

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №4

Работа со строками в языке Python3

Выполнил студент группы

ИТС-б-о-20-1

Попов Д.В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший

преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2021

Цель работы: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python3.

Порядок выполнения работы:

Пример 1.

```
Введите предложение: текст текст текст
Предложение после замены: текст_текст_текст
Process finished with exit code 0
```

Рисунок 1 – пример работы программы

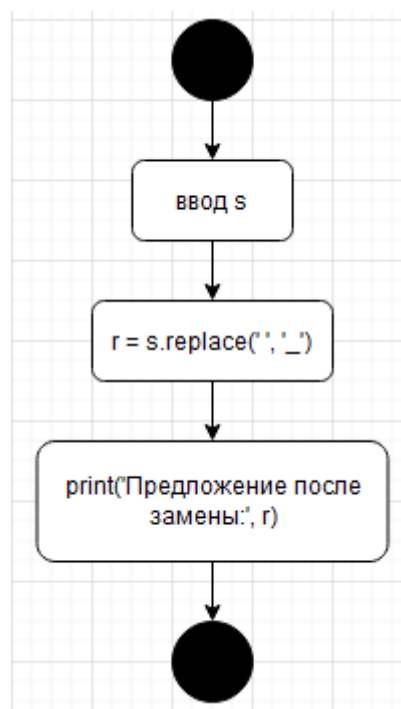


Рисунок 2 – UML диаграмма для примера 1

Пример 2.

```
Введите слово: слово
слово
Process finished with exit code 0
```

Рисунок 3 – пример работы программы при вводе «слово»

```

Введите слово: ввод
вд
Process finished with exit code 0

```

Рисунок 4 – пример работы программы при вводе «ввод»

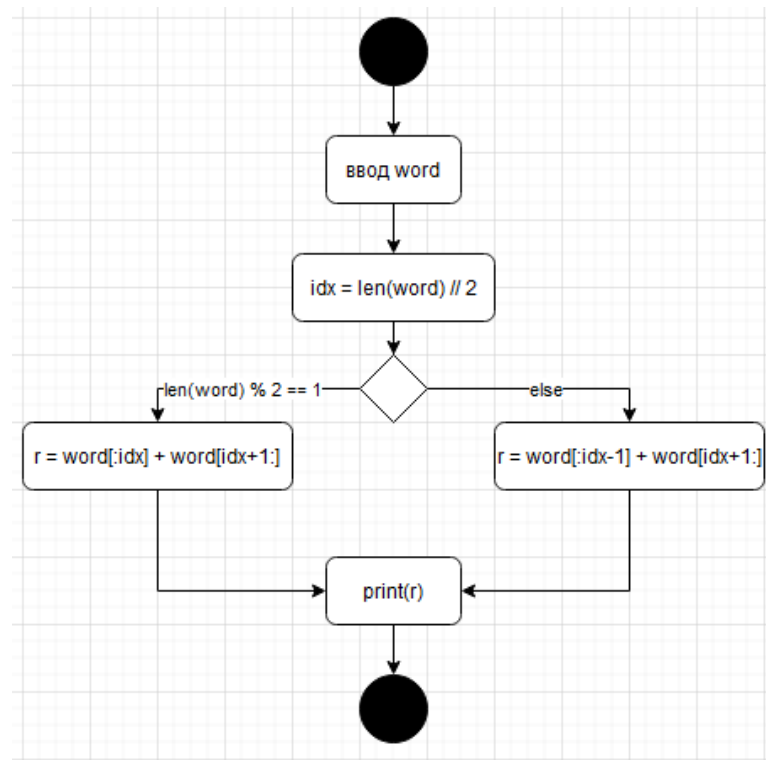


Рисунок 5 – UML диаграмма для примера 2

Пример 3.

```

Введите предложение: текст текст текст
Введите длину: 20
текст  текст  текст
Process finished with exit code 0

```

Рисунок 6 – пример работы программы при вводе «20»

```

Введите предложение: текст текст текст
Введите длину: 5
Заданная длина должна быть больше длины предложения
Process finished with exit code 1

```

Рисунок 7 – пример работы программы при вводе «5»

