

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Программирование Python»
Тема: ОСНОВНЫЕ УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ. WIKIPEDIA API.

Студентка гр. 0382

Кривенцова Л.С.

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2020

Цель работы.

Изучить основы языка Python, модуль Wikipedia API и его функции посредством написания кода программы для решения конкретной задачи.

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.

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.

Модули.

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. Вы можете использовать существующие модули в своих программах, для этого вам необходимо их импортировать. Для примера рассмотрим модуль math, который содержит имена (т.е. переменные и функции) для работы с числами:

```
>>> import math
```

После импорта модуля вы можете использовать все имена, которые в нем определены, используя следующий синтаксис:

```
<имя_модуля>.<имя>
```

```
>>> math.pi 3.141592653589793
```

```
>>> math.sqrt(4) 2.0
```

● *pi* - это переменная, определенная в модуле *math*, которая хранит число *Пи*;

• `sqrt()` - это функция, определенная в модуле `math`, которая возвращает квадратный корень из неотрицательного числа. Также в Python есть синтаксис для импорта отдельных имен из модуля:

```
from <имя_модуля> импорт <имена>
```

После импорта таким способом вы можете использовать импортированные имена без указания имени модуля:

```
>>> from math import factorial
```

```
>>> factorial(4)24
```

Функции модуля `wikipedia`:

Функция	Описание	Возвращаемое значение
<code>page(title)</code>	Поиск страницы	Объект класса <code>WikipediaPage</code> , который представляет собой страничку сервиса <code>Wikipedia</code> , название которой - строка <code>title</code>
<code>languages()</code>	Поиск всех возможных языков сервиса	Словарь, ключами которого являются сокращенные названия языков, а значениями - названия. Например: <pre>>>> wikipedia.languages()['ru'] 'русский'</pre>
<code>set_lang(lang)</code>	Установить язык <code>lang</code> , как язык запросов в текущей программе.	<code>None</code>

Атрибуты класса WikipediaPage (страницы сервиса Wikipedia):

Поле класса	Описание	Возвращаемое значение
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:

Используется модуль Wikipedia API и его различные функции (*page()*, *languages()*, *set_lang()*).

На вход программы подаётся строка, в которой хранятся через запятую названия страниц, а в конце (последним элементом) сокращённая форму языка.

Чтобы установить язык запросов в текущей программе, функция *setlang(mainstring)* проверяет, есть ли в возможных языках сервиса язык с сокращённым названием последнего слова введённой строки. Если да, устанавливается язык с помощью функции *wikipedia.set_lang*. Если нет, программа печатает «*no results*» и завершается.

Чтобы найти максимальное число слов в кратком содержании страниц, функция *maximum(pages)* запускает цикл *for*, в котором каждая следующая страница сравнивается с предыдущей (сравниваются длины их кратких содержаний), и если длина её краткого содержания оказывается больше

сохранённого ранее максимума (или равно ему), то значение максимума перезаписывается (*max*), а также запоминается название этой страницы (в переменную *ttle*). В конце цикла в переменных *max*, *ttle* оказываются число слов наибольшего краткого содержания среди введённых страниц, и название этой страницы.

Чтобы построить цепочку, ведущую через введенные страницы, в главной функции организован цикл *for*. Он последовательно подаёт в функцию *chain(string1, string2)* по два названия страниц, между которыми нужно найти связь (прямую или через звено в ссылках). Затем, принимает из функции названия страниц, которые являются путём от одной страницы к следующей, и с помощью цикла *for* добавляет их в список результата.

Чтобы вернуть путь от одной страницы к следующей, функция создаёт по их названиям *spage1* и *spage2* - объекты класса *WikipediaPage*, который представляет собой странички сервиса *Wikipedia*. Чтобы построить путь, в переменную *links1* записывается список ссылок, содержащихся на странице *spage1*, и проверяет, есть ли среди них название страницы *spage2*. Если есть, переменной *rez* присваивается название страницы *spage2*. Если нет, запускается цикл *for*. Каждую итерацию значению *i* присваиваются ссылки страницы *spage1*. Проверяется, существует ли страница с таким названием с помощью функции *is_page_valid(page)*. Если нет, итерация завершается. Если да, то переменной *spage3* присваивается объект класса *WikipediaPage*, который представляет собой страничку сервиса *Wikipedia*. Если в ссылках этой страницы находится название страницы *spage2*, то переменной *rez* присваивается название страниц *spage3* и *spage2*, и цикл завершается с помощью *break*. Если нет, итерация завершается. Функция возвращает путь (списком), в главной функции обновляется переменная *result*. Когда цикл *for* в главной функции заканчивается, программа выводит результат *result*.

