

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: ТЕМА: Основные управляющие конструкции. Wikipedia API

Студентка гр. 0382

Охотникова Г.С.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Научиться работать с модулем Wikipedia, который позволяет программно работать с wiki-страничками сервиса Wikipedia, разделять программу на функции для решения отдельных подзадач и закрепить применение циклов.

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" выводит на экран это максимальное количество и название страницы, 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, между которыми может быть одна промежуточная страница или не быть промежуточных страниц.

Основные теоретические положения.

В лабораторной работе были использованы следующие управляющие конструкции языка, встроенные(библиотечные) структуры данных:

in, not in — операция проверки вхождения.

len() — определение длины строки.

for — цикл для итераций с известным числом повторений.

list.append() — добавление элемента в конец списка.

str.split() — разделяет строку по разделителю, разбиение происходит слева-направо. Возвращает список, состоящий из кусков строки.

Модуль *wikipedia* и его конструкции:

page(title) — возвращает объект класса *WikipediaPage*, который представляет собой страничку сервиса *Wikipedia*, название которой - строка *title*.

languages() — словарь, ключами которого являются сокращенные названия языков, а значениями - названия.

set_lang(lang) — устанавливает язык *lang*, как язык запросов в текущей программе.

page.summary — краткое содержание страницы *page*.

page.title — название страницы *page*.

page.links — список названий страниц, ссылки на которые содержит страница *page*.

Выполнение работы.

s — строка, принимаемая на вход.

res — отвечает за результат, который возвращает вторая функция.

lang() — функция, которая получает на вход исходную строку и проверяет, является ли ее последний элемент одним из языков сервера или нет и возвращает соответственно истину или ложь.

max_number_of_words() — функция, получающая на вход исходную строку. Ищет максимальное число слов в кратком содержании страницы,

возвращает это максимальное количество и название страницы, у которой оно обнаружилось. Если максимальных значений несколько, возвращает последнее. *max* — переменная, отвечающая за максимальное число слов, изначально равна нулю, так как с ней происходит сравнении количества слов в кратком содержании каждой страницы. *t* — переменная, которой присваивается название страницы с максимальным количеством слов в кратком содержании.

chain() — функция, получающая на вход исходную строку. Строит список-цепочку из страниц и возвращает полученный список. *l* — переменная, которая отвечает за итоговый список, изначально в *l* уже находится первая страница, так как с нее в любом случае начинается цепочка. Затем идет проверка, есть ли в ссылках первой страницы название второй страницы. Если да, то вторая страница добавляется в цепочку с помощью метода *list.append()*. Если нет, то проверяются ссылки каждой страницы, на которые есть ссылки у первой страницы. Тогда, если вторая страница нашлась в этих ссылках, то к цепочке добавляется название страницы, в ссылках которой оказалась вторая страница, и сама вторая страница. Далее цикл повторяется аналогично.

Если возвращенное значение первой функции *True*, то происходит вызов двух других функций и вывод результатов на экран, а если нет, то выводится сообщение '*no results*'.

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, Атмосфера, ru	119 Атмосфера [‘Айсберг’, ‘Абляция(гляциология)’, ‘Атмосфера’]	Выходные данные верны
2.	Книга, Автор, ru	95 Книга [‘Книга’, ‘Автор’]	Выходные данные верны
3.	Айсберг, Буран, ru	73 Айсберг [‘Айсберг’, ‘Буран’]	Выходные данные верны

Выводы.

Было изучено, как работать с модулем Wikipedia, который позволяет программно работать с wiki-страничками сервиса Wikipedia, разделять программу на функции для решения отдельных подзадач и закрепить применение циклов.

Была решена задача, на вход которой подается строка с названиями страниц. Были выполнен поиск страницы с максимальным количеством слов в кратком содержании и составлена цепочка из страниц с помощью конструкций модуля Wikipedia.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab1.py

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

def lang(s):
    if s[-1] in wikipedia.languages():
        wikipedia.set_lang(s[-1])
        return True
    else:
        return False

def max_number_of_words(s):
    max = 0
    for i in range(len(s)-1):
        if len(wikipedia.page(s[i]).summary.split()) >= max:
            max = len(wikipedia.page(s[i]).summary.split())
            t = wikipedia.page(s[i]).title
    return max, t

def chain(s):
    l = [s[0]]
    for i in range(len(s) - 2):
        if s[i+1] in wikipedia.page(s[i]).links:
            l.append(s[i+1])
        else:
            for j in wikipedia.page(s[i]).links:
                if s[i+1] in wikipedia.page(j).links:
                    l.append(j)
                    l.append(s[i + 1])
                    break
    return l

s = input().split(' ', ' ')
if lang(s):
    res = max_number_of_words(s)
    print(res[0], res[1])
    print(chain(s))
else:
    print('no results')
```