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

Студентка гр. 0382 _____ Тихонов С.В.

Преподаватель _____ Шевская Н.В.

Санкт-Петербург
2020

Цель работы.

Рассмотреть понятия парадигм программирования и освоить объектно-ориентированное программирование в Python на практике.

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.

Основные теоретические положения.

До настоящего времени мы использовали только интерактивный режим для запуска программ. Как уже упоминалось ранее, в Python можно сохранить весь программный код в файл и запустить его. Такой файл в языке Python называется модуль.

Функции модуля Wikipedia:

Функция	Описание	Возвращаемое значение
page(title)	Поиск страницы	Объект класса WikipediaPage, который представляет собой страничку сервиса Wikipedia, название которой - строка title
languages()	Поиск всех возможных языков сервиса	Словарь, ключами которого являются сокращенные названия языков, а значениями - названия.
set_lang (lang)	Установить зык lang, как язык запросов в текущей программе	None

Атрибуты класса WikipediaPage (страницы сервиса Wikipedia):

Поле класса	Описание	Возвращаемое значени
page.summary	Краткое содержание страницы page	Строка
page.title	Название страницы page	Строка
page.links	Список названий страниц, ссылки на которые содержит страница page	Список строк

Выполнение работы.

Ход решения:

В самом начале программы необходимо импортировать модуль wikipedia строкой import wikipedia.

Для начала следует считать входные данные. Их записываем в inp при помощи input() и метода .split(‘,’).

Далее берём inp[-1] и с помощью функции set_lang(inp[-1]) проверем, является ли inp[-1] одним из ключей в словаре языков сервиса полученного с помощью функции wikipedia.languages(). Если нет, то выводим “no results” и возвращаем „False’’.

Если данный язык является одним из языков сервиса, то при помощи wikipedia.set_lang(inp[-1]) устанавливается этот язык, как зык запросов в текущей программе, и возвращается значение „True’’

Если значение set_lang() - True, то выполнение программы продолжается и из списка входных данных при помощи метода inp.pop(-1) удаляется элемент с названием языка.

Далее для реализации второй подзадачи используется пользовательская функция max_summary(), принимающая список введённых названий страниц, в которой существуют переменные:

max_words – предназначена для хранения целого числа – количества слов в самом длинном кратком содержании страницы;

longest_page – предназначена для хранения сроки – названия страницы с максимальный количеством слов в кратком содержании.

В цикле for, количество итераций которого равно количеству введённых названий страниц, в каждой итерации оператором if с помощью функции len проверяется количество слов в кратком содержании очередной страницы, если оно больше текущего значения переменной max_words, это количество записывается в max_words. Функция возвращает кортеж из двух элементов: max_words, longest_page. Функцией print(max_summary[0], max_summary[1]) выводится результат.

Для решения третьей подзадачи реализована пользовательская функция make_chain(), принимающая список названий введённых страниц, в которой создаётся список ans, нулевым элементом которого является название первой страницы.

С помощью цикла for для каждого названия страницы кроме последнего создаётся переменная links, хранящая список ссылок этой страницы. Далее с помощью оператора in проверяется название следующей страницы на вхождения в

список links, если название входит в список, значит между страницами нет промежуточной, поэтому в конец списка ans добавляется название следующей страницы.

Если же название следующей страницы не входит в список links, то с помощью цикла for для каждой страницы из links пользовательской функцией is_page_valid() проверяется факт существования этой страницы, если страница существует, то создаётся список sub_links ссылок этой страницы и, если следующая страница списка inp входит в список sub_links, значит текущая страница из списка links является промежуточной, поэтому в конец списка ans сначала добавляется она, а потом — следующая страница из списка inp, после чего производится выход из цикла оператором break. .

Таким образом создаётся список страниц, в который если требуется, включены промежуточные страницы. Этот список и возвращает функция make_chain(). Результат выводится функцией print(make_chain(inp)).

Разработанный 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 правильно
2.	Айсберг, IBM, hh	no results	Программа работает правильно

Вывод.

В ходе работы были изучены основные управляющие конструкции языка Python и модуль wikipedia, разработана программа, устанавливающая язык запросов, выполняющая поиск максимального количества слов в кратком содержании страниц, названия которых вводятся с клавиатуры в одну строку, и строит список-цепочку из введённых страниц. Результат поиска и построенную цепочку выводит на экран.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb1.py

```
import wikipedia
 def is_page_valid(page): try:
            wikipedia.page(page) except
       Exception:
            return False re-
       turn True
 def set_lang(lg):
       if lg in wikipedia.languages(): wikipedi-
            a.set_lang(lg) return True
       else:
            return False
 def max_summary(pages):
       max_words = 0
       longest_page = ''
       for i in range(len(pages)):
         if len(wikipedia.page(pages[i]).summary.split()) >= max_words: max_words =
              len(wikipedia.page(pages[i]).summary.split())
                 longest_page = wikipedia.page(inp[i]).title return
       max_words, longest_page
 def make_chain(pages): ans =
       [pages[0]]
       for i in range(len(pages) - 1):
            links = wikipedia.page(pages[i]).links if pages[i + 1]
            in links:
                 ans.append((pages[i + 1])) else:
                 for j in range(len(links)):
                      if is_page_valid(links[j]):
                           sub_links = wikipedia.page(links[j]).links if pages[i + 1] in
                           sub_links:
                                ans.append(links[j])
                                ans.append(pages[i + 1]) break
       return ans
 inp = input().split(', ') if not
 set_lang(inp[-1]):
       print("no results") else:
       inp.pop(-1)
print(max_summary(inp)[0], max_summary(inp)[1])
print(make_chain(inp))
```