

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

Литягин С. М.

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 программе использовались следующие управляющие конструкции языка:

Встроенные функции языка Python:

- *input()* – функция ввода информации
- *print()* – функция вывода информации
- *len()* - возвращает целое число (int) равное количеству элементов объекта
- *range()* – возвращает числа в заданном промежутки (есть возможность задать шаг)

Встроенные методы Python:

- *str.split()* – возвращает список слов, разделенных аргументом
- *list.append()* – добавляет новый элемент в конец списка

Операторы:

- *if(<условие>): <действия_1> else: <действия_2>* - если выполняется заданное условие, то выполняется *действие_1*, в противном случае – *действие_2*
- *in* – если объект до оператора является элементом объекта после оператора, то возвращается True, иначе – False
- *break* – досрочно завершает цикл
- *return* – возвращение значения

Циклы:

- *for <переменная> in <итерируемый объект>: <действие>* - каждое значение переменной, находящейся в итерируемом объекте, выполняется *<действие>*

Функции модуля Wikipedia API:

- *page(title)* – возвращает страницу сервиса Wikipedia, название которой – *title*
- *languages()* – возвращает словарь, ключами которого являются сокращенные названия язык, а значениями – названия
- *set_lang(lang)* – устанавливает язык *lang*, как язык запросов в текущей программе

Поля модуля Wikipedia API:

- *page.summary* – возвращает краткое содержание страницы *page*
- *page.title* – возвращает название страницы *page*
- *page.links* – возвращает список названий страниц, ссылки на которые содержит страница *page*

Пользовательские функции:

- *def <наименование функции>(<аргументы>): <действия>* - при вызове данной функции в программе выполняются *<действия>*

Выполнение работы.

В самом начале программы подключаем модуль *wikipedia* строкой *import wikipedia*.

Затем в переменной *p* функцией *input().split(',')* формируется список, состоящий из подстрок входящих данных, разделенных запятой с пробелом методом *split(',')*.

Пользовательские функции, созданные для решения определенных задач в задании:

1. *language_exist(language)*. В данную функция поступает некоторый аргумент *language*. Далее функция производит проверку, является ли данный аргумент ключом для словаря *wikipedia.language()*. Если является, то функция командой *wikipedia.set_lang(language)* установит языком запросов *language* и вернет значение 1, иначе – 0.
2. *max_words(p)*. В данную функция поступает некий список *p*. Далее в функции создаются 2 переменные *max_w*, значением которой является количество слов в кратком содержании страницы *p[0]* (значение получается строкой *len(wikipedia.page(p[0]).summary.split())*), и *page*, названием которой является название страницы *p[0]* (значение получается строкой *wikipedia.page(p[0]).title*). Затем с помощью цикла *for i in range(len(p) - 1)* происходит поиск страницы, имеющей максимальное количество слов, среди всех страниц списка *p*. Это происходит путем сравнения значения переменной *max_w* и значения, полученного строкой *len(wikipedia.page(p[i]).summary.split())*. В случае, если полученное значение больше значения *max_w*, то переменная *max_w* получает значение *len(wikipedia.page(p[i]).summary.split())*, а переменная *page* получает значение *wikipedia.page(p[i]).title*. В конце функции возвращается значение переменных *max_w* и *page*.
3. *chain(p)*. В данную функцию поступает некий список *p*. Далее в функции создается список *a[p[0]]*, первым элементом которого является первый элемент списка *p*. Затем начинается цикл *for i in*

range(len(p)-2). В нем происходит следующее: создается список *links_first*, в котором хранятся ссылки страницы *p[i]* (значения получаются строкой *wikipedia.page(p[i]).links*). Затем среди этих ссылок ищется элемент *p[i+1]*. В случае того, что такая ссылка найдена, то в список *a* вносится элемент *p[i+1]*. В противном случае начинается цикл *for o in range(len(links_first))*. В нем сначала проверяется, существуют ли ссылки у страницы *links_first[o]* с помощью функции *is_page_valid(page)*. Если ссылки существуют, то проверяется, находится ли элемент *p[i+1]* в ссылках страницы *links_first[o]*. Когда такая ссылка найдена, то в список *a* вносится сначала значение *links_first[o]*, а затем значение *p[i+1]*. В результате функция вернет список *a*, в котором хранится цепочка из страниц.

После ввода данных программа проверяет функцией *language_exist(language)*, есть ли такой введенный язык в возможных языках сервисах. Если язык не найден, то переменная *k* получает значение 0, иначе – значение 1. В зависимости от значения переменной *k* программа либо завершает свою работу, выводя при этом “no results”, либо продолжает работу.

В случае продолжения работы происходит следующее: переменные *max_w* (максимальное количество слов) и *page* (наименование страницы) получают значения с помощью функции *max_words(p)*. Затем эти значения выводятся.

Кроме того выводится результат выполнения функции *chain(p)*.

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

Выводы.

В ходе работы были изучены основные управляющие конструкции языка Python и модуля Wikipedia API.

Разработана программа, выполняющая считывание входных данных с клавиатуры с помощью *input()* и *split()*.

Первая задача была решена функцией *language_exist()*, которая была реализована с помощью логических операторов *if(): else:* и *in*, а также функцией *wikipedia.set_language()*.

Вторая задача была решена функцией *max_w()*, которая была реализована с помощью цикла *for*, логического оператора *if(): else:*, функций *len()*, *wikipedia.page()*, методами *summary*, *title* и *split()*.

Третья задача была решена функцией *chain()*, которая была реализована с помощью цикла *for*, логических операторов *if(): else:* и *in*, функций *len()*, *append()* и *wikipedia.page()*, метода *links*.

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 max_words(p):
    max_w = len(wikipedia.page(p[0]).summary.split())
    page = wikipedia.page(p[0]).title
    for i in range(len(p) - 1):
        if len(wikipedia.page(p[i]).summary.split()) > max_w:
            max_w = len(wikipedia.page(p[i]).summary.split())
            page = wikipedia.page(p[i]).title
    return max_w, page

def language_exist(language):
    if language in wikipedia.languages():
        wikipedia.set_lang(language)
        return 1
    else:
        return 0

def chain(p):
    a = [p[0]]
    for i in range(len(p)-2):
        links_first = wikipedia.page(p[i]).links
        if p[i+1] in links_first:
            a.append(p[i+1])
        else:
            for o in range(len(links_first)):
                if is_page_valid(links_first[o]):
                    if p[i+1] in wikipedia.page(links_first[o]).links:
```

```

        a.append(links_first[0])
        a.append(p[i+1])
        break

    return a

p = input().split(", ")
k = language_exist(p[-1])
if k == 1:
    max_w, page = max_words(p)
    print(max_w, page)
    print(chain(p))
else:
    print("no results")

```