

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 № 2.3
дисциплины «Программирование на Python»

Вариант 10

Выполнил:
Кенесбаев Хилол Куат улы
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

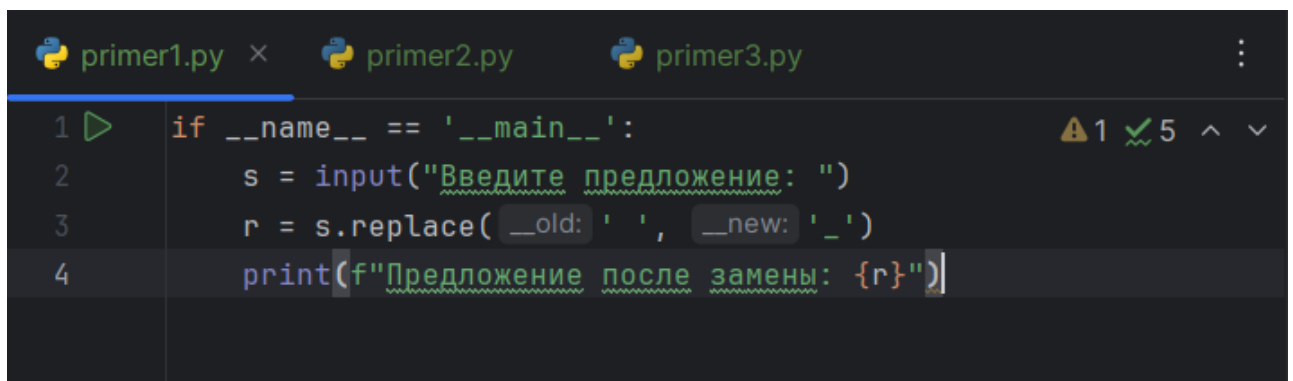
Ставрополь, 2023 г.

Тема: Работа со строками в языке питон

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

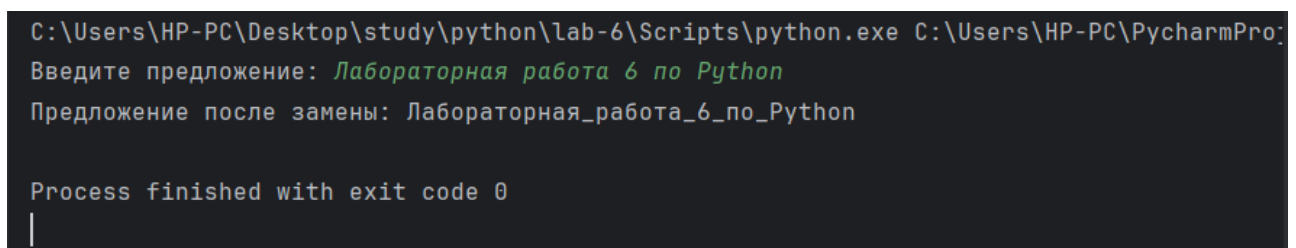
Порядок выполнения работы:

1. Проработал примеры лабораторной работы:



