

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 №4
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Плещенко Данила Георгиевич
1 курс, группа ИТС-б-о-21-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук, доцент
кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

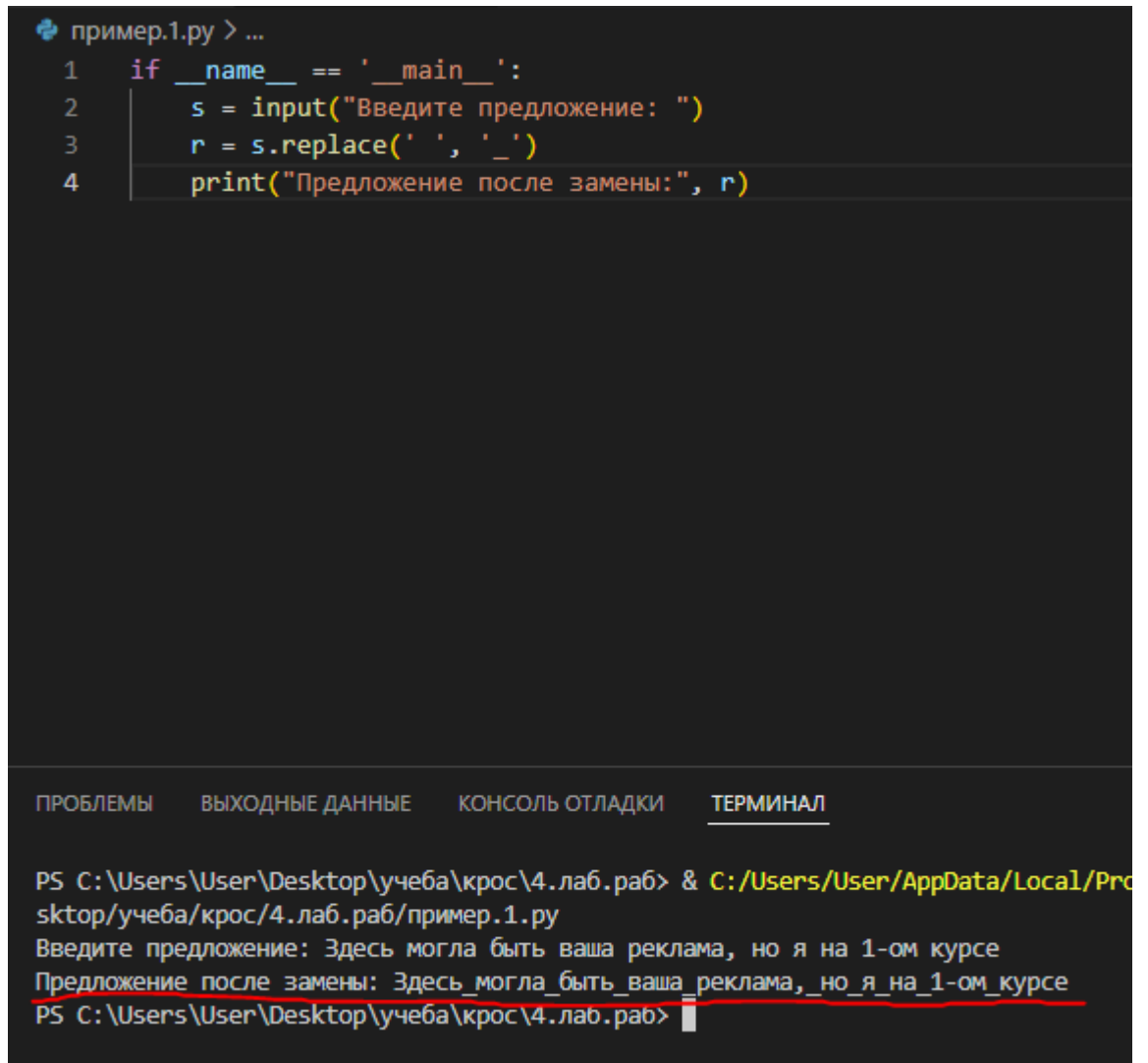
Тема: Работа со строками в языке Python

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

Ход работы

Пример 1

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



```

+ пример.1.py > ...
1  if __name__ == '__main__':
2      s = input("Введите предложение: ")
3      r = s.replace(' ', '_')
4      print("Предложение после замены:", r)

