

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО  
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №6 по дисциплине  
«Основы программной инженерии»

Выполнил студент  
2 курса, группы ПИЖ-б-о-20-1  
Тотубалина С.С.

Проверил:  
Доцент кафедры инфокоммуникаций,  
Воронкин Р.А.

Ставрополь, 2021 г

## Ход работы

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     s = input("Enter the sentence: ")
6     r = s.replace(' ', '_')
7     print("The sentence after replacement: {}".format(r))
```

Рис. 1 – код программы lab.6\_ex.1.py

```
Enter the sentence: Hello, world!
The sentence after replacement: Hello,_world!

Process finished with exit code 0
```

Рис. 2 – пример работы программы lab.6\_ex.1.py

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     word = input("Enter the word: ")
6
7     idx = len(word) // 2
8     if len(word) % 2 == 1:
9         # Длина слова нечетная.
10        r = word[:idx] + word[idx+1:]
11    else:
12        # Длина слова четная.
13        r = word[:idx-1] + word[idx+1:]
14
15    print(r)
```

Рис. 3 – код программы lab.6\_ex.2.py

```
Enter the word: loop
lp

Process finished with exit code 0
```

Рис. 4 – вывод программы lab.6\_ex.2.py при чётной длине слова

```
Enter the word: cat
ct

Process finished with exit code 0
```

Рис. 5 – вывод программы lab.6\_ex.2.py при нечётной длине слова

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      import sys
5
6  ▶  if __name__ == '__main__':
7      s = input("Введите предложение: ")
8      n = int(input("Введите длину: "))
9
10     # Проверить требуемую длину.
11     if len(s) >= n:
12         print(
13             "Заданная длина должна быть больше длины предложения",
14             file=sys.stderr
15         )
16         exit(1)
17
18     # Разделить предложение на слова.
19     words = s.split(' ')
20     # Проверить количество слов в предложении.
21     if len(words) < 2:
22         print(
23             "Предложение должно содержать несколько слов",
24             file=sys.stderr
25         )
26         exit(1)
27
28     # Количество пробелов для добавления.
29     delta = n
```

Рис. 6 – код программы lab.6\_ex.3.py

```

28     # Количество пробелов для добавления.
29     delta = n
30     for word in words:
31         delta -= len(word)
32
33     # Количество пробелов на каждое слово.
34     w, r = delta // (len(words) - 1), delta % (len(words) - 1)
35
36     # Сформировать список для хранения слов и пробелов.
37     lst = []
38
39     # Пронумеровать все слова в списке и перебрать их.
40     for i, word in enumerate(words):
41         lst.append(word)
42
43         # Если слово не является последним, добавить пробелы.
44         if i < len(words) - 1:
45             # Определить количество пробелов.
46             width = w
47             if r > 0:
48                 width += 1
49                 r -= 1
50
51         # Добавить заданное количество пробелов в список.
52         if width > 0:
53             lst.append(' ' * width)
54
55     # Вывести новое предложение, объединив все элементы списка lst.
56     print(''.join(lst))

```

Рис. 7 – код программы lab.6\_ex.3.py

```

Введите предложение: cat is sleeping
Введите длину: 25
cat      is      sleeping

```

Рис. 8 – вывод при верном вводе

```

Введите предложение: cat is sleeping
Введите длину: 10
Заданная длина должна быть больше длины предложения

Process finished with exit code 1

```

Рис. 9 – вывод при неправильном вводе

```

1 ▶ 1 #!/usr/bin/env python3
2   2 # -*- coding: utf-8 -*-
3
4 ▶ 3 if __name__ == '__main__':
5     bucket_list = input("Введите текст: ")
6     result = len(bucket_list.split(' '))
7     print("В тексте " + str(result) + " слов.")

```

Рис. 10 – индивидуальное задание №1 (Вариант № 22), код программы individual\_6.1.py

```

Введите текст: однажды в студеную зимнюю пору
В тексте 5 слов.

