МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №5

Выполнил студент группы ИВТ-б-о-20-1
Павленко М.С. « »_____20__г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверила Воронкин Р.А._____

(подпись)

Ставрополь 2021

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

https://github.com/0Roni0/5-lab.git

Ход работы

1. Проработал примеры из учебника:

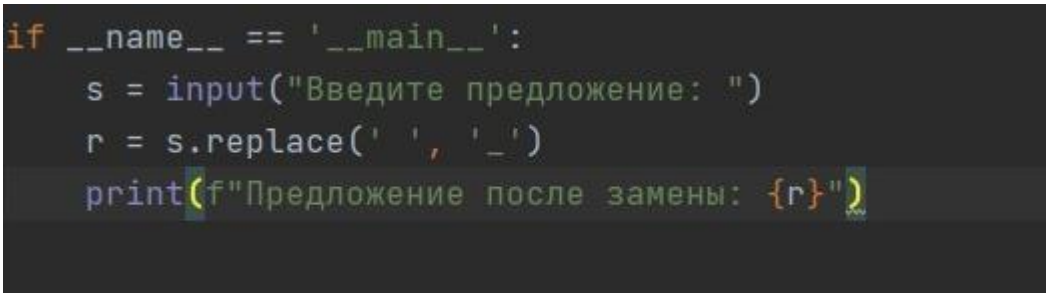

Рисунок 1 – Пример 1

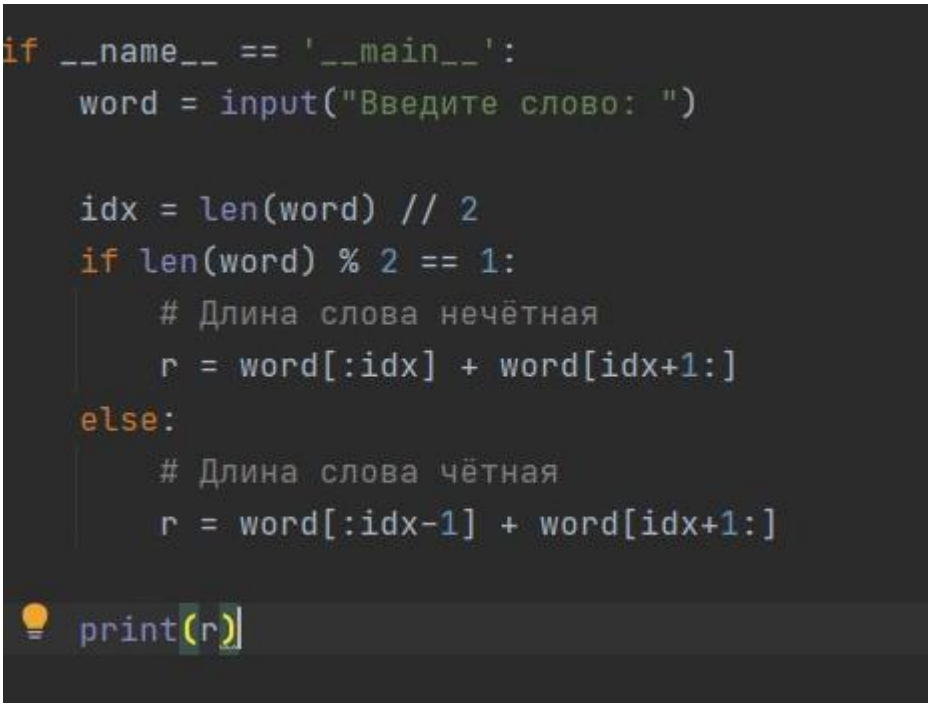

Рисунок 2 – Пример 2

```python
import sys

if __name__ == '__main__':
    s = input("Введите предложение: ")
    n = int(input("Введите длину: "))

    # Проверить требуюемую длину
    if len(s) >= n:
        print("Заданная длина должна быть больше длины предложения", file=sys.stderr)
        exit(1)

    # Разделить предложение на слова
    words = s.split(' ')
    # Проверить количество слов в предложении.
    if len(words) < 2:
        print("Предложение должно содержать несколько слов", file=sys.stderr)
        exit(1)

    # Количество пробелов для добавления
    delta = n
    for word in words:
        delta -= len(word)

    # Количество пробелов на каждое слово
    w, r = delta // (len(words) - 1), delta % (len(words) - 1)

    # Сформировать список для хранения слов и пробелов
    lst = []

    # Пронумеровать все слова в списке и перебрать их
    for i, word in enumerate(words):
        lst.append(word)
```

```
    # Если слово не является последним, добавить пробелы
    if i < len(words) - 1:
        # Определить количество пробелов
        width = w
        if r > 0:
            r -= 1

        # Добавить заданное количество проебелов в список
        if width > 0:
            lst.append(' ' * width)

# Вывести новое предложение, объединив все элементы списка lst
print(''.join(lst))
```