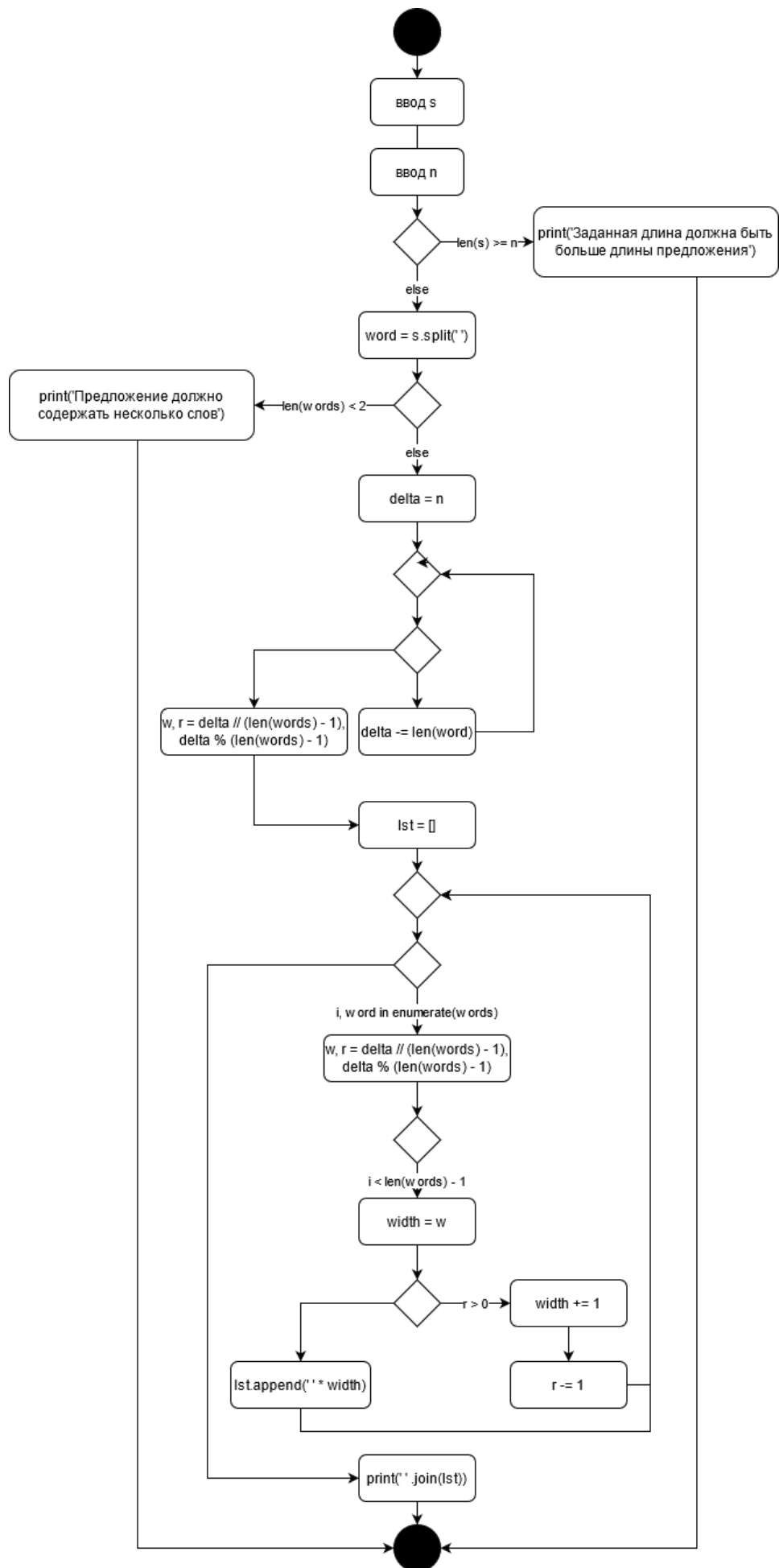


Рисунок 8 – UML диаграмма для примера 3

Индивидуальное задание 1.

11. Дано предложение. Составить программу, которая выводит все вхождения в предложение двух заданных символов.

```
Введите текст текст текст текст
Введите 2 символа ст
ст
ст
ст

Process finished with exit code 0
```

Рисунок 9 – Пример работы программы

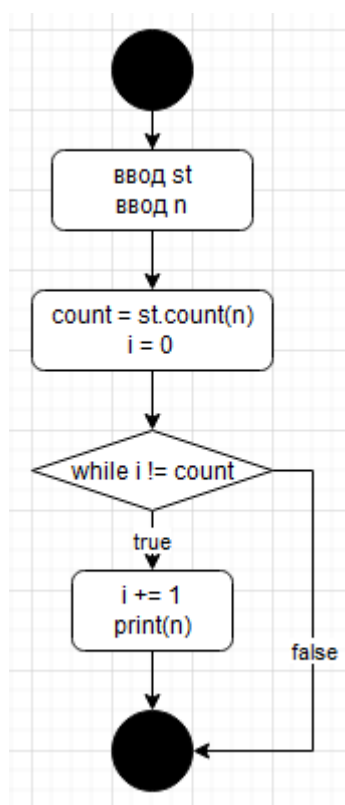


Рисунок 10 – UML диаграмма для индивидуального задания 1

Индивидуальное задание 2 .

