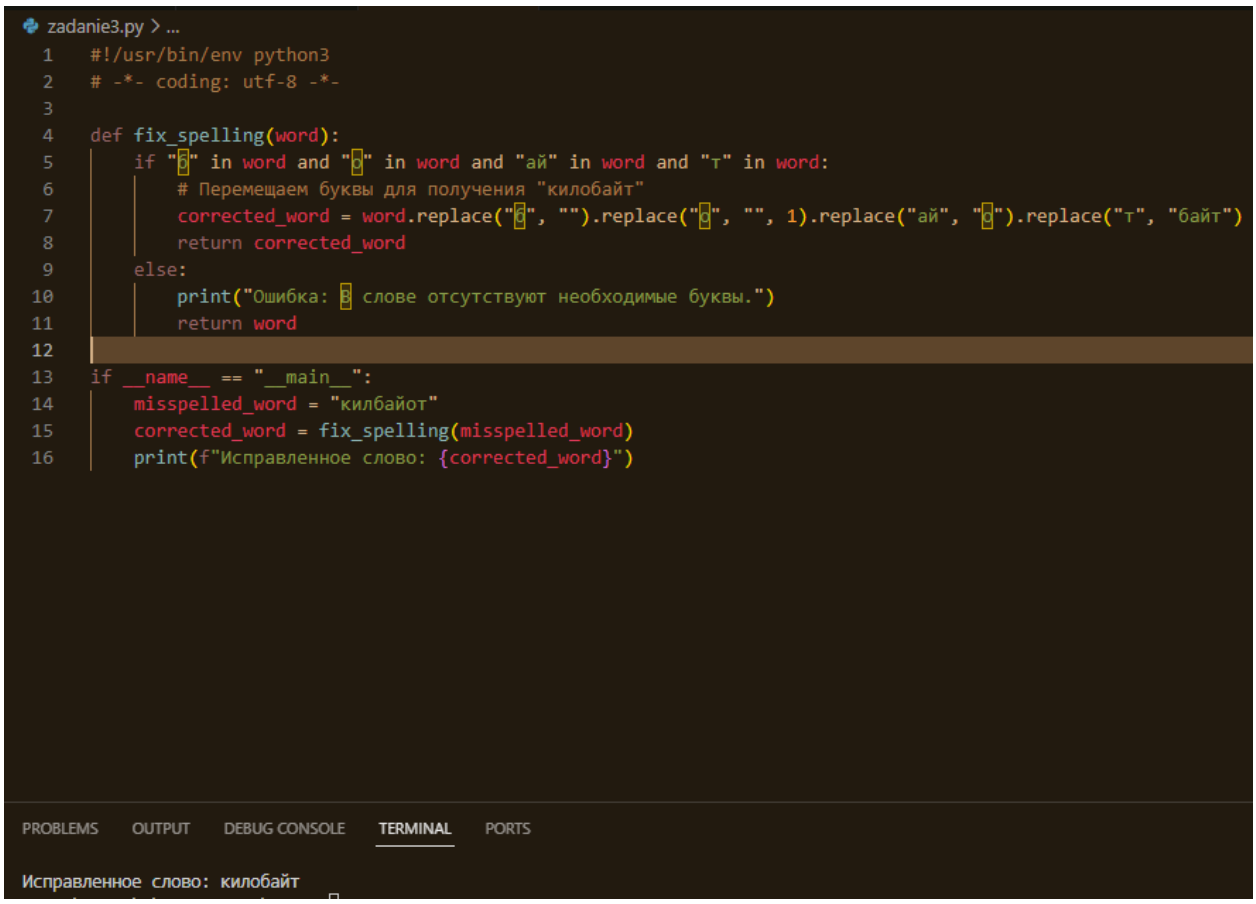

Рисунок 5 . Выполнение 3 задания

Дано предложение. Напечатать все слова, которые встречаются в нем по одному разу.