```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```

PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> & C:/Users/User/AppData/Local/Programs/Python/Python38-64/Python.exe C:\Users\User\Desktop\учеба\крос\4.лаб.раб\пример.1.py
Введите предложение: Здесь могла быть ваша реклама, но я на 1-ом курсе
Предложение после замены: Здесь_могла_быть_ваша_реклама,_но_я_на_1-ом_курсе
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб>

```

Рисунок 1. Работа программы «Пример 1»

Пример 2

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

```

пример.2.py > ...
1  if __name__ == '__main__':
2      word = input("Введите слово: ")
3      idx = len(word) // 2
4      if len(word) % 2 == 1:
5          # Длина слова нечетная.
6          r = word[:idx] + word[idx+1:]
7      else:
8          # Длина слова четная.
9          r = word[:idx-1] + word[idx+1:]
10     print(r)

```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```

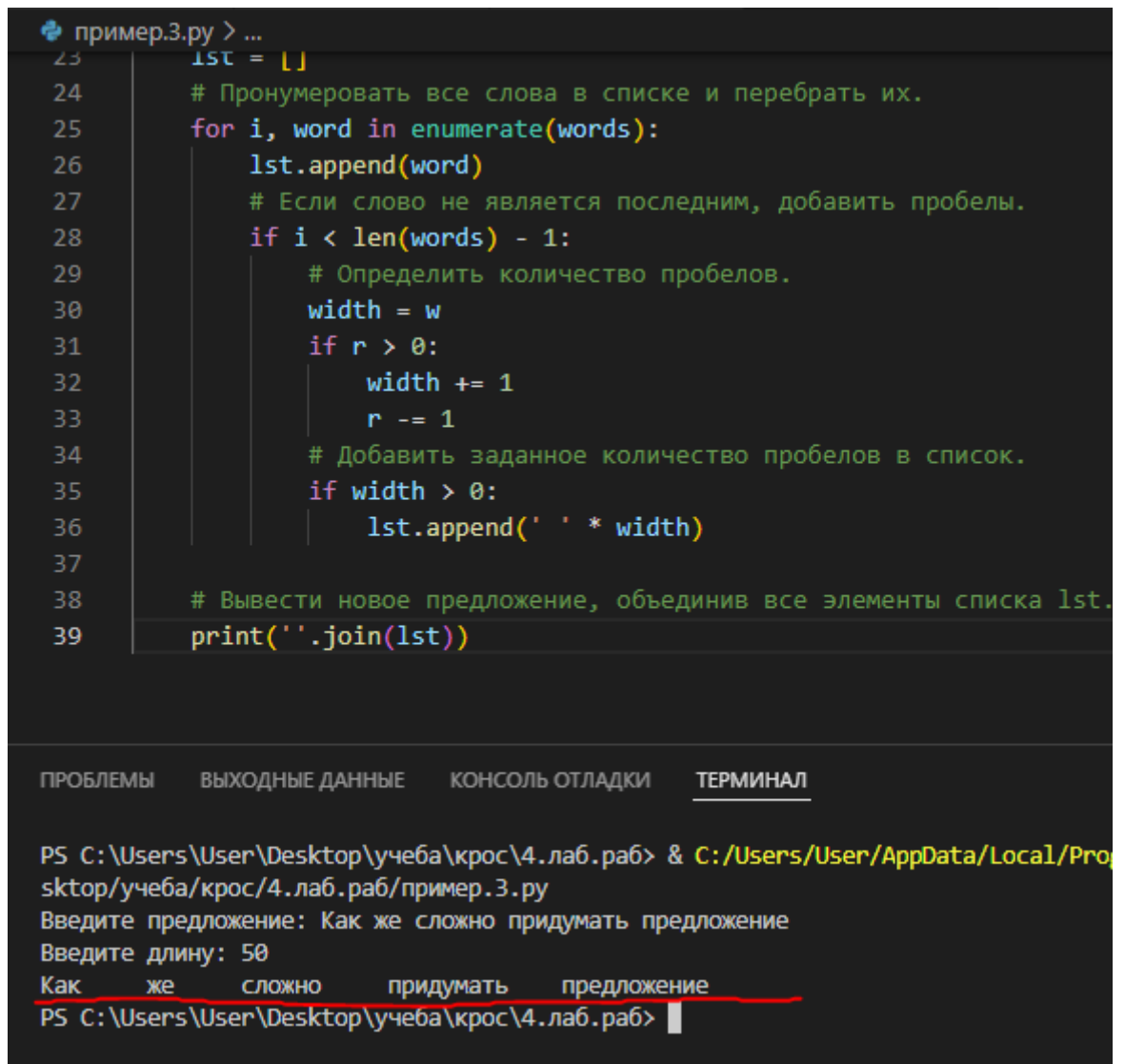
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> & C:/U
sktop/учеба/крос/4.лаб.раб/пример.2.py
Введите слово: PUBG
PG
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб>