Рисунок 3 – Пример 3

2. Сделал uml-диаграммы для всех примеров

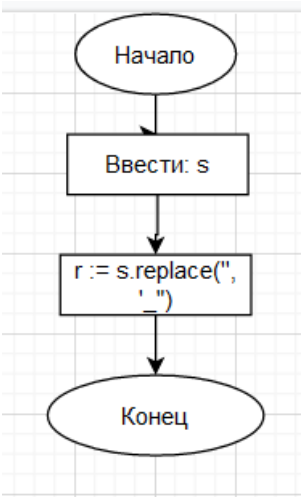

Рисунок 6 – Диаграмма для примера 1

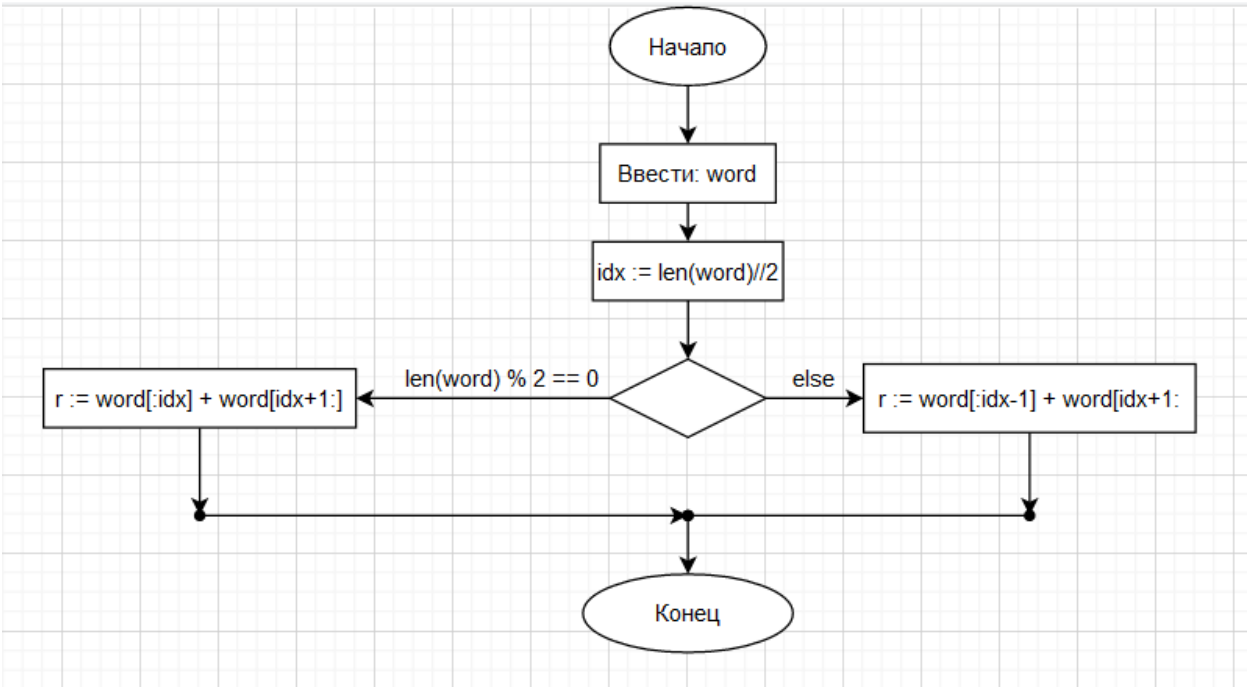

Рисунок 7 – Диаграмма для примера 2

