

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

**Отчет по лабораторной работе №4
Работа со строками в языке Python**

Выполнил студент группы

ИТС-б-з-22-1

Потеев А.С. « »_____2023г.

Подпись студента_____

Работа защищена « »_____2023г.

Проверил доцент, кандидат технических
наук, доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

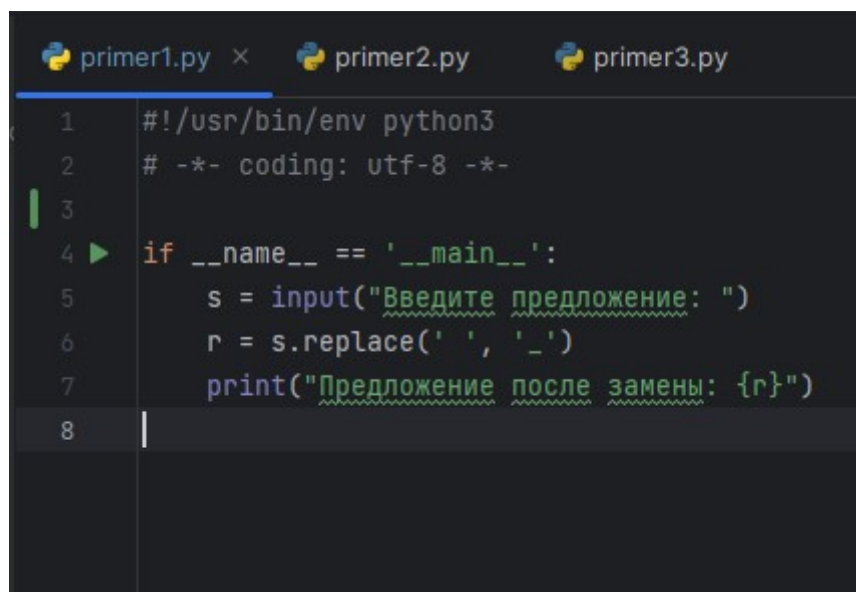
Ставрополь, 2023 г.

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

Порядок выполнения работы:

Создал общедоступный репозиторий на GitHub (<https://github.com/poteevprog/Lab4>)

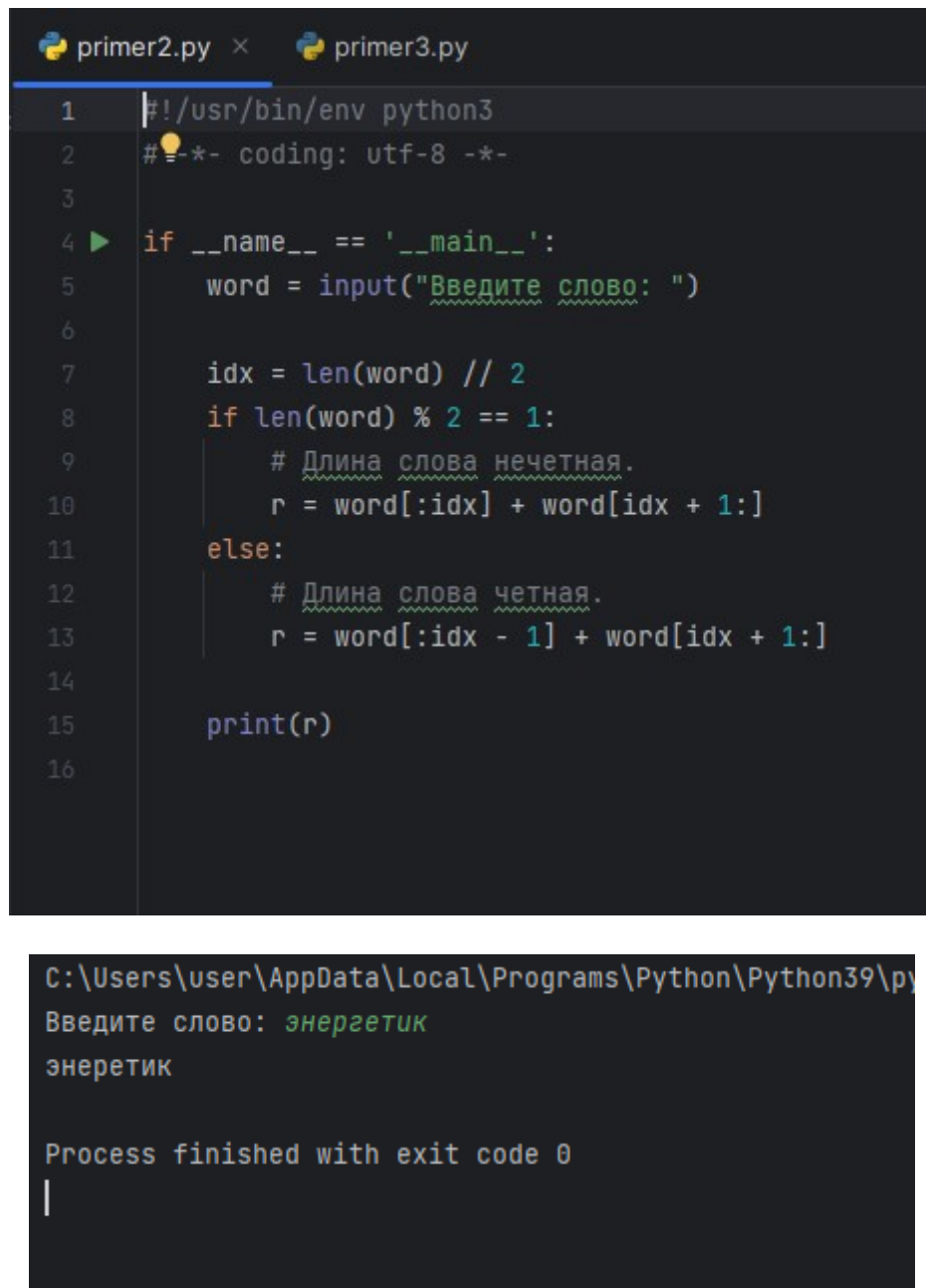
Пример 1. Дано предложение. Все пробелы в нем заменить символом «_».