```

Рисунок 2. Работа программы «Пример 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.



```

пример.3.py > ...
23 lst = []
24 # Пронумеровать все слова в списке и перебрать их.
25 for i, word in enumerate(words):
26     lst.append(word)
27     # Если слово не является последним, добавить пробелы.
28     if i < len(words) - 1:
29         # Определить количество пробелов.
30         width = w
31         if r > 0:
32             width += 1
33             r -= 1
34         # Добавить заданное количество пробелов в список.
35         if width > 0:
36             lst.append(' ' * width)
37
38 # Вывести новое предложение, объединив все элементы списка lst.
39 print(''.join(lst))

```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```

PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> & C:/Users/User/AppData/Local/Programs/Python/Python39-64/Python.exe C:\Users\User\Desktop\учеба\крос\4.лаб.раб\пример.3.py
Введите предложение: Как же сложно придумать предложение
Введите длину: 50
Как    же    сложно    придумать    предложение
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб>

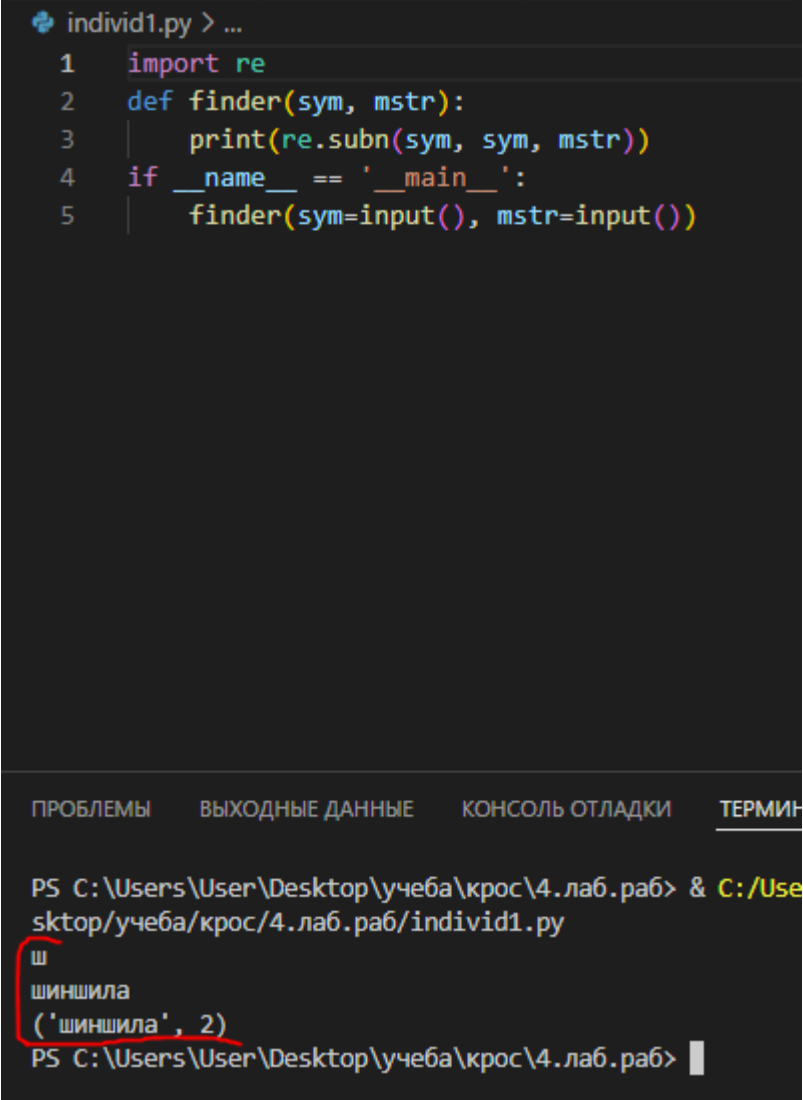
```

Рисунок 3. Работа программы «Пример 3»

Индивидуальные задания

Задание 1

Дано предложение. Определить число вхождений в него некоторого символа.