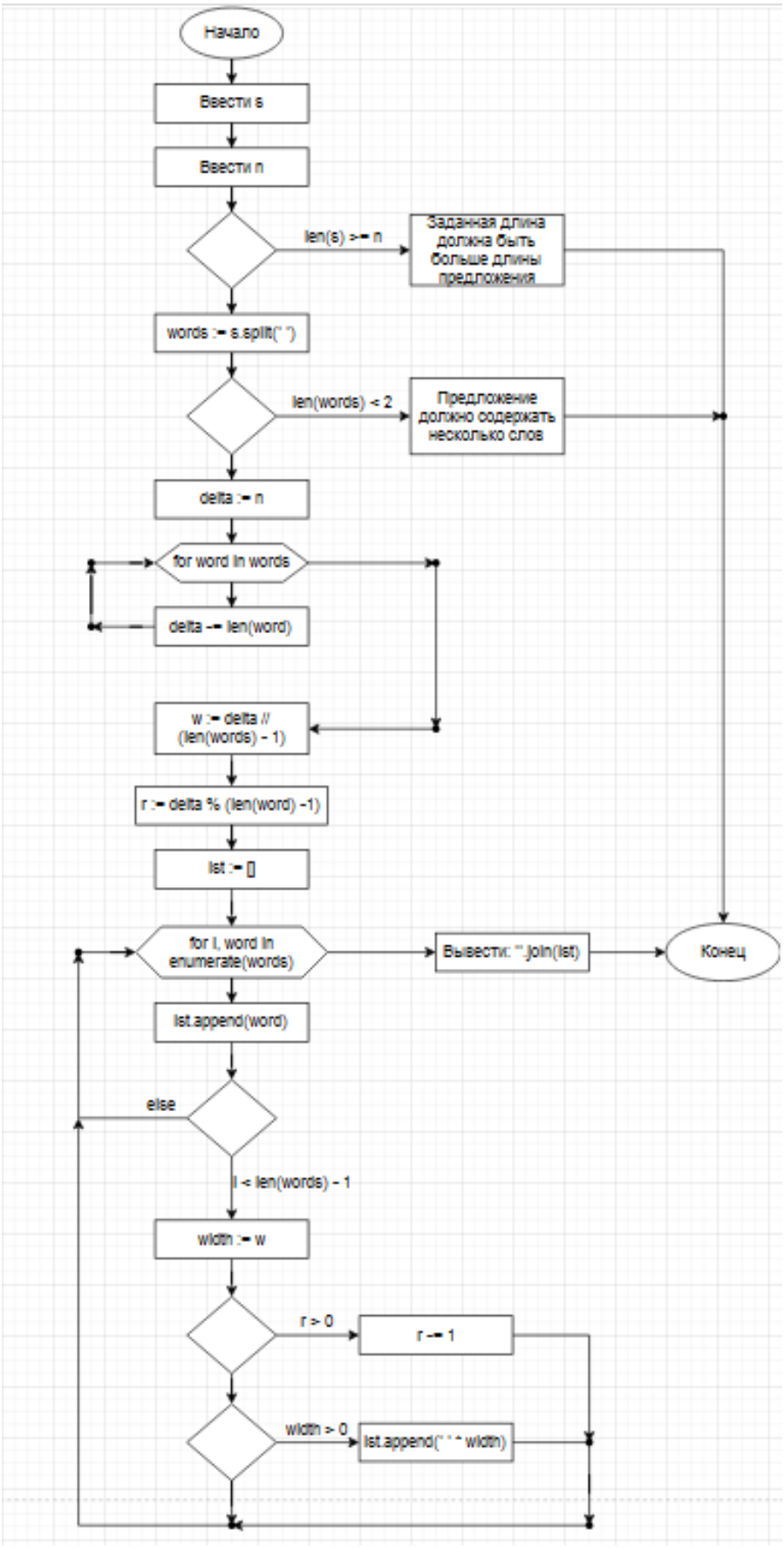

Рисунок 8 – Диаграмма примера 3

Задание 1

1. Код задания 1

```
s = input("Введите предложение: ")
print(f"Число пробелов равно: {s.count(' ')}")
```