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      s = input("Введите предложение: ")
6      r = s.replace(' ', '_')
7      print("Предложение после замены: {r}")
8
```

Рисунок 1 – Окно программы примера 1

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



```
primer2.py x primer3.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      word = input("Введите слово: ")
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)
16
```

```
C:\Users\user\AppData\Local\Programs\Python\Python39\py
Введите слово: энергетик
энергетик

Process finished with exit code 0
|
```

Рисунок 2 – Окно программы примера 2

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.

```
primer3.py x
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
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))
57

```

```

C:\Users\user\AppData\Local\Programs\Python\Python39\python.exe
Введите предложение: я люблю кошек и собак
Введите длину: 30
я    люблю    кошек    и    собак

Process finished with exit code 0
|

```

Рисунок 3 – Окно программы примера 3

Индивидуальное задание 1.

12. Дано предложение. Вывести все имеющиеся в нем буквосочетания нн.

```
individual1.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  ► if __name__ == '__main__':
5      x = input("Введите предложение: ")
6      k = x.count('нн')
7      if k == 0:
8          print("В предложении нет 'нн'.")
9      else:
10         for i in range(k):
11             print('нн')
12
```

```
"C:\Users\user\Desktop\учеба\1 курс\2 семестр\Основы кроссплатформенного пр
Введите предложение: деревянный стол, стеклянный бокал, оловянный солдатик
нн
нн
нн

Process finished with exit code 0
```

Рисунок 4 – Окно программы для первой задачи и проверка работоспособности кода.

Индивидуальное задание 2.

12. Дано предложение. Напечатать все символы, расположенные между первой и второй запятыми. Если второй запятой нет, то должны быть напечатаны все символы, расположенные после единственной имеющейся запятой.

```
individ2.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      x = input("Введите предложение: ")
6      z = ','
7      res = [i for i in range(len(x)) if x.startswith(z, i)]
8
9      if len(res) == 0:
10         print("В предложении нет запятых.")
11     elif len(res) == 1:
12         print(x[res[0] + 1:])
13     else:
14         print(x[res[0] + 1:res[1]])
15
```

```
"C:\Users\user\Desktop\учеба\1 курс\2 семестр\Основы кроссплатформенного про
Введите предложение: мама мыла раму, а Митя, Таня, Боря смотрели телевизор.
а Митя

Process finished with exit code 0
|
```

Рисунок 5 – Окно программы для второй задачи и проверка работоспособности кода.

Индивидуальное задание 3.

12. Путем вставок и удаления символов исправить ошибки:

- в слове прроцесор;
- во фразе теекстовыйфайл;
- во фразе програма и аллгоритм;
- во фразе процессор и паммать.

```
individual3.py ×
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  ▶ if __name__ == '__main__':
5      a = 'прроцесор'
6      b = 'теекстовыйфайл'
7      c = 'програма и аллгоритм'
8      d = 'процесор и паммать'
9
10     a = a.replace("pp", "p")
11     a = a.replace("c", "cc")
12
13     b = b.replace("ee", "e")
14     b = b[0:9] + ' ' + b[9:]
15
16     c = c.replace("лл", "л")
17     c = c.replace("м", "мм", 1)
18
19     d = d.replace("мм", "м")
20     d = d.replace("c", "cc")
21
22     print(a)
23     print(b)
24     print(c)
25     print(d)
```

```
"C:\Users\user\Desktop\учеба\1 курс\2 семестр
процессор
текстовый файл
программа и алгоритм
процессор и память

