**ИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**«Сибирский государственный университет науки и технологий   
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Языки программирования

|  |
| --- |
| Строки |

Руководитель А.В. Проскурин

подпись, дата инициалы, фамилия

Обучающийся БПИ22-02, 221219040 К.В. Трифонов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2023 г.

# ЦЕЛЬ РАБОТЫ

Изучение основных средств языка программирования Python, необходимых для обработки произвольного текста.

**Задачи работы**: - изучение операций и методов обработки строк; - приобретение навыков обработки произвольного текста.

# порядок выполнения работы

1. Изучите теоретические сведения к данной лабораторной работе.

2. Ознакомьтесь с постановкой задачи (вариант задания вычисляется как ваш\_номер\_в\_списке\_группы % количество\_вариантов).

3. Составьте программу на языке Python и выполните ее отладку.

4. Подготовьте отчет по лабораторной работе. Отчет должен включать в себя:

− титульный лист;

− цель лабораторной работы;

− постановку задачи;

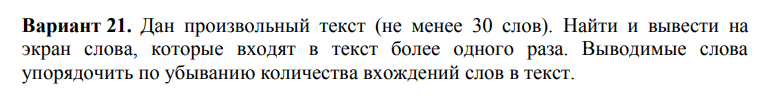
− текст программы с комментариями;

− результаты работы программы на тестовых исходных данных;

− краткие ответы на контрольные вопросы;

− выводы по лабораторной работе.

5. Защитите лабораторную работу перед преподавателем.



# ХОД РАБОТЫ

**Текст программы:**

**Содержимое файла main.py:**

if \_\_name\_\_ == "\_\_main\_\_":

    A = []

    s = *list*(map(*lambda* *i*: i.strip(",.:"), input("Введите строку: ").lower().split()))

Выравнивание строки входящей строки по нижнему регистру и деление строки на массив с помощью split(), очистка символов ‘,’ ‘.’ ‘:’ с помощью лямбда функции и метода strip().

for i in range(len(s)):

        c = s.count(s[i])

        if c>1:

            A.append((c,s[i]))

    С помощью метода count() проводим подсчёт каждого слова в массиве, если совпадений больше 1, то добавляем кортеж (количество, слово) его появлений в массив ответов A.

    answ = *list*(*set*(A))

    answ.sort()

    answ.reverse()

Удаление повторов методом set() и обратная сортировка списка.

    for i in answ:

        print(*f*"|{i[1]*:^10*}|{*str*(i[0])*:^10*}|")

Вывод каждого кортежа массива на экран в виде таблицы с использованием f-строк.

# Тестирование

Пример работы программы с исходным массивом со строкой “Раз два три два два раз три два один два”:



Пример работы программы со стихом А.С. Пушкина:

Я помню чудное мгновенье:

Передо мной явилась ты,

Как мимолетное виденье,

Как гений чистой красоты.

В томленьях грусти безнадежной,

В тревогах шумной суеты,

Звучал мне долго голос нежный

И снились милые черты.

Шли годы. Бурь порыв мятежный

Рассеял прежние мечты,

И я забыл твой голос нежный,

Твои небесные черты.

В глуши, во мраке заточенья

Тянулись тихо дни мои

Без божества, без вдохновенья,

Без слез, без жизни, без любви.

Душе настало пробужденье:

И вот опять явилась ты,

Как мимолетное виденье,

Как гений чистой красоты.

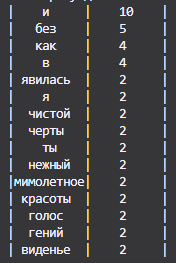
И сердце бьется в упоенье,

И для него воскресли вновь

И божество, и вдохновенье,

И жизнь, и слезы, и любовь.

Результат работы программы:



Наиболее часто встречающимися оказались слова “и” и “без”.

# ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

**Раскройте понятие строки в языке программирования Python.**

1. Строка в Python — это последовательность символов, представленных в виде текста, и она может содержать буквы, цифры, символы пунктуации и другие символы.
2. **Что такое строковый литерал?**

Строковый литерал — это представление строки в исходном коде программы. Он заключается в одинарные (' '), двойные (" ") или тройные (''' ''' или """ """) кавычки.

1. **Зачем нужны тройные кавычки при написании строки?**

Тройные кавычки используются для создания многострочных строк, сохраняя форматирование текста, и также позволяют включать одинарные и двойные кавычки внутри строки без экранирования.

1. **Как обозначается символ переноса на новую строку?**

Символ переноса на новую строку обозначается как '\n' в строках Python.

1. **Как работают со строками операции «+», «\*»?**

Операция '+' используется для конкатенации (сложения) строк, а '\*' для повторения строки заданное количество раз.

1. **Каким образом производится обращение к отдельным символам строки?**

Обращение к отдельным символам строки происходит через индексацию, например, **my\_string[0]** вернет первый символ строки.

1. **Как найти длину строки?**

Длину строки можно найти с использованием функции **len()**, например, **len(my\_string)**.

1. **Каким образом выбрать часть строки?**

Часть строки выбирается с использованием срезов (slicing), например, **my\_string[1:4]** вернет подстроку с 1-го по 3-й символы.

1. **Какие способы создания строк вы знаете?**

Строки можно создавать с использованием одинарных, двойных кавычек или тройных кавычек. Также можно преобразовывать другие типы данных в строки с помощью функции **str()**.

1. **Как разбить строку по определённому символу?**

Строку можно разбить на части с использованием метода **split()**, указав разделитель, например, **my\_string.split(',')**.

1. **Как создать одну строку из списка строк?**

Список строк можно объединить в одну строку с использованием метода **join()**, например, **' '.join(my\_list)**.

1. **Какие основные типы методов строк существуют?**

Основные типы методов строк включают методы для поиска подстрок, замены, преобразования регистра, удаления пробелов и многие другие.

1. **Какие методы и операции обработки строк вы использовали?**

Некоторые методы строк, которые часто используются, включают **lower()** (преобразование в нижний регистр), **upper()** (в верхний регистр), **replace()** (замена подстроки), а также операции конкатенации и срезы.

1. **Каким образом можно выводить на экран значения переменных и выражений внутри строки?**

Для вывода значений переменных внутри строки используется форматирование строк. Например, с использованием f-строк или метода **format()**.

1. **Как вывести на экран значения в столбец с заданной шириной?**

Значения могут быть выведены в столбец с заданной шириной, используя методы форматирования строк, такие как **f-строки** или метод **format()**, с заданием ширины поля вывода.

# ВЫВОД

В ходе выполнения лабораторной работы были успешно изучены и применены основные операции и методы обработки строк в языке Python, такие как **split()** и **lower()**. Они позволяют корректно сравнивать слова и избежать искажений при анализе. Можно сделать вывод, что язык Python является мощным инструментом для работы с текстовой информацией, предоставляя богатый набор функций для обработки и анализа строк.