

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

Гудов Н.Р.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Научиться работать с базовыми управляющими конструкциями языка Python, изучить модуль Wikipedia API.

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. Предположим, нам на вход поступила строка: Айсберг, IBM, ru

В числе ссылок страницы с названием "Айсберг", есть страница с названием, которая содержит ссылку на страницу с названием "Буран", у которой есть ссылка на страницу с названием "IBM" -- это и есть цепочка с промежуточным звеном в виде страницы "Буран".

Гарантируется, что существует или одна промежуточная страница или ноль: т.е. в числе ссылок первой страницы можно обнаружить вторую. 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.

Функции модуля Wikipedia API:

`page(title)` – возвращает объект класса `WikipediaPage`, который представляет собой страничку сервиса Wikipedia,

`languages()` – возвращает словарь, ключами которого являются сокращенные названия языков, а значениями – названия.

`set_lang(lang)` - устанавливает язык `lang`, как язык запросов в текущей программе.

Атрибуты класса `WikipediaPage`:

`page.summary` – поле класса `page` модуля `Wikipedia`, которое возвращает строку, содержащую краткое содержание страницы `page`.

`page.title` – поле класса `page` модуля `Wikipedia`, которое возвращает строку, содержащую краткое содержание страницы `page`.

`page.links` – поле класса `page` модуля `Wikipedia`, которое возвращает список названий страниц, ссылка на которые содержит страница `page`.

Выполнение работы.

Входные данные записываются в список `inf`, после чего пользовательской функцией `setlang` определяется возможность установить предложенный язык или сразу завершить программу, предварительно выдав сообщение об отсутствии результата. В случае, когда программе удастся установить требуемый язык, следующим шагом является определение названия страницы, которой соответствует наибольшее из содержаний среди представленных страниц. Эту работу выполняет пользовательская функция `maxsum`, которая содержит в себе два параметра `words`, отвечающая за количество слов в кратком содержании и `trpage`, отвечающая за название страницы. Далее функция `makelist` составляет список уже имеющихся заголовков и вставляет между ними заголовки страниц,

находящихся в промежутке. Для этого проверяется следующий заголовок, точнее его связь с предыдущим. Гарантия наличия связи определяется предложенной функцией `is_page_valid`, далее проверяются и определяются промежуточные страницы, характеризующиеся своими заголовками.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	A-10, GAU-8, ru	79 Fairchild Republic A-10 Thunderbolt II ['A-10', 'GAU-8']	Работает правильно
2.	Москва, Россия, ru	346 Россия ['Москва', 'Россия']	Работает правильно

Выводы.

В ходе лабораторной работы были изучены основные управляющие конструкции языка Python и модуля Wikipedia API. Создана программа, отвечающая условиям задачи.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла menu1.py

```
import wikipedia

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

def makelist(pages):#3
    lst = [pages[0]]
    for i in range(len(pages) - 1):
        link = wikipedia.page(pages[i]).links
        if pages[i + 1] in link:
            lst.append((pages[i + 1]))
        else:
            for j in range(len(link)):
                if is_page_valid(link[j]):
                    slink = wikipedia.page(link[j]).links
                    if pages[i + 1] in slink:
                        lst.append(link[j])
                        lst.append(pages[i + 1])
                        break
    return lst

def maxsum(pages):#2
    words = -1
    mpage = ''
    for i in range(len(pages)):
        if len(wikipedia.page(pages[i]).summary.split()) >= words:
            words = len(wikipedia.page(pages[i]).summary.split())
            mpage = wikipedia.page(pages[i]).title

    return words, mpage

def setlang(lan):#1
    if lan in wikipedia.languages():
        wikipedia.set_lang(lan)
        return True
    else:
        return False

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

```
if setlang(inf[-1]):#1
    inf.pop(-1)
    print(maxsum(inf)[0], maxsum(inf)[1]) #2
    print(makelist(inf)) #3

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