Переменные:

1. Главной функции:

mainstring; - список, который принимает на вход строку входных данных.

pages; - список названий страниц, в котором хранятся все элементы *vvodstrk*, кроме последнего (т. к. он обозначает язык).

result; - список, в котором хранится последовательность страниц, соединенных цепочкой (3 подзадача).

cycle; - переменная счётчик, которая используется в цикле *for*.

new; - переменная, которой присваивается список, возвращаемый функцией *spisoksslk*.

k; - переменная счётчик, которая используется во вложенном цикле *for*.

2.Переменные функции *maximum(pages)*:

ttle; - переменная, в которой хранится название страницы, имеющей наибольшее количество слов в кратком содержании.

max; - переменная, в которой хранится наибольшее количество слов в кратком содержании (среди введенных страниц).

length; - переменная, в которую каждую итерацию записывается длина (количество слов) краткого содержания текущей страницы.

i; - переменная счётчик для цикла *for*.

strng; - переменная, которой каждую итерацию присваивается объект класса *WikipediaPage*, который представляет собой текущую страничку сервиса *Wikipedia*.

3.Переменные функции *chain(string1, string2)*:

spage1; - объект класса *WikipediaPage*, который представляет собой страничку сервиса *Wikipedia* с названием, которое передаётся в функцию. Страница, от которой ищется связь(путь) к следующей.

spage2; - объект класса *WikipediaPage*, который представляет собой страничку сервиса *Wikipedia* с названием, которое передаётся в функцию. Страница, к которой ищется связь(путь).

rez; - список, который хранит путь от одной введенной страницы, к следующей.

links1; - список ссылок страницы *spage1*.

i; -переменная счётчик для цикла *for*.

spage3; - объект класса *WikipediaPage*, который представляет собой страничку сервиса *Wikipedia*. Страница, которая может являться звеном между *spage1* и *spage2*.

links3; -список ссылок страницы список ссылок страницы *spage3*.

Функции.

1. Главная.

Функция осуществляет ввод с клавиатуры строки исходных данных, вызывает функции *setlang(mainstring)* и *maximum(pages)*. Организует цикл для передачи данных в функцию *chain(string1, string2)*. Выводит на экран результат поиска цепочки — список *result*.

2. *is_page_valid(page)*.

Функция проверяет существование странички в *wikipedia* с таким названием. Принимает название страницы. Возвращает *True*, если *wiki*-страница с таким названием существует и *False* в ином случае.

3. *setlang(mainstring)*.

Принимает как аргумент строку — ***mainstring***, в конце которой хранится сокращённое название языка. Проверяет, есть ли такой язык в возможных языках сервиса, если нет, выводит строку "*no results*", возвращает значение *False* и завершает выполнение программы. В случае, если язык есть, устанавливает его как язык запросов в текущей программе и возвращает значение *True*.

4. *maximum(pages)*.

Функция выполняет поиск максимального числа слов в кратком содержании среди введенных страниц. Выводит на экран это максимальное

количество (переменная *max*) и название страницы (*title*), у которой оно обнаружилось. На вход подаётся аргумент:

pagess — список страниц, среди которых выполняется поиск.

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 верно
2.	Медведь, Лес, ru	114 Лес ['Медведь', 'Хищные', 'Лес']	Программа работает верно
3.	Книга, Писатель, rus	no results	Программа работает верно

Выводы.

Были изучены основы языка Python.

Разработана программа, работающая с модулем Wikipedia API, 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 введенных 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
    return True

def setlang(mainstring):
    if mainstring[len(mainstring) - 1] in wikipedia.languages():
        wikipedia.set_lang(mainstring[-1])
        return True
    else:
        return False

def maximum(pages):
    title = ''
    max = 0
    length = 0
    for i in pages:
        length = 0
        strng = wikipedia.page(i)
        length = len(strng.summary.split())
        if length >= max:
            max = length
            title = strng.title
    return max, title

def chain(string1, string2):
    spage1 = wikipedia.page(string1)
    spage2 = wikipedia.page(string2)
    rez = []
    links1 = []
    links1 = spage1.links
    if string2 in links1:
        rez.append(string2)
    else:
        for i in links1:
            if is_page_valid(i):
                spage3 = wikipedia.page(i)
                links3 = spage3.links
                if string2 in links3:
                    rez.append(i)
                    rez.append(string2)
                    break
    return(rez)

mainstring = input().split(' ', ' ')
pages = mainstring[0:len(mainstring)-1]
```

```
if setlang(mainstring)==False:
    print('no results')
else:
    print(maximum(pages)[0], maximum(pages)[1])

    result = []
    result.append(pages[0])

    for cycle in range(len(pages)-1):
        new = chain(pages[cycle], pages[cycle+1])
        if new == []:
            break
        else:
            for k in new:
                result.append(k)
    print(result)
```