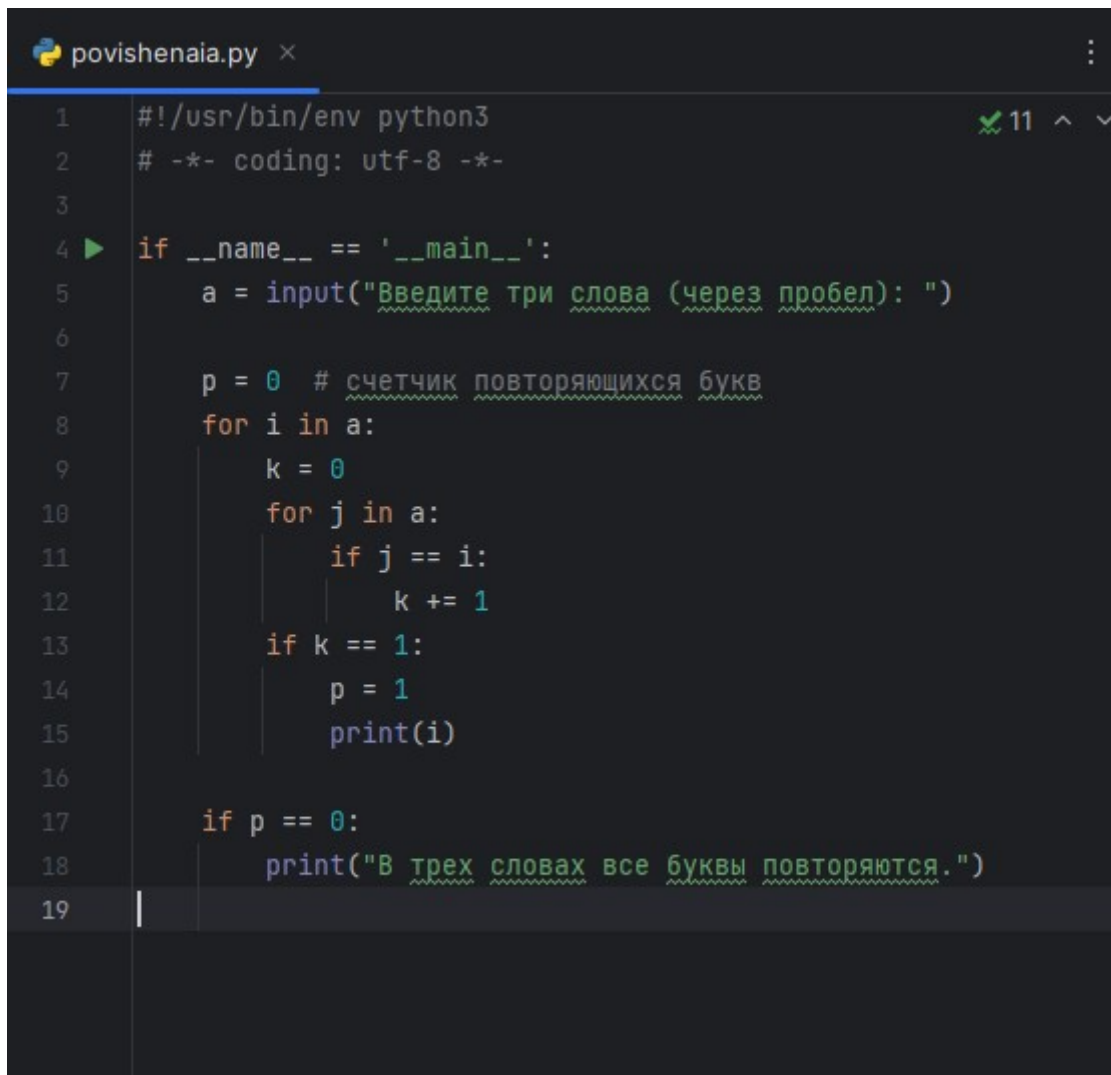
Process finished with exit code 0
```

Рисунок 6 – Окно программы для третьей задачи и проверка работоспособности кода.

Задание повышенной сложности.

12. Даны три слова. Напечатать неповторяющиеся в них буквы.

```
povishenaia.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  ▶ if __name__ == '__main__':
5      a = input("Введите три слова (через пробел): ")
6
7      p = 0 # счетчик повторяющихся букв
8      for i in a:
9          k = 0
10         for j in a:
11             if j == i:
12                 k += 1
13             if k == 1:
14                 p = 1
15                 print(i)
16
17         if p == 0:
18             print("В трех словах все буквы повторяются.")
19
```



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      a = input("Введите три слова (через пробел): ")
6
7      p = 0 # счетчик повторяющихся букв
8      for i in a:
9          k = 0
10         for j in a:
11             if j == i:
12                 k += 1
13         if k == 1:
14             p = 1
15             print(i)
16
17     if p == 0:
18         print("В трех словах все буквы повторяются.")
19
```

Рисунок 7 – Окно программы для задания повышенной сложности и проверка работоспособности кода.

Вывод: Я на лабораторной работе приобрел навыки по работе со строками при написании программ с помощью языка программирования Python.