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

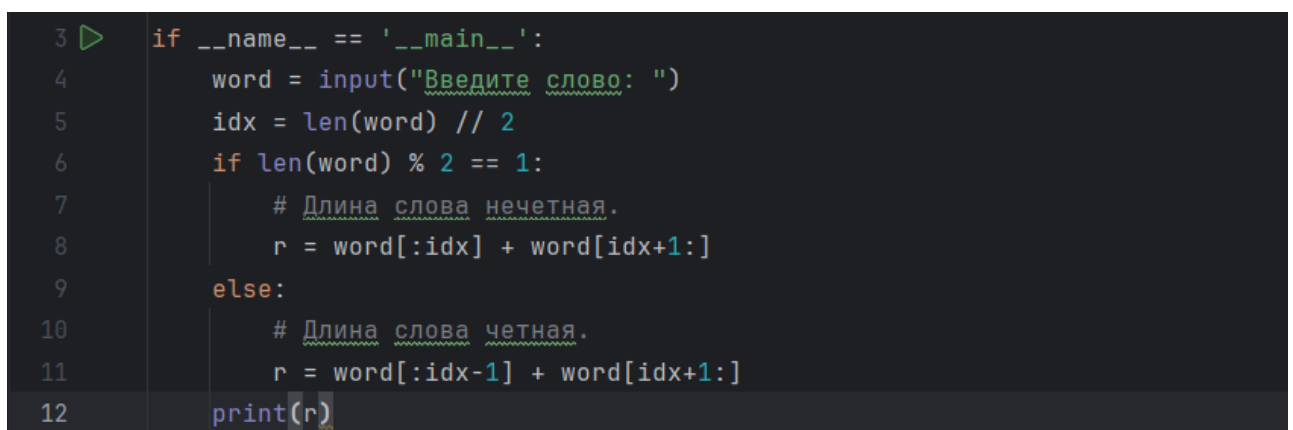
Рисунок 1 primer1.py



```
C:\Users\HP-PC\Desktop\study\python\lab-6\Scripts\python.exe C:\Users\HP-PC\PycharmPro:
Введите предложение: Лабораторная работа 6 по Python
Предложение после замены: Лабораторная_работа_6_по_Python

Process finished with exit code 0
|
```

Рисунок 2 Запуск программы primer1.py



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

Рисунок 3 primer2.py

```

C:\Users\HP-PC\Desktop\study\python\lab-6\Scripts\python.exe C:\Users\HP-PC\PycharmProjects\lab-6\primer2.py
Введите слово: hello
hello

Process finished with exit code 0

```

Рисунок 4 Запуск программы primer2.py

```

1  import sys
2
3
4  if __name__ == '__main__':
5      s = input("Введите предложение: ")
6      n = int(input("Введите длину: "))
7      # Проверить требуемую длину.
8      if len(s) >= n:
9          print(
10             "Заданная длина должна быть больше длины предложения",
11             file=sys.stderr
12         )
13         exit(1)
14     # Разделить предложение на слова.
15     words = s.split(' ')
16     # Проверить количество слов в предложении.
17     if len(words) < 2:
18         print(
19             "Предложение должно содержать несколько слов",
20             file=sys.stderr
21         )
22
23 if __name__ == '__main__':

```

Рисунок 5 primer3.py

```

C:\Users\HP-PC\Desktop\study\python\lab-6\Scripts\python.exe C:\Users\HP-PC\PycharmProjects\lab-6\primer3
Введите предложение: лабораторная работа 6 по Python
Введите длину: 45
лабораторная    работа    6    по    Python

Process finished with exit code 0

```

Рисунок 6 Запуск программы primer3.py

2. Выполнил первое индивидуальное задание:

Дано предложение. Вывести все буквы “м, и, н” в нем

Код программы:

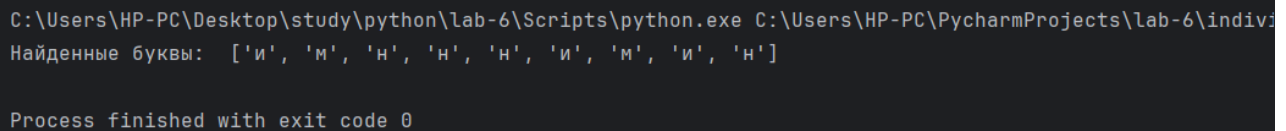
```

def find_letters(sentence):
    result = [char for char in sentence if char in ['м', 'и', 'н']]
    return result

```

```
# Заданное предложение
input_sentence = "Пример заданного предложения, содержащего буквы 'м', 'и', 'н'."

# Вызываем функцию для поиска букв и выводим результат
found_letters = find_letters(input_sentence)
print("Найденные буквы: ", found_letters)
```



```
C:\Users\HP-PC\Desktop\study\python\lab-6\Scripts\python.exe C:\Users\HP-PC\PycharmProjects\lab-6\individualnoe1.py
Найденные буквы: ['и', 'м', 'н', 'н', 'н', 'и', 'м', 'и', 'н']

Process finished with exit code 0
```

