

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Программирование»**  
**ТЕМА: ОСНОВНЫЕ УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ ЯЗЫКА PYTHON 3.**  
**WIKIPEDIA API**

Студент гр.0382

Диденко Д.В.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

## **Цель работы.**

Изучение базовых управляющих конструкций языка Python и модуля Wikipedia.

## **Задание.**

Используя вышеописанные инструменты, написать 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, вывести 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.**

Для выполнения данного задания потребовались следующие элементы языка Python:

Встроенные функции Python3:

input(), print() - для ввода и вывода информации соответственно;

len() - возвращает длину списка, строки или массива;

range() - генерирует ряд чисел в рамках заданного диапазона;

Функции Wikipedia API:

`page()` – выполняет поиск страницы

`languages()` – поиск всех возможных языков сервиса

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

Встроенные методы Python 3:

`split()` - разбивает строку с помощью указанного спецсимвола и возвращает список подстрок.

`pop()` – удаляет элемент списка по указанному индексу и возвращает его (в случае, когда индекс не указан, удаляет последний элемент списка).

`append()` – добавляет элемент в конец списка.

Атрибуты класса `WikipediaPage`:

`page.summary` – показывает краткое содержание страницы.

`page.title` – название страницы `page`.

`page.links` - список названий страниц, ссылки на которые содержит страница `page`.

Основные управляющие конструкции:

`for()`: – цикл.

`if:...elif:...else:` - условный оператор.

Операторы:

`in` – проверяет элемент на вхождение.

`not` – изменяет значение оператора на противоположное.

`return` – возвращает значение функции.

## **Выполнение работы.**

Исходный код решения задачи см.в приложении А.

В программе объявлены следующие функции:

`put_lang(lang)` — принимает в качестве аргумента строку (предполагается, что это краткая форма записи языка) и проверяет, есть ли эта строка в списке возможных языков википедии — `wikipedia.languages()`. Если есть — устанавливает как язык запросов в программе `wikipedia.set_lang(lang)` и возвращает значение «True», если же нет - «False».

*maxCW(nameTitle)* — принимает в качестве аргумента список. В теле функции объявляются переменные *maX* (для хранения максимального количества слов в кратком содержании среди всех страниц, содержащихся в переданном списке) и *title\_max* (для хранения названия этой страницы), изначальные значения — 0 и «». В цикле *for* перебираются наименования страниц, оператор *if* проверяет : если количество слов в кратком содержании текущей страницы больше, чем значение в переменной *maX*, то *maX* изменяет значение на новое, а *title\_max* изменяет значение на наименование этой страницы. Функция возвращает два значения — *maX, title\_max*.

*get\_chain(nameTitle)* - возвращает цепочку элементов между первым и последним элементами списка, принятого в качестве аргумента. Объявляется список *chain*, содержащий первый элемент переданного списка и булева переменная *sign*, содержащая значение *True*. В цикле *for* переменная *i* принимает значения индексов списка до предпоследнего (последний элемент не имеет смысла рассматривать), в теле цикла переменная *sign = True*, с помощью *if* проверяется, входит ли страница с индексом *i+1* (т. е. следующая) в список ссылок страницы с индексом *i*. Если да, к списку *chain* добавляется страница с индексом *i+1* и цикл совершает следующую итерацию. В противном случае в цикле *for* через переменную *j* рассматриваются страницы, ссылки на которые имеет страница с индексом *i*. Если *sign == True*, то , если элемент списка с индексом *i+1* входит в список ссылок страницы *j*, к списку *chain* добавляется страница *j* и переменная *sign* принимает значение *False*, чтобы не делать лишних итераций. После этого по такому же алгоритму проверяется каждый следующий элемент исходного списка, кроме последнего, т. к. за ним нет элемента и ссылку не имеет смысла искать.

Конструкция *nameTitle\_val = input().split(',')* принимает на вход строку, состоящую из наименований страниц и последнего слова – языка запросов, перечисленных через запятую и преобразует ее в список. С помощью функции *put\_lang()* проверяется, имеется ли указанный язык в списке возможных языков википедии, если нет – программа завершает свое выполнение и выводит в консоль “no results”. Иначе – указанный язык устанавливается в

качестве основного языка запросов в текущей программе. После этого язык удаляется из введенного списка за ненадобностью и программа выводит на экран через функцию `print` два результата работы функции `maxCW` – `maxCW(nameTitle_val)[0]` и `maxCW(nameTitle_val)[1]` содержащие информацию об максимальном количестве слов в кратком содержании страницы и название этой страницы. В следующей строке – результат работы функции `get_chain(nameTitle_val)` – список-цепочка от первого элемента списка до последнего.

### Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	Айсберг, IBM, ru	0115 IBM ['Айсберг', 'Буран', 'IBM']	Программа работает верно
2.	Айсберг, IBM, reter	no results	Программа работает верно

### Выводы.

Были изучены базовые управляющие конструкции языка Python 3 и некоторые функции и атрибуты Wikipedia API.

Разработана программа, которая принимает на вход список страниц википедии и язык, затем выводит на экран информацию о максимальном количестве слов в кратком содержании среди всех страниц и название этой страницы и список-цепочку от первой до последней страницы.

Для ввода и вывода информации использовались встроенные функции Python 3 — `input()` и `print()`.

Первая задача выполнена с помощью объявленной функции `put_lang()`, логических операторов `if:else:`, оператора вхождения `in` и функции `set_lang()` модуля `wikipedia API`.

Вторая — с помощью объявленной функции *maxCW()* и цикла *for* с логическими операторами *if:else:*, функции *len()*, метода *.split()* и поля класса *.summary* модуля *wikipedia API*.

Третья — с помощью объявленной функции *get\_chain()*, встроенных функций *range()*, *len()*; метода *.append()* и поля класса *.links*, цикла *