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

```

Введите текст жишижишы
Ошибка жы - 1
Ошибка шы - 1

Process finished with exit code 0

```

Рисунок 11 – Пример работы программы

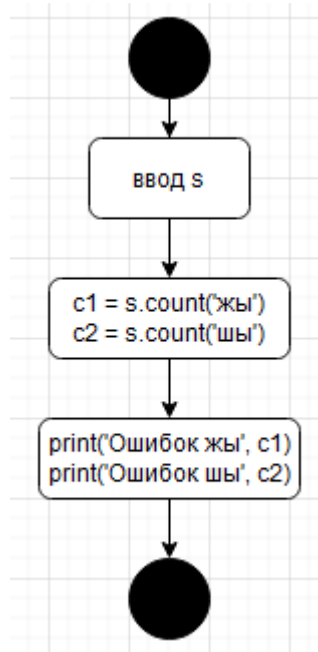


Рисунок 12 – UML диаграмма для индивидуального задания 2

Индивидуальное задание 3.

9. Дано слово, оканчивающее символом «.». Составить программу, которая вставляет некоторую заданную букву после буквы с заданным номером.

```

Введите слово текст.
Символ ы
Номер символа 3
тексыт.

Process finished with exit code 0

```

Рисунок 13 – Пример работы программы

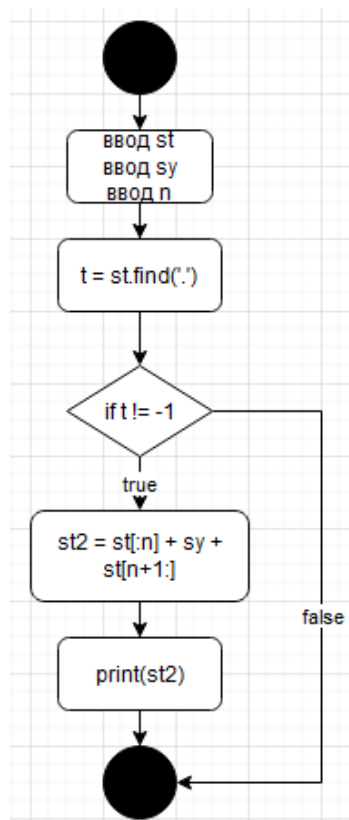


Рисунок 14 – UML диаграмма для индивидуального задания 3

Задание повышенного уровня.

8. Даны два слова. Напечатать только те буквы слов, которые встречаются в обоих словах лишь один раз. Например, если заданные слова процессор и информация, то ответом должно быть: п е ф м а я.

```

Первое слово процессор
Второе слово информация
п
е
н
ф
м
а
я

Process finished with exit code 0
  
```

Рисунок 14 – Пример работы программы

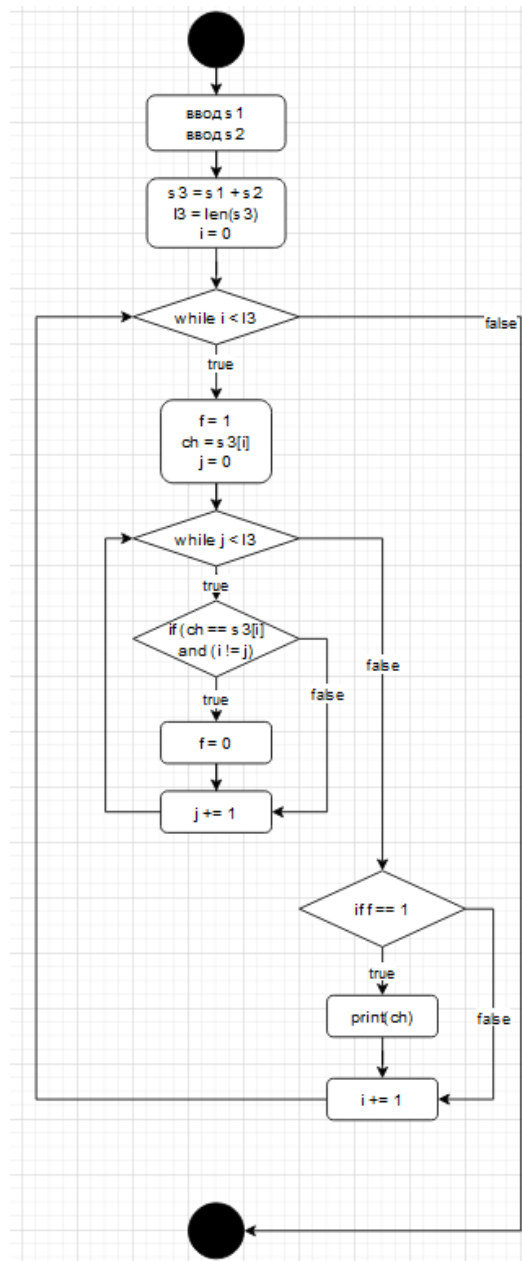


Рисунок 15 – UML диаграмма для задания повышенного уровня

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе со строками при написании программ с помощью языка программирования Python3.

