**Практическая работа №7. Функции. Область видимости переменных**

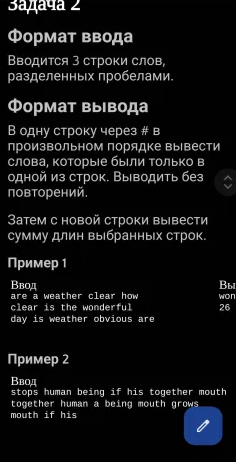
Задание

Выбрать задачи из файлов с вариантами в соответствии с таблицей.

Ссылка на варианты [ElenaKuzmina/PR7\_Function](https://github.com/ElenaKuzmina/PR7_Function)

Написать программный модуль, содержащий функции, реализующие решение задачи. Обязательно должна быть функция main(), содержащая вызовы остальных функций. В отчет включить для каждой задачи:

|  |  |  |
| --- | --- | --- |
| [Шумов Дмитрий](https://school.mosreg.ru/marks?school=2000002775450&group=2257826504610747540&student=2000000668822&tab=stats" \t "_blank) | Вариант 2, задача 2 | Вариант 1, задача 3 |

1. Шумов Д. ИСП.23.1А
2. 

· Начало программы.

· Получение входных строк.

· Создать пустой список strings.

· Запустить бесконечный цикл:

· Вывести сообщение и запросить у пользователя ввод строки через дефис или ввести "HELLO" для выхода.

· Если введенная строка содержит "HELLO", выйти из цикла.

· В противном случае добавить введенную строку в список strings.

· Фильтрация слов.

· Задать целевое слово target\_word как "computer".

· Создать множество target\_letter\_set, содержащие уникальные буквы из target\_word.

· Создать пустой список filtered\_strings для хранения отфильтрованных строк.

· Обработка каждой строки.

· Для каждого элемента string из списка strings с его индексом:

· Разделить строку по символу "-" и удалить лишние пробелы, чтобы получить список words.

· Создать множество unique\_words для хранения уникальных слов из списка words.

· Создать пустое множество valid\_words для хранения допустимых слов.

· Проверка каждого слова.

· Для каждого word в unique\_words:

· Вычислить количество общих букв между word и target\_letter\_set.

· Если количество общих букв меньше или равно 3, добавить word в множество valid\_words.

· Сортировка и форматирование слов.

· Отсортировать valid\_words в обратном порядке, сохраняя в переменной sorted\_words.

· Если индекс строки кратен 3, преобразовать все слова в sorted\_words в верхний регистр.

· Формирование результата.

· Объединить sorted\_words в строку с пробелами и добавить в список filtered\_strings.

· Вывод результата.

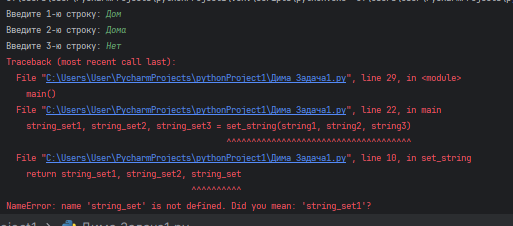
· Если список filtered\_strings не пуст, вывести все отфильтрованные строки. В противном случае вывести сообщение о том, что строки не были введены.

· Конец программы.

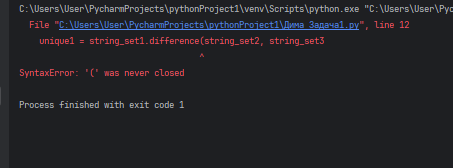
1. Программный код модуля

def input\_string():  
 string1 = input("Введите 1-ю строку: ")  
 string2 = input("Введите 2-ю строку: ")  
 string3 = input("Введите 3-ю строку: ")  
 return string1, string2, string3  
def set\_string(string1, string2, string3):  
 string\_set1 = set(string1.split())  
 string\_set2 = set(string2.split())  
 string\_set3 = set(string3.split())  
 return string\_set1, string\_set2, string\_set3  
def filtration(string\_set1, string\_set2, string\_set3):  
 unique1 = string\_set1.difference(string\_set2, string\_set3)  
 unique2 = string\_set2.difference(string\_set1, string\_set3)  
 unique3 = string\_set3.difference(string\_set1, string\_set2)  
 all\_strings = "#".join(unique1.union(unique2, unique3))  
 return all\_strings  
def string\_len(all\_strings):  
 lens = len(all\_strings)  
 return lens  
def main():  
 string1, string2, string3 = input\_string()  
 string\_set1, string\_set2, string\_set3 = set\_string(string1, string2, string3)  
 all\_strings = filtration(string\_set1, string\_set2, string\_set3)  
 lens = string\_len(all\_strings)  
 print("Уникальные слова:", all\_strings)  
 print("Длина строки:", lens)  
  
  
main()