Рисунок 7 Результат работы программы individualnoe1.py

3. Выполнил второе индивидуальное задание:

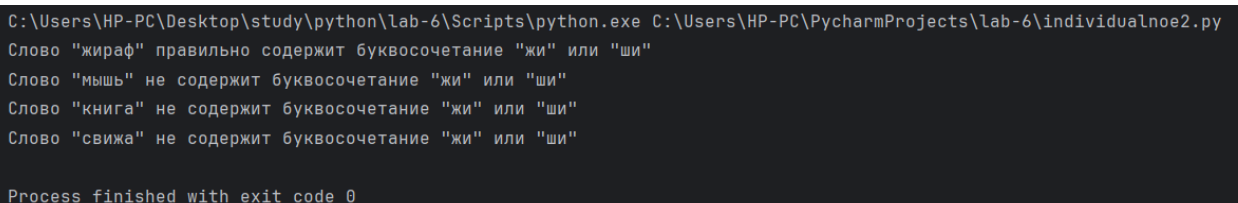
Дана последовательность слов. Проверить, правильно ли в ней записаны буквосочетания “жи” и “ши”.

Код программы:

```
def check_sequence(words_sequence):
    for word in words_sequence:
        if "жи" in word or "ши" in word:
            print(fСлово "{word}" правильно содержит буквосочетание "жи"
или "ши")
        else:
            print(fСлово "{word}" не содержит буквосочетание "жи" или "ши")
```

```
# Заданная последовательность слов
input_words_sequence = ["жираф", "мышь", "книга", "свижа"]

# Вызываем функцию для проверки последовательности слов
check_sequence(input_words_sequence)
```



```
C:\Users\HP-PC\Desktop\study\python\lab-6\Scripts\python.exe C:\Users\HP-PC\PycharmProjects\lab-6\individualnoe2.py
Слово "жираф" правильно содержит буквосочетание "жи" или "ши"
Слово "мышь" не содержит буквосочетание "жи" или "ши"
Слово "книга" не содержит буквосочетание "жи" или "ши"
Слово "свижа" не содержит буквосочетание "жи" или "ши"

Process finished with exit code 0
```

Рисунок 8 Результат работы программы individualnoe2.py

4. Выполнил второе индивидуальное задание:

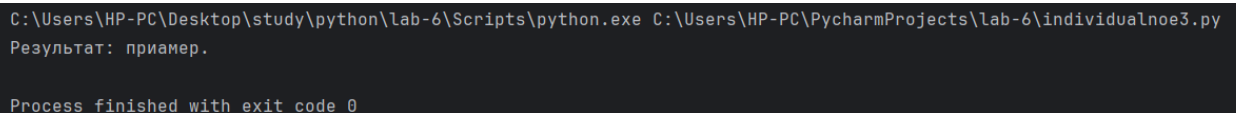
Дано слово, оканчивающееся символом “. ”. Вставить заданную букву после первой буквы “и”.

Код программы:

```
def insert_letter(word, letter_to_insert):
    for i, char in enumerate(word):
        if char == 'и':
            word = word[:i+1] + letter_to_insert + word[i+1:]
            break
    return word

# Заданное слово и буква для вставки
input_word = "пример." # Пример заданного слова
inserted_letter = "а" # Буква для вставки

# Вызываем функцию для вставки буквы после первой буквы "и" в слове
resulting_word = insert_letter(input_word, inserted_letter)
print("Результат:", resulting_word)
```



```
C:\Users\HP-PC\Desktop\study\python\lab-6\Scripts\python.exe C:\Users\HP-PC\PycharmProjects\lab-6\individualnoe3.py
Результат: пример.

Process finished with exit code 0
```

Рисунок 9 Результат работы программы individualnoe3.py

Ответы на контрольные 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 - подавляют экранирование, строки в тройных апострофах или кавычках.

3. Какие операции и функции существуют для строк?

Оператор сложения (+), умножения (*), принадлежности подстроки (in).

Функции:

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 [].

Индексация 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?