Контрольные вопросы:

1. Что такое строки в языке Python?

Строки в Python – последовательности символов, используемые для хранения и представления текстовой информации.

2. Какие существуют способы задания строковых литералов в языке

Python?

1) строки в апострофах и в кавычках;

2) экранированные последовательности экранированные

последовательности (начинаются с символа \) позволяют вставить символы,

которые сложно ввести с клавиатуры;

3) строки в тройных апострофах или кавычках — главное достоинство

строк в тройных кавычках в том, что их можно использовать для записи

многострочных блоков текста.

3. Какие операции и функции существуют для строк?

Оператор сложения строк + и оператор умножения строк * - оператор создает несколько копий строки.

4. Как осуществляется индексирование строк?

В Python строки являются упорядоченными последовательностями символьных данных и могут быть проиндексированы. Индексация строк

начинается с нуля.

5. Как осуществляется работа со срезами для строк?

Python также допускает возможность извлечения подстроки из строки.

Если s это строка, выражение формы s[m:n] возвращает часть s, начинающуюся с позиции m и до позиции n.

6. Почему строки Python относятся к неизменяемому типу данных?

Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять.

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

буквы?

`.istitle()` определяет, начинаются ли слова строки с заглавной буквы. 8.

Как проверить строку на вхождение в неё другой строки?

Оператор `in` возвращает `True`, если подстрока входит в строку, и `False`, если нет.

9. Как найти индекс первого вхождения подстроки в строку?

`.find` или `.index`. Эти методы идентичны, за исключением того, что `.index` вызывает исключение `ValueError`, если объект не найден.

10. Как подсчитать количество символов в строке? `Len()`- возвращает длину строки.

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

`.count` подсчитывает количество вхождений подстроки в строку.

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

Эта функция официально названа литералом отформатированной строки, но обычно упоминается как f-строки. `f` перед кавычками строки укажет `python`, что это f-строка вместо стандартной.

13. Как найти подстроку в заданной части строки?

`.find` ищет в строке заданную подстроку.

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

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

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

`.isdigit()` определяет, состоит ли строка из цифр.

16. Как разделить строку по заданному символу?

`.split` делит строку на список из подстрок.

`s.split()` ведет себя как `s.rsplit()`, за исключением того, что при указании `<maxsplit>`, деление начинается с левого края `s`

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

`.islower()` определяет, являются ли буквенные символы строки строчными.

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

`.isupper()` определяет, являются ли буквенные символы строки заглавными. Используя срезы строки мы проверим это.

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

В некоторых языках это возможно, но Python при попытке выполнения подобной операции будет выдана ошибка `TypeError`.

19. Как «перевернуть» строку?

Можно перевернуть строку при помощи среза, `s[::-1]`

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

Метод `.join()` умеет объединять элементы списков в строки, разделяя отдельные строки с использованием заданного символа.

21. Как

`.lower()` преобразует все буквенные символы в строчные и `.upper()` преобразует все буквенные символы в заглавные.

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

Мы будем обращаться к символам строки по индексам.

```
animal = 'fish'
```

```
animal[0].upper() + animal[1:-1] + animal[-1].upper() #=> 'FisH'
```

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

`.isupper()` определяет, являются ли буквенные символы строки заглавными.

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

`s.splitlines()` делит `s` на строки и возвращает их в списке

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

`.replace` заменяет вхождения подстроки в строке.

27. Как проверить то, что строка начинается с заданной

последовательности символов, или заканчивается заданной последовательностью символов?

Для ответа на этот вопрос можно прибегнуть, соответственно, к методам `startswith()` и `endswith()`

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

`.isspace()` определяет, состоит ли строка только из пробельных символов.

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

Будет создана новая строка, представляющая собой исходную строку,

повторённую три раза

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

`.title()` преобразует первые буквы всех слов в заглавные

31. Как пользоваться методом `partition()` ?

`.partition(<sep>)` делит строку на основе разделителя.

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

`string.rfind` ищет в строке заданную подстроку, начиная с конца.

Вывод: при выполнении заданий были приобретены навыки по работе со строками при написании программ с помощью языка программирования

Python3.

Ссылка на репозиторий: <https://github.com/12W300/fourth.git>