1. Скриншоты ошибок с описанием ошибки и способом ее исправления

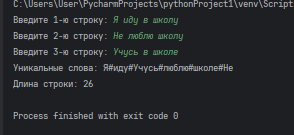


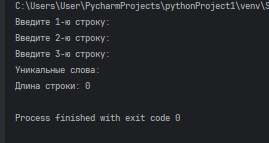
Ошибка в написании переменной



Ошибка в написании программного кода

1. Скриншоты тестирования программного модуля задачи





1. Ссылка на свой репозиторий github с программным модулем

<https://github.com/Dima-Shum/PR>

1. ИСП.23.1А
2. 
3. Описание переменных с указанием имени, типа, назначения, уровня видимости (в виде таблицы)

· ****Начало программы.****

· ****Получение строк от пользователя.****

· Запросить у пользователя ввод трех строк.

· Сохранить каждую строку в переменные string1, string2 и string3.

· ****Преобразование строк в множества.****

· Разделить каждую строку на слова и создать множество для каждой строки:

· string\_set1 – множество уникальных слов из string1.

· string\_set2 – множество уникальных слов из string2.

· string\_set3 – множество уникальных слов из string3.

· ****Поиск уникальных слов.****

· Найти слова, которые присутствуют только в одной из строк:

· unique1 – слова, которые есть только в string\_set1.

· unique2 – слова, которые есть только в string\_set2.

· unique3 – слова, которые есть только в string\_set3.

· ****Объединение уникальных слов.****

· Объединить уникальные слова из всех трех множеств в одну строку all\_strings, разделяя их символом "#".

· ****Подсчет длины результата.****

· Определить длину объединенной строки all\_strings и сохранить это значение в переменной lens.

· ****Вывод результатов.****

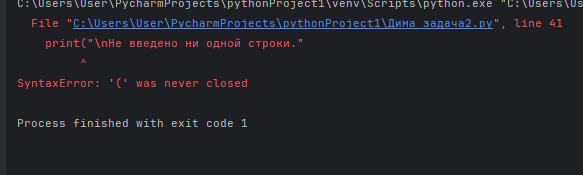
· Вывести на экран строку с уникальными словами и длину этой строки.

· ****Конец программы.****

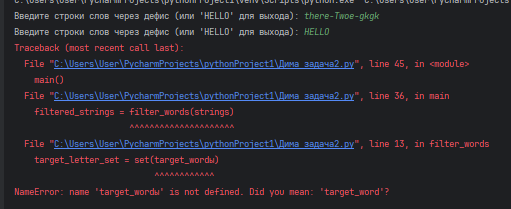
1. Программный код модуля

def get\_input\_strings():  
 strings = []  
 while True:  
 user\_input = input("Введите строки слов через дефис (или 'HELLO' для выхода): ")  
 if 'HELLO' in user\_input:  
 break  
 strings.append(user\_input)  
 return strings  
  
  
def filter\_words(strings):  
 target\_word = "computer"  
 target\_letter\_set = set(target\_word)  
 filtered\_strings = []  
 for index, string in enumerate(strings):  
 words = [word.strip() for word in string.split('-')]  
 unique\_words = set(words)  
 valid\_words = set()  
 for word in unique\_words:  
 common\_letters = len(set(word) & target\_letter\_set)  
 if common\_letters <= 3:  
 valid\_words.add(word)  
  
 sorted\_words = sorted(valid\_words, reverse=True)  
  
 if index % 3 == 0:  
 sorted\_words = [word.upper() for word in sorted\_words] # Поменяли на нужный уровень  
  
 filtered\_strings.append(" ".join(sorted\_words))  
 return filtered\_strings  
  
  
def main():  
 strings = get\_input\_strings()  
 if strings:  
 filtered\_strings = filter\_words(strings)  
 print("\nРезультат:")  
 for result in filtered\_strings:  
 print(result)  
 else:  
 print("\nНе введено ни одной строки.")  
  
  
# Запуск программы  
main()

1. Скриншоты ошибок с описанием ошибки и способом ее исправления

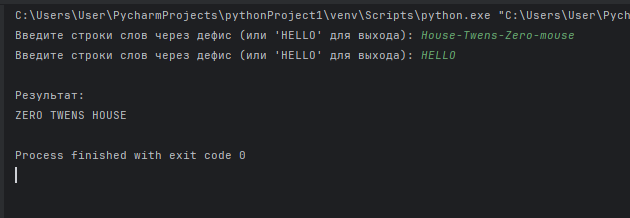


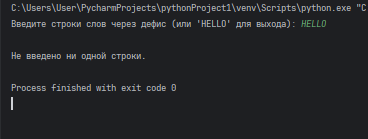
Ошибка в записи программы. - исправление: верно написать неправильную часть программы



Ошибка в записи переменной

1. Скриншоты тестирования программного модуля задачи





1. Ссылка на свой репозиторий github с программным модулем

<https://github.com/Dima-Shum/PR>