Python также допускает возможность извлечения подстроки из строки, известную как «string slice». Если s это строка, выражение формы s[m:n] возвращает часть s, начинающуюся с позиции m, и до позиции n, но не включая позицию.

Существует еще один вариант синтаксиса среза, о котором стоит упомянуть. Добавление дополнительного «:» и третьего индекса означает шаг, который указывает, сколько символов следует пропустить после извлечения каждого символа в срезе.

6. Почему строки Python относятся к неизменяемому типу данных?

Строки – один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. На самом деле нет особой необходимости изменять строки. Обычно можно легко

сгенерировать копию исходной строки с необходимыми изменениями.

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

`string.istitle()` определяет, начинаются ли слова строки с заглавной буквы.

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

`s.find(<sub>)` возвращает первый индекс в `s` который соответствует началу строки `<sub>` , сели же в `s` нет `<sub>`, то функция выдаст -1

9. Как найти индекс первого вхождения подстроки в строку?

`s.find(<sub>)` возвращает первый индекс в `s` который соответствует началу строки `<sub>` , сели же в `s` нет `<sub>`, то функция выдаст -1

10. Как подсчитать количество символов в

строке? `len(s)` возвращает количество символов в строке `s`.

11. Как подсчитать то, сколько раз определённый символ

встречается в строке?

`s.count(<sub>)` возвращает количество точных вхождений подстроки

`<sub>` в `s`.

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

В Python версии 3.6 был представлен новый способ форматирования строк. Эта функция официально названа литералом отформатированной строки, но обычно упоминается как f- строки (f-string).

Возможности форматирования строк огромны и не будут подробно описана здесь.

Одной простой особенностью f-строк, которые вы можете начать использовать сразу, является интерполяция переменных. Вы можете указать имя переменной непосредственно в f-строковом литерале (f'string'), и python заменит имя соответствующим значением.

Пример: `print(f"Произведение {n} на {m} равно {prod}")`, где `m`, `n`, `prod` это переменные.

13. Как найти подстроку в заданной части строки? `s.find(подстрока, начало, конец)`.

14. Как вставить содержимое переменной в строку, воспользовавшись методом `format()`?
`print('{}'.format(s))`.

15. Как узнать о том, что в строке содержатся только цифры? `s.isdigit()` возвращает `True` когда строка `s` не пустая и все ее символы являются цифрами, а `False` если нет.

16. Как разделить строку по заданному символу? `str.split('заданный символ')`.

17. Как проверить строку на то, что она составлена только из строчных букв?
`s.islower()` возвращает `True`, если строка `s` не пустая, и все содержащиеся в ней буквенные символы строчные, а `False` если нет. Не алфавитные символы игнорируются.

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

`S[0].islower()` выдаст `True` если строка начинается со строчной буквы и

`False` если нет.

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

20. Как «перевернуть» строку? `s[::-1]`.

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

`str.join('-', s)`, где `s` – это список строк.

22. Как привести всю строку к верхнему или нижнему регистру? `s.upper()`, `s.lower()`.

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

`string[0].upper() + string[1:-1] + string[-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()`?

`splitlines()` делит `s` на строки и возвращает их в список. Любой из

следующих символов или последовательностей символов считается

границей строки: `\n`, `\r`, `\r\n`, `\v` или же `\x0b`, `\f` или же `\x0c`, `\x1c`, `\x1d`, `\x1e`, `\x85`, `\u2028`, `\u2029`.

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

`s.replace(old, new)`.

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

`str.startswith()` и `str.endswith()`.

28. Как узнать о том, что строка включает в себя только пробелы? `s.isspace()`.

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

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

`s.title()`.

31. Как пользоваться методом `partition()`?

Разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

32. В каких ситуациях пользуются методом `rfind()`?

`rfind()` и `find()` оба используются для поиска вхождения подстроки в строку, но есть различие в том, что `rfind()` ищет справа налево (с конца строки), в то время как `find()` ищет слева направо (с начала строки). То

есть `rfind()` находит последнее вхождение, а `find()` первое вхождение подстроки в строку.

Вывод: в результате выполнения работы были приобретены навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.