

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Информатика»
Тема: Основные управляющие конструкции. Wikipedia API.

Студент гр. 0382

Шангичев В.А.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Изучить и применить на практике основные управляющие конструкции языка Python.

Задание.

Напишите программу, которая принимает на вход строку вида

название_страницы_1, название_страницы_2, ... название_страницы_n,
сокращенная_форма_языка

и делает следующее:

1. Проверяет, есть ли такой язык в возможных языках сервиса, если нет, выводит строку "no results" и завершает выполнение программы. В случае, если язык есть, устанавливает его как язык запросов в текущей программе.

2. Ищет максимальное число слов в кратком содержании страниц "название_страницы_1", "название_страницы_2", ... "название_страницы_n", выводит на экран это максимальное количество и название страницы (т.е. её **title**), у которой оно обнаружилось. Считается, что слова разделены пробельными символами.

Если максимальных значений несколько, выведите последнее.

3. Строит список-цепочку из страниц и выводит полученный список на экран.

Элементы списка-цепочки - это страницы "название_страницы_1", "название_страницы_2", ... "название_страницы_n", между которыми может быть одна промежуточная страница или не быть промежуточных страниц.

Цепочка должна быть кратчайшей, т.е. если существуют две цепочки, одна из которых содержит промежуточную страницу, а вторая нет, стройте цепочку без промежуточного элемента.

Основные теоретические положения.

- Библиотека `wikipedia` на `python` для работы со страницами Wikipedia;
- Функции ввода/вывода стандартной библиотеки `print()` и `input()`;

- Методы строк `strip()` и `split()` – первая убирает пробелы в начале и в конце строки, а вторая преобразовывает строку в массив строк по заданному разделителю;
- Метод списка `pop()` – удаляет значение списка с указанным индексом и возвращает его в качестве ответа;
- Логические конструкции `if else` – если условие `if` верно, то выполняется первый блок, в противном случае выполняется блок `else`;
- Метод словаря `keys()` – возвращает список всех ключей словаря;
- Цикл `for` – перебирает все значения в указанном диапазоне `range()`;
- Функция стандартной библиотеки `len()` – возвращает длину списка или строки;
- Функции библиотеки `Wikipedia` `set_lang` и `languages` – первая устанавливает язык сервиса, а вторая возвращает словарь всех возможных языков;
- Функция, библиотеки `wikipedia` `page()` – возвращает объект страницы по указанному названию;
- Метод класса `WikipediaPage` `summary()` – возвращает строку краткого содержания страницы;
- Атрибут класса `WikipediaPage` `title` – хранит название страницы;
- Конструкция `return` – используется для возвращения значения из функции;
- Функция стандартной библиотеки `str()` – присваивает объекту тип “string”;
- Атрибут класса `WikipediaPage` `links` – возвращает список с названиями всех ссылок на странице;

- Метод списков `append()` добавляет элемент в конец списка;
- Конструкция `break` – используется для выхода из цикла;

Выполнение работы.

1. На вход программы с помощью функции `input()` поступает строка. Полученная строка разделяется при помощи метода `split(",")` (предполагается, что названия страниц перечислены через запятую). Затем полученный массив обрабатывается при помощи генерации списков. К каждой строке применяется метод `strip()`. Список сохраняется в переменную `request`. Далее при помощи метода `pop()` из списка извлекается последний элемент (язык) и сохраняется в переменную `language`.

2. С помощью логической конструкции `if` проверяется наличие языка в списке доступных языков сервиса. Если условие ложно, то выводится сообщение “no results”. В противном случае выполнение программы продолжается.

3. На следующем этапе выполняется функция `find_max()`, принимающая на вход список названий страниц и возвращающая название страницы с максимальным количеством слов в кратком содержании и количество слов. Для этого внутри функции объявляется переменная `max_words`, которой присваивается начальное значение `-1` и переменная `page_title` со значением `""` (пустая строка). Далее используется цикл `for`, пробегающий через значения от `0` до значения длины списка (невключительно). В теле цикла мы получаем объект страницы по названию страницы с текущим индексом с помощью функции `page()` и выполняем проверку: больше ли количество слов в кратком содержании данной страницы чем значение `max_words`? Если да, то переменной `max_words` присваивается количество слов, а переменной `page_title` – название

текущей страницы. В последствии конечные значения `page_title` и `max_words` выводятся через запятую.

4. Для получения цепочки используется функция `get_chain()`. В теле данной функции объявляется список `chain` для хранения элементов цепочки. Далее используется цикл `for`. Данный цикл перебирает значения от 0 до значения, на 2 меньшего, чем длина списка. В теле цикла мы извлекаем два, идущих друг за другом, названия страниц по индексам и получаем по ним два объекта класса `WikipediaPage`, которые сохраняем в переменные `page_from` и `page_to`. Добавляем название первой страницы в список `chain`. Проверяем, содержится ли название второй страницы в списке ссылок первой. Если нет, то вновь используется цикл `for`. С его помощью мы обращаемся к каждому элементу в списке ссылок по индексу. Затем получается объект страницы и проверяется, входит ли название страницы с индексом `i+1` в список ссылок текущей страницы. Если да, то в список `chain` добавляется название это текущей страницы, и выполнение внутреннего цикла прекращается. После завершения основного цикла в список `chain` добавляется последний элемент списка названий страниц и возвращается список `chain`. Полученная цепочка выводится на экран.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Айсберг, IBM, ru	115 IBM ['Айсберг', 'Буран', 'IBM']	Программа работает правильно.
2.	Айсберг, IBM, dfcz	no results	Программа работает правильно.

Выводы.

Были изучены и применены основные управляющие конструкции Python.

Была разработана программа для определения названия страницы с максимальным количеством слов в кратком содержании и построения цепочки страниц.

Для первой подзадачи использовалась функция `set_lang()`.

Для второй подзадачи использовалась функция `find_max()`.

Для третьей подзадачи использовалась функция `get_chain()`.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import wikipedia

def is_page_valid(page):
    try:
        wikipedia.page(page)
    except Exception:
        return False
    return True

def find_max(pages):
    max_words = -1
    page_name = ''
    for i in range(len(pages)):
        page = wikipedia.page(pages[i])
        if len(page.summary.split()) >= max_words:
            page_name = page.title
            max_words = len(page.summary.split())
    return max_words, page_name

def get_chain(pages):
    chain = list()
    for i in range(0, len(pages)-1):
        page_from, page_to = wikipedia.page(pages[i]),
wikipedia.page(pages[i+1])
        chain.append(pages[i])
        if pages[i+1] not in page_from.links:
            for k in range(len(page_from.links)):
                current_page = wikipedia.page(page_from.links[k])
                if pages[i+1] in current_page.links:
                    chain.append(page_from.links[k])
                    break
        chain.append(pages[-1])
    return chain

# ---начало---
request = [x.strip() for x in input().split(',')]
```

```
language = request.pop(-1)

# подзадача 1
if language in wikipedia.languages().keys():
    wikipedia.set_lang(language)

    # подзадача 2
    max_words, page_title = find_max(request)
    print(str(max_words), page_title)

    # подзадача 3
    chain = get_chain(request)
    print(chain)

else:
    print('no results')
```