Рисунок 9 – Задание 1

2. Результат работы

```
Введите предложение: Четвёртое января
Число пробелов равно: 1
```

Рисунок 10 – результат работы

3. Диаграмма

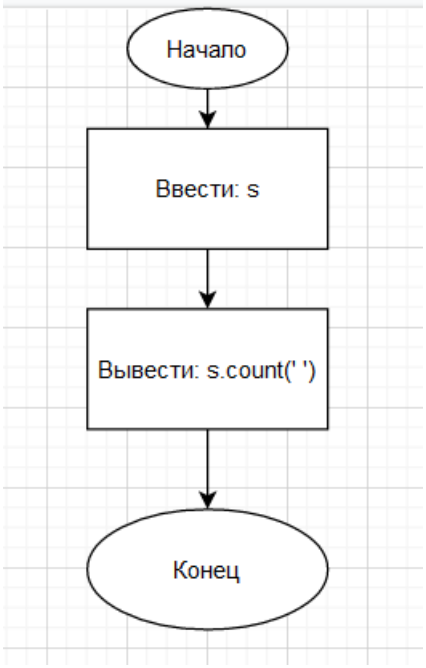

Рисунок 11 – Диаграмма задания 1

Задание 2

1. Код задания 2

```
s = input("Введите предложение: ")
# Порядковые номера
rep1 = 0
rep2 = 0

# Проходит по всем символам в предложении
for i in range(1, len(s)):
    if s[i-1] == s[i]:
        rep1 = i-1
        rep2 = i

if rep1 == 0 and rep2 == 0:
    print("Повторяющихся символов нет")
else:
    print(f"Порядковые номера повторяющихся символов: {rep1} и {rep2}")
```

Рисунок 12 – Код задания 2

2. Результат работы

```
Введите предложение: Странное дело
Порядковые номера повторяющихся символов: 4 и 5
```