```
individ1.py > ...
1  import re
2  def finder(sym, mstr):
3      print(re.subn(sym, sym, mstr))
4  if __name__ == '__main__':
5      finder(sym=input(), mstr=input())
```

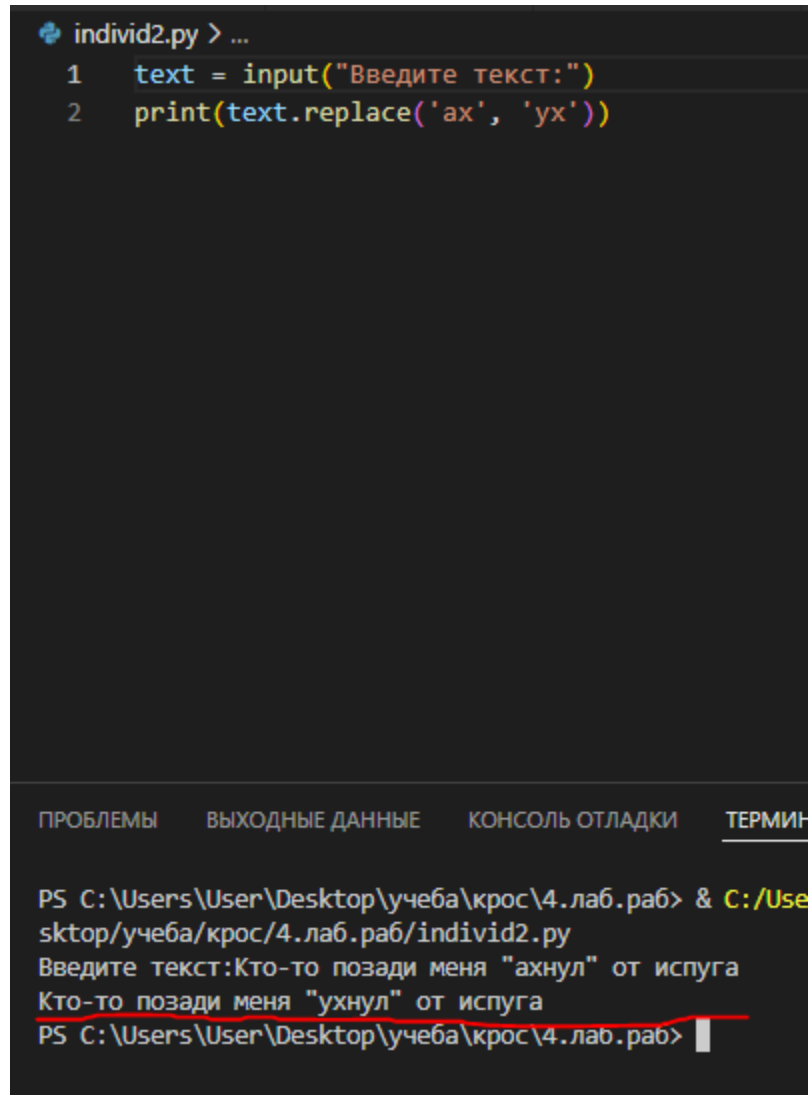
ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> & C:/Use
sktop/учеба/крос/4.лаб.раб/individ1.py
ш
шиншила
('шиншила', 2)
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> |
```

Рисунок 4. Работа программы «Задание 1»

