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 ЛАБОРАТОРНОЙ РАБОТЕ №6
дисциплины «Программирование на Python»

Выполнил:
Середа Кирилл Витальевич
1 курс, группа ИВТ-б-о-22-1,
09.03.01 «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 со строками на языке Python

Цель: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

Ход выполнения:

1) Пример 1 Дано предложение. Все пробелы в нем заменить символом «_».

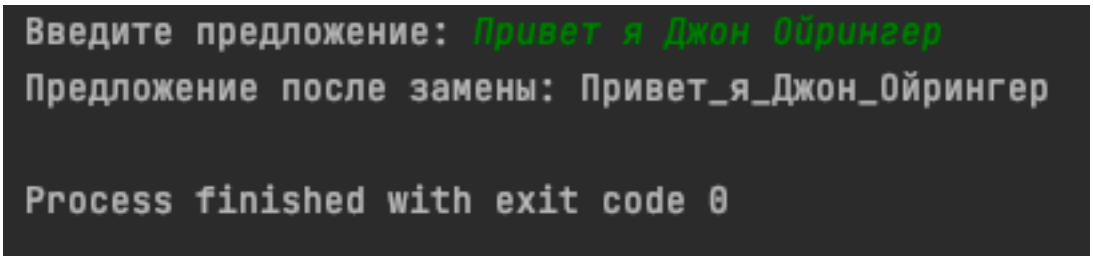

Рисунок 1 – Результат работы программы

Код программы:

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

if __name__ == '__main__':
    s = input("Введите предложение: ")
    r = s.replace(' ', '_')
    print(f"Предложение после замены: {r}")
```

2) Пример 2 Дано слово. Если его длина нечетная, то удалить среднюю букву, в противном случае – две средние буквы.

Рисунок 2 – Результат работы программы

Код программы:

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

if __name__ == '__main__':
    word = input("Введите слово: ")
    idx = len(word) // 2
    if len(word) % 2 == 1:
        # Длина слова нечетная.
        r = word[:idx] + word[idx+1:]
    else:
        # Длина слова четная.
        r = word[:idx-1] + word[idx+1:]
    print(r)
```

3) Пример 3 Дана строка текста, в котором нет начальных и конечных пробелов. Необходимо изменить ее так, чтобы длина строки стала равна заданной длине (предполагается, что требуемая длина не меньше

исходной). Это следует сделать путем вставки между словами дополнительных пробелов. Количество пробелов между отдельными словами должно отличаться не более чем на 1

```
Введите предложение: Эрмитаж в Санкт-Петербурге
Введите длину: 40
Эрмитаж         в         Санкт-Петербурге

Process finished with exit code 0
```

Рисунок 3 - Результат работы программы

Код программы:

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

import sys
if __name__ == '__main__':
    s = input("Введите предложение: ")
    n = int(input("Введите длину: "))
    # Проверить требуюемую длину.
    if len(s) >= n:
        print(
            "Заданная длина должна быть больше длины предложения",
            file=sys.stderr
        )
        exit(1)
    # Разделить предложение на слова.
    words = s.split(' ')
    # Проверить количество слов в предложении.
    if len(words) < 2:
        print(
            "Предложение должно содержать несколько слов",
            file=sys.stderr
        )
        exit(1)
    # Количество пробелов для добавления.
    delta = n
    for word in words:
        delta -= len(word)
    # Количество пробелов на каждое слово.
    w, r = delta // (len(words) - 1), delta % (len(words) - 1)
    # Сформировать список для хранения слов и пробелов.
    lst = []
    # Пронумеровать все слова в списке и перебрать их.
    for i, word in enumerate(words):
        lst.append(word)
        # Если слово не является последним, добавить пробелы.
        if i < len(words) - 1:
            # Определить количество пробелов.
            width = w
            if r > 0:
                width += 1
                r -= 1
            # Добавить заданное количество пробелов в список.
            if width > 0:
                lst.append(' ' * width)
```

```
    # Вывести новое предложение, объединив все элементы списка lst.
    print(''.join(lst))
```

4) Индивидуальное задание 1 – Вариант 1 - Дано название футбольного клуба. Напечатать его на экране «столбиком».

```
Введите название футбольного клуба: Динамо
Д
и
н
а
м
о
```

Рисунок 4 - Результат работы программы

Код программы:

```
club_name = input("Введите название футбольного клуба: ")

# Вывод названия клуба "столбиком"
for char in club_name:
    print(char)
```

5) Индивидуальное задание 2 – Вариант 1 - Дано предложение. Определить, есть ли буква “а” в нем. В случае положительного ответа найти также порядковый номер первой из них.

```
Введите предложение: Америка, а точнее США
6
```

Рисунок 5 - Результат работы программы

Код программы:

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

import sys

if __name__ == '__main__':
    text = input("Введите предложение: ")

    for i in range(len(text)):
        if text[i] == "a":
            print(i)
            break
        elif i == len(text)-1:
            print(
                'Буква "a" не найдена',
                file=sys.stderr
            )
```

6) Дано слово.