Process finished with exit code 0

```

Рис. 11 – индивидуальное задание №1(Вариант № 22), результат программы individual\_6.1.py

```

1 ▶ 1 #!/usr/bin/env python3
2   2 # -*- coding: utf-8 -*-
3
4 ▶ 3 if __name__ == '__main__':
5     word = input(str("Введите слово: "))
6     word = list(word)
7
8     third = word[2]
9     last = word[-1]
10
11     word[2] = last
12     word[-1] = third
13
14     print("".join(word))

```

Рис. 12 – индивидуальное задание №2 (Вариант № 22), код программы individual\_6.2.py

```

Введите слово: привет
пртвеи

Process finished with exit code 0

```

Рис. 13 – индивидуальное задание №2 (Вариант № 22), результат программы individual\_6.2.py

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4  ▶  if __name__ == '__main__':
5
6      sentence = input("Введите предложение: ")
7      sentence = sentence.replace('с', '')
8      sentence = sentence.replace('С', '')
9      sentence = sentence.replace('с', '')
10     sentence = sentence.replace('С', '')
11     print(sentence)

```

Рис. 14 – индивидуальное задание №3 (Вариант № 22), код программы individual\_6.1.py

```

Введите предложение: Люблю весну в начале мая
Люблю вену в начале мая

Process finished with exit code 0

```

Рис. 15 – индивидуальное задание №3 (Вариант № 22), результат программы individual\_6.1.py

Ответы на вопросы:

1. Что такое строки в языке Python?

Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

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

Строки в апострофах и в кавычках, экранированные последовательности - служебные символы, "Сырые" строки, строки в тройных апострофах или кавычках.

3. Какие операции и функции существуют для строк?

Сложение, умножение, оператор принадлежности. Строковых функций в Python много, вот некоторые из них:

`chr()` – Преобразует целое число в символ

`ord()` – Преобразует символ в целое число

`len()` – Возвращает длину строки

`str()` – Изменяет тип объекта на `string`

4. Как осуществляется индексирование строк?

В Python строки являются упорядоченными последовательностями символьных данных и могут быть проиндексированы. Доступ к отдельным символам в строке можно получить, указав имя строки, за которым следует число в квадратных скобках `[]`. Индексации строк начинается с нуля: у первого символа индекс 0, следующего 1 и так далее. Индекс последнего символа в python — “длина строки минус один”.

5. Как осуществляется работа со срезами для строк?

Если `s` это строка, выражение формы `s[m:n]` возвращает часть `s`, начинающуюся с позиции `m`, и до позиции `n`, но не включая позицию. Если пропустить первый индекс, срез начинается с начала строки. Аналогично, если опустить второй индекс `s[n:]`, срез длится от первого индекса до конца строки.

6. Почему строки Python относятся к неизменяемому типу данных?

Более легкое представление в памяти.

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

`s.istitle()`

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

`if s1 in s2`

9. Как найти индекс первого вхождения подстроки в строку?

`s.find(<sub>).`

10. Как подсчитать количество символов в строке?

`len(s)`

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

`s.count(<char>).`

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

f-строки упрощают форматирование строк. Пример: `print(f' This is {name}, he is {age} years old')`

13. Как найти подстроку в заданной части строки?

`string.find(<sub>[, <start>[, <end>]])`

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

`'Hello, { }!'.format('Vasya')`

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

`string.isdigit()`

16. Как разделить строку по заданному символу?

`'foo.bar.baz.qux'.rsplit(sep='.')` – пример разделения

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

`string.islower()`

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

`s[0].isupper()`

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

С точки зрения математической операции нельзя, можно лишь только вывести из без разделения друг от друга

20. Как «перевернуть» строку?



`s[::-1]` – при помощи среза.

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

`'-'.join(<iterable>)`

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

К верхнему – `string.upper()`, к нижнему – `string.lower()`.

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

`s[0].upper() s[len(s) - 1].upper()`

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

`s.isupper()`

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

Если нужно сохранить символы, обозначающие конец слов.

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

`s.replace('что заменить', 'на что заменить')`

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

`string.endswith(<suffix>[, <start>[, <end>]]), str.startswith(prefix[, start[, end]])`

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

`s.isspace()`

29. Что случится, если умножить некую строку на 3?

Будет получена копия исходной строки в трёхкратном размере.

30. Как привести к верхнему регистру первый символ каждого слова в строке?

`s.title()`

31. Как пользоваться методом `partition()` ?

`s.partition(<sep>)` отделяет от `s` подстроку длиной от начала до первого вхождения `<sep>` .

Возвращаемое значение представляет собой кортеж из трех частей: Часть s до <sep>

Разделитель <sep> Часть s после <sep>

32. В каких ситуациях пользуются методом rfind() ?

Когда нужен индекс последнего вхождения подстроки в строку.