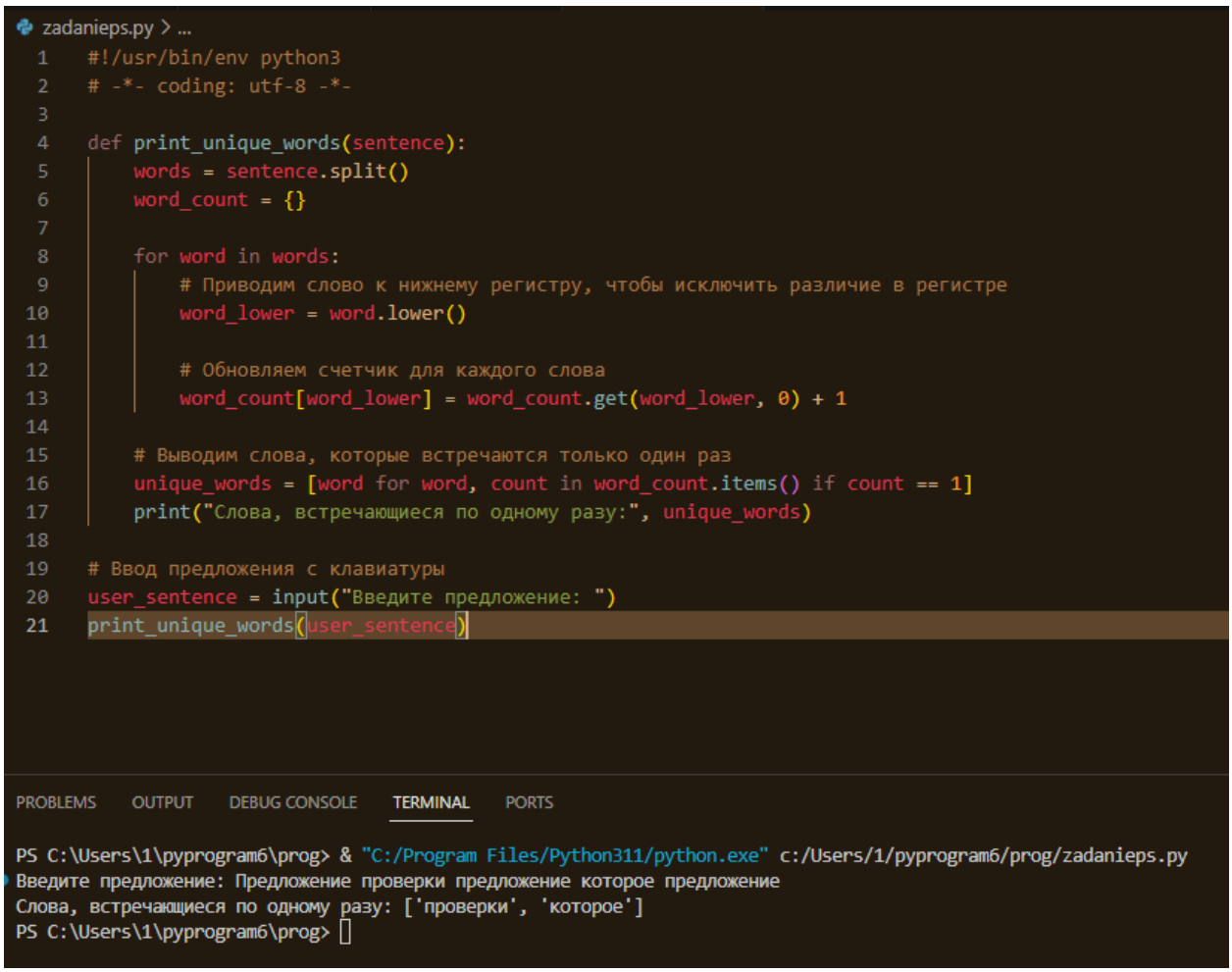


Рисунок 6 . Выполнение усложненного индивидуального задания

Ответы на контрольные вопросы:

1. **Что такое строки в языке Python?**

Строки в Python представляют собой последовательность символов и используются для хранения текстовой информации.

2. **Какие существуют способы задания строковых литералов в языке Python?**

Строковые литералы можно задавать с использованием одинарных (' '), двойных (" "), или тройных (''' ''' или """ """) кавычек.

3. **Какие операции и функции существуют для строк?**

Существуют различные операции, такие как конкатенация (+), умножение (*), методы для изменения регистра (upper(), lower()), получения длины (len()), извлечения подстрок (substring()), и многие другие.

4. **Как осуществляется индексирование строк?**

Строки индексируются, начиная с 0. Доступ к отдельным символам осуществляется с использованием квадратных скобок, например, my_string[0] вернет первый символ строки.

5. **Как осуществляется работа со срезами для строк?**

Срезы в Python создаются с использованием оператора :. Например, my_string[1:4] вернет подстроку, начиная со второго символа до четвёртого (не включая).

6. **Почему строки Python относятся к неизменяемому типу данных?**

Строки являются неизменяемыми, потому что после создания строки её нельзя изменить напрямую. Операции над строками создают новые строки.

7. **Как проверить то, что каждое слово в строке начинается с заглавной буквы?**

Можно воспользоваться методом istitle().

8. **Как проверить строку на вхождение в неё другой строки?**

Используйте оператор in: substring in my_string.

9. **Как найти индекс первого вхождения подстроки в строку?**

Используйте метод find().

10. **Как подсчитать количество символов в строке?**

Используйте функцию len().

11. **Как подсчитать то, сколько раз определённый символ встречается в строке?**

Воспользуйтесь методом count().

12. **Что такое f-строки и как ими пользоваться?**

F-строки представляют собой строковые литералы, начинающиеся с буквы 'f' или 'F'. Они позволяют встраивать значения переменных и выражений в строку.

13. **Как найти подстроку в заданной части строки?**

Используйте метод find() с указанием диапазона индексов.

14. **Как вставить содержимое переменной в строку, воспользовавшись методом format()?**

Пример: "Привет, {}!".format(name).

15. **Как узнать о том, что в строке содержатся только цифры?**

Используйте методы isdigit() или isnumeric().

16. **Как разделить строку по заданному символу?**

Используйте метод split().

17. **Как проверить строку на то, что она составлена только из строчных букв?**

Используйте метод islower().

18. **Как проверить то, что строка начинается со строчной буквы?**

Используйте метод str.startswith().

19. **Можно ли в Python прибавить целое число к строке?**

Нет, так как строки неизменяемы. Операции над строками создают новые строки.

20. **Как «перевернуть» строку?**

Используйте срезы: reversed_string = my_string[::-1].

21. **Как объединить список строк в одну строку, элементы которой разделены дефисами?**

Используйте метод join(): result = "-".join(my_list).

22. **Как привести всю строку к верхнему или нижнему регистру?**

Используйте методы upper() или lower().

23. **Как преобразовать первый и последний символы строки к верхнему регистру?**

Используйте методы capitalize() и title().

24. **Как проверить строку на то, что она составлена только из прописных букв?**

Используйте метод isupper().

25. **В какой ситуации вы воспользовались бы методом splitlines()?**

Метод splitlines() используется для разделения строки по символам новой строки (\n) и возвращения списка строк.

26. **Как в заданной строке заменить на что-либо все вхождения некоей подстроки?**

Используйте метод replace().

27. **Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?**

Используйте методы startswith() и endswith().

28. **Как узнать о том, что строка включает в себя только пробелы?**

Используйте метод isspace().