Рисунок 13 – Результат работы задания 2

3. Диаграмма

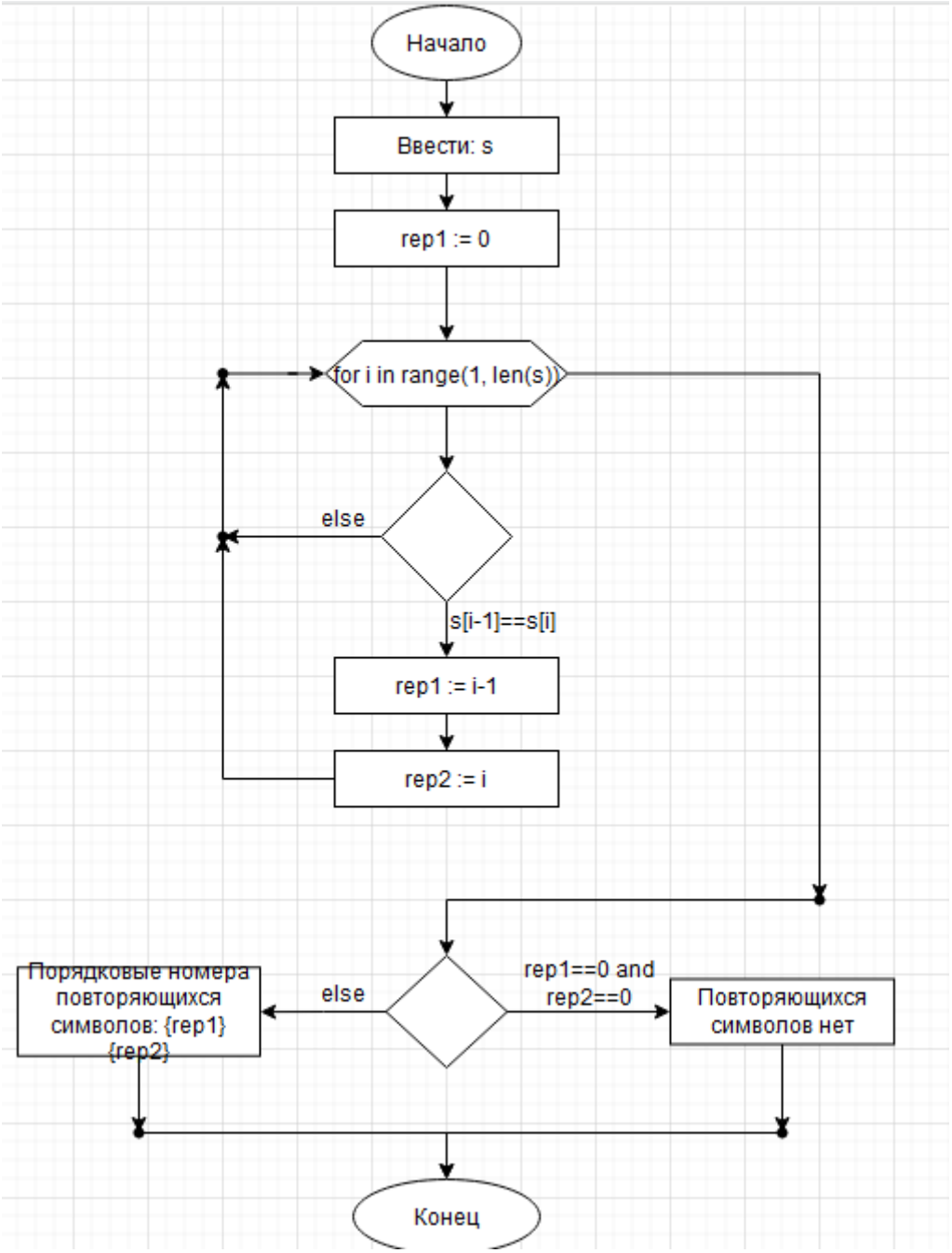

Рисунок 14 – Диаграмма задания 2

Задание 3

1. Код задания 1

```python
s = str(input("Введите предложение: "))
k = int(input("Введите число k: "))
k -= 1

# Переставить его последнюю букву наместо k-й
s = s.replace(s[k], s[len(s)-1])
# k - ю передвинуть вправо на одну позицию
s = s.replace(s[k+1], s[k])
# (k+1) - ю передвинуть вправо на одну позицию
s = s.replace(s[k+2], s[k+1])
# предпоследнюю букву вправо на одну позицию.
s = s.replace(s[len(s)-1], s[len(s)-2])

print(s)
```

Рисунок 15 – Код задания 3

2. Результат

```
Введите предложение: 123456789
Введите число k: 3
128886788
```

Рисунок 16 – Результат работы задания 3

3. Диаграмма

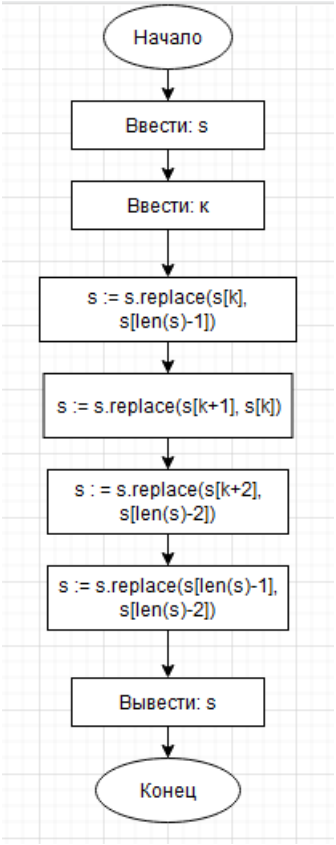

Рисунок 17 – Диаграмма задания 3

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 языкеPython?

– строки в апострофах и в кавычках;

– экранированные последовательности – экранированные последовательности (начинаются с символа \) позволяют вставить символы, которые сложно ввести с клавиатуры;

– 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 привести всю строку к верхнему или нижнему регистру?

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