Задание 2

Дано предложение. Заменить в нем все вхождения буквосочетания **ах** на **ух**.



```
individ2.py > ...
1  text = input("Введите текст:")
2  print(text.replace('ах', 'ух'))
```

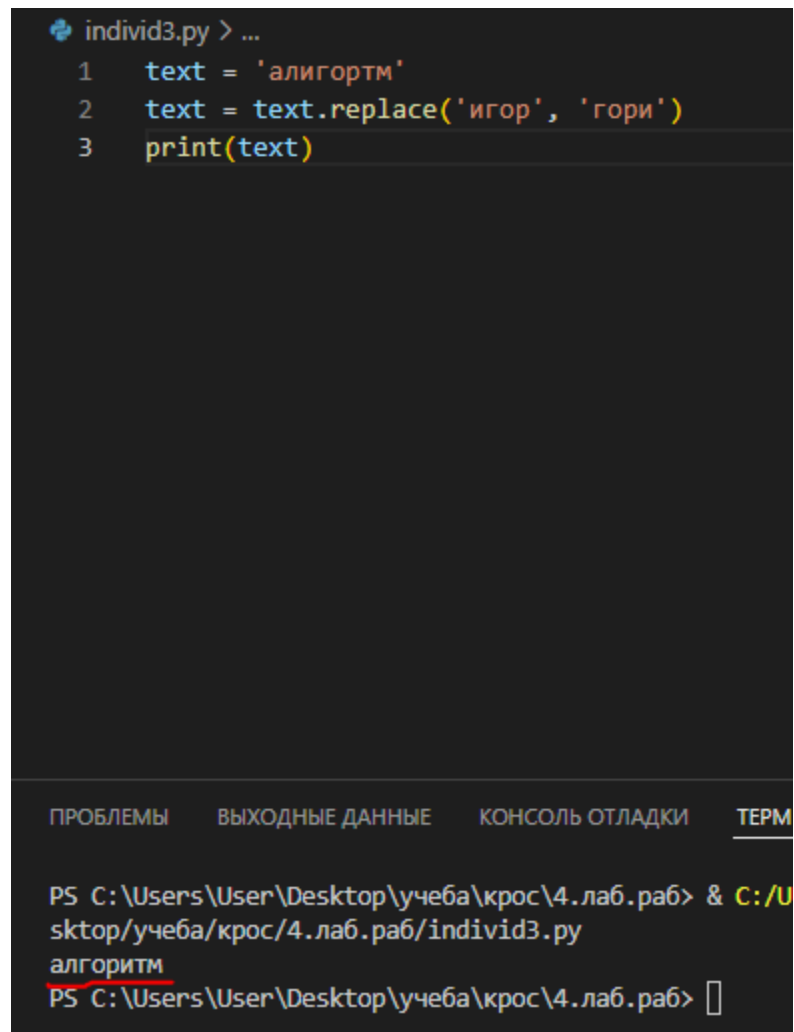
ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> & C:/Use
sktop/учеба/крос/4.лаб.раб/individ2.py
Введите текст:Кто-то позади меня "ахнул" от испуга
Кто-то позади меня "ухнул" от испуга
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> |
```

Рисунок 5. Работа программы «Задание 2»

Задание 3

Дано ошибочно написанное слово алигортм. Путем перемещения его букв получить слово алгоритм.



```
individ3.py > ...
1 text = 'алигортм'
2 text = text.replace('игор', 'гори')
3 print(text)
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> & C:/U
sktop/учеба/крос/4.лаб.раб/individ3.py
алгоритм
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> █

Рисунок 6. Работа программы «Задание 3»

Повышенная сложность

Дано предложение. Определить количество слов:

- начинающихся с буквы н;
- оканчивающихся буквой р.

```
indmax.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  if __name__ == '__main__':
4      t = input()
5      r = sum(1 for x in t.split() if x[0] == "н")
6      q = sum(1 for x in t.split() if x[-1] == "р")
7      print(r, q)
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> & C:/Users/User/App
sktop/учеба/крос/4.лаб.раб/indmax.py
нор нро рно рон
2 1
PS C:\Users\User\Desktop\учеба\крос\4.лаб.раб> |
```

Рисунок 7 Работа программы повышенной сложности

Вывод: Я приобрёл навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.