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 _____ Злобин А. С.

Преподаватель _____ Шевская Н. В.

Санкт-Петербург

2020

Цель работы.

Изучить основные управляющие конструкции Python и модуля Wikipedia API

Задание.

Используя вышеописанные инструменты, 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.

Основные теоретические положения.

В данной лабораторной работе были использованы следующие конструкции языка python:

- Встроенные функции Python:
  - input() - возвращает считывемое с консоли значение
  - print() - выводит на консоль принимаемое в качестве аргумента значение
  - len() - возвращает длину объекта, принятого в качестве аргумента
  - range() - возвращает последовательность чисел в заданном диапазоне с заданным шагом
- Встроенные методы Python:
  - str.split() - возвращает список всех слов в строке, используя значение аргумента в качестве разделителя (по умолчанию это пробел)
  - list.append() - добавляет переданный obj в существующий список.
  - str.strip() - удаляет се пробелы в начале и конце строки str
- Операторы
  - if: <последовательность действий 1> else: <последовательность действий 2>— если значение выражения после оператора if и перед двоеточием - true, выполняет блок кода с одинаковым уровнем отступа после if, если false – блок кода после else
  - in— если объект перед оператором является подстрокой или элементом объекта после оператора – значение выражения – true, в противном случае – false
  - break —прерывает выполнение цикла
  - return — используется в функциях для возвращения каких-либо значений
- Циклы

- or <переменная> in <итерируемый объект>:— для каждогозначения переменной, находящегося в итерируемом объекте,выполняет блок кода с одинаковым уровнем отступа после двоеточия

- Функции Wikipedia API:
  - page(title) – возвращает объект класса WikipediaPage, который представляет собой страничку сервиса Wikipedia, название которой - строка title
  - languages() –возвращает словарь, ключами которого являются сокращенные названия языков сервиса, а значениями – полные названия
  - set_lang(lang) – устанавливает язык lang как язык запросов в текущей программе
- Обращения к полям
  - page.summary— поле класса page модуля Wikipedia, возвращает многострочный литерал – краткое содержание страницы page
  - page.title — поле класса page модуля Wikipedia, возвращает строку – название страницы page
  - page.links—поле класса page модуля Wikipedia, возвращает список строк – названий страниц, ссылки на которые содержит страница page

Выполнение работы.

В самом начале программы необходимо импортировать модуль wikipedia.

Для того, чтобы приступить к решению поставленных задач, необходимо считать данные. Это осуществляется с помощью функции input() и метода split(). Результат (список строк) записываеися в переменную beg_str. Далее избавляемся от лишних пробелов в строках с помощью метода .strip(). Переменной language присвоим значение последней строки в списке beg_str,

так как последнее слово во входных данных это язык, и с помощью метода .pop() исключим эту сторку из списка.

1. Выполнение первой подзадачи осуществляется с помощью пользовательской функции is_language_valid(lang), которая на вход принимает строку (в данном случае language) и проверяет, есть ли такой ключ в словаре, который возвращает метод wikipedia.languages(). Устанавливает язык language как язык запросов с помощью функции wikipedia.set_lang(lang) и возвращает True, если есть, и False в противоположном случае.

2. Выполнение второй подзадачи осуществляется с помощью пользовательской функции max_outline(pages_names), которая на вход получает список строк, которые являются названиями страниц. В начале задаётся несколько переменных: size_of_page_max = 0 — максимальный размер краткого содержания(в начале равен 0), page_max_name = '' - название страницы с максимальным кратким содержанием(в начале равна пустой строке), size_of_page = 0 — размер страницы в текущей итерации цикла For.

В цикле for переменная i поочерёдно принимает каждое значение из списка pages_names. Для каждого значения определяет длинну краткого содержания страницы size_of_page с помощью функции len() и метода .split(), и если оно больше либо равно size_of_page_max, то size_of_page_max принимает значение size_of_page, а page_max_name принимает значение i. Функция возвращает список из двух элементов, первый из которых равен size_of_page_max, а второй — page_max_name.

3. Выполнение третьей подзадачи осуществляется с помощью пользовательской функции list_chain(pages_names), которая на вход получает список строк, которые являются названиями страниц. В начале создаётся список chain из одного элемента pages_names[0], и переменная quantity, равная длинне списка pages_names, уменьшенной на 1 (т. к. далее в цикле for где перебираются значения от 0 до quantity происходит обращение к следующему элементу списка)

В цикле for для каждой страницы, кроме последней создаётся список ссылок этой страницы first_links и с помощью оператора in проверяется, есть ли в этом массиве переменная pages_names[i+1] (содержится ли ссылка на следующую по списку страницу). Если есть, то к списку chain добавляется pages_names[i+1]. Если же нет, то для каждой ссылки из first_links проверяется, есть ли такая страница в wikipedia, и если да, то для этой страницы формируется список ссылок second_links, и проверяется, входит ли в этот список pages_names[i+1]. Если входит, то к списку chain добавляется pages_names[i+1] и цикл прерывается с помощью break.

Функция возвращает список chain.

Разработаный программный код см. в приложении А.

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 верно
2.	Айсберг, IBM, qwesa	no results	Программа работает верно

Выводы.

Были изучены основные управляющие конструкции Python и модуля Wikipedia API

Была написана программа, которая считывает данные с помощью функции input() и метода .split() и выводит результат с помощью функции print().

Первая подзадача была решена с помощью функции set_lang(lang).

Вторая подзадача была решена с помощью функции max_outline(pages_names), в которой с помощью цикла for и проверки условия if находится название страницы с самым длинным описанием.

Третья подзадача была решена с помощью функции list_chain(pages_names), которая с помощью циклов for и проверки условий if else проверяла, есть ли на одной странице ссылки на следующую (или есть, но через промежуточную страницу) и создавала список-цепочку этих страниц с помощью метода .append().

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

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


def is_language_valid(lang):
      if lang in wikipedia.languages():
            wikipedia.set_lang(lang)
            return True
      else:
            return False


def max_outline(pages_names):
      size_of_page_max = 0
      page_max_name = ''
      size_of_page = 0
      for i in pages_names:
            i_page = wikipedia.page(i)
            size_of_page = len((i_page.summary).split())
            if size_of_page >= size_of_page_max:
                  size_of_page_max = size_of_page
                  page_max_name = i_page.title
      return [size_of_page_max, page_max_name]


def list_chain(pages_names):
      chain = [pages_names[0]]
      quantity = len(pages_names)-1
      for i in range(quantity):
            first_links = wikipedia.page(pages_names[i]).links
            if pages_names[i+1] in first_links:
                  chain.append(pages_names[i+1])
            else:
                  for j in first_links:
                        if is_page_valid(j):
                              second_links                              =
(wikipedia.page(j)).links
                              if pages_names[i+1] in second_links:
                                    chain.append(j)
                                    chain.append(pages_names[i+1])
                                    break
      return chain


beg_str = input().split(',')
```

```
beg_str = [i.strip() for i in beg_str]
language = beg_str[len(beg_str)-1]
beg_str.pop()
if is_language_valid(language):
      res = max_outline(beg_str)
      print (res[0], res[1], sep = ' ')
      print (list_chain(beg_str))
else:
      print ("no results")
```