- Удалить из него третью букву.
- Удалить из него k-ю букву.

```
Введите слово: Россия
Введите номер буквы для удаления: 1
Исходное слово: Россия
Измененное слово: осия
```

Рисунок 6 – Результат работы программы

Код программы:

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

if __name__ == '__main__':
    word = input("Введите слово: ")
    k = int(input("Введите номер буквы для удаления: "))
    if k <= 0 or k > len(word):
        print("Некорректный номер буквы для удаления.")
    else:
        new_word = word[:2] + word[3:]
        new_word = new_word[:k - 1] + new_word[k:]
        print("Исходное слово:", word)
        print("Измененное слово:", new_word)
```

7) Задание повышенной сложности - Вариант 7: Даны два слова. Напечатать только те буквы слов, которые есть лишь в одном из них (в том числе повторяющиеся). Например, если заданные слова процессор и информация, то ответом должно быть: п е с с и ф м а я.

```
Введите первое слово:Процессор
Введите второе слово:Информация
П е с с И н ф м а и я
```

Рисунок 7 – Результат работы программы

Код программы:

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

if __name__ == '__main__':

    word1 = input("Введите первое слово:")
    word2 = input("Введите второе слово:")
    nullword = ''
    for i in range(len(word1)):
        if word1[i] not in word2:
            nullword += word1[i]+' '

    for i in range(len(word2)):
        if word2[i] not in word1:
            nullword += word2[i]+' '
```

```
print(nullword)
```

Ответы на вопросы:

1. Что такое строки в языке Python? В Python строка представляет собой последовательность символов и является неизменяемым типом данных.

2. Какие существуют способы задания строковых литералов в языке Python? Существуют три способа задания строк:

- Одинарные кавычки: str1 = 'Пример строки'
- Двойные кавычки: str2 = "Ещё один пример"
- Тройные кавычки для многострочных строк: str3 = '''Много строк пример'''

3. Какие операции и функции существуют для строк? Некоторые операции и функции для строк включают + (конкатенация), * (повторение), len() (длина строки), str() (преобразование в строку).

4. Как осуществляется индексирование строк? Индексация начинается с 0. Например, my_string[0] вернет первый символ строки.

5. Как осуществляется работа со срезами для строк? Срезы задаются с использованием start:stop:step. Например, my_string[1:5] вернет подстроку с индексами 1 до 4.

6. Почему строки Python относятся к неизменяемому типу данных? Строки в Python не могут быть изменены после создания, что обеспечивает их неизменяемость и сохранение данных.

7. Как проверить то, что каждое слово в строке начинается с заглавной буквы? Можно использовать метод istitle(), например: my_string.istitle().

8. Как проверить строку на вхождение в неё другой строки? Можно использовать оператор in. Например: substring in my_string.

9. Как найти индекс первого вхождения подстроки в строку? Используйте метод find(): index = my_string.find(substring).

10. Как подсчитать количество символов в строке? Используйте функцию len(): length = len(my_string).

11. Как подсчитать то, сколько раз определённый символ встречается в строке? Можно использовать метод count(): count = my_string.count(character).

12. Что такое f-строки и как ими пользоваться? F-строки — это способ форматирования строк, включая значения переменных. Пример: f"Привет, {name}!".

13. Как найти подстроку в заданной части строки? Используйте метод find(), указав диапазон индексов: index = my_string[start:end].find(substring).

14. Как вставить содержимое переменной в строку, воспользовавшись методом format()? Используйте метод format(): "Привет, {}!".format(name).

15. Как узнать о том, что в строке содержатся только цифры? Можно воспользоваться методом isdigit(): my_string.isdigit().

16. Как разделить строку по заданному символу? Используйте метод split(): my_list = my_string.split('-').

17. Как проверить строку на то, что она составлена только из строчных букв? Можно воспользоваться методом islower(): my_string.islower().

18. Как проверить то, что строка начинается со строчной буквы? Используйте метод islower() для первого символа: my_string[0].islower().

19. Можно ли в Python прибавить целое число к строке? Нет, строки и числа не могут быть прямо сложены. Требуется преобразование, например, result = str(number) + my_string.

20. Как «перевернуть» строку? Используйте срез с отрицательным шагом: reversed_string = my_string[::-1].

21. Как объединить список строк в одну строку, элементы которой разделены дефисами? Используйте метод join(): result_string = '-'.join(my_list).

22. Как привести всю строку к верхнему или нижнему регистру? Методы upper() и lower(): upper_case = my_string.upper(), lower_case = my_string.lower().

23. Как преобразовать первый и последний символы строки к верхнему регистру? Можно использовать upper() для первого символа и срез для последнего: result = my_string[0].upper() + my_string[1:-1] + my_string[-1].upper().

24. Как проверить строку на то, что она составлена только из прописных букв? Метод isupper(): my_string.isupper().

25. В какой ситуации вы воспользовались бы методом splitlines()? Метод splitlines() используется для разделения строки на список строк по символам новой строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки? Метод replace(): modified_string = my_string.replace(substring, new_substring).

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов? Методы startswith() и endswith(): my_string.startswith(prefix), my_string.endswith(suffix).

28. Как узнать о том, что строка включает в себя только пробелы? Можно использовать метод isspace(): my_string.isspace().

29. Что случится, если умножить некую строку на 3? Строка будет повторена три раза: result = my_string * 3.

30. Как привести к верхнему регистру первый символ каждого слова в строке? Можно воспользоваться методом title(): result = my_string.title().

31. Как пользоваться методом partition()? Метод partition() разбивает строку на три части по первому вхождению заданного разделителя, возвращая кортеж: (before, separator, after).

32. В каких ситуациях пользуются методом rfind()? Метод rfind() используется для поиска последнего вхождения подстроки в строку и возвращает индекс этого вхождения.

Вывод: в ходе выполнения лабораторной работы было изучено понятие строк и методы работы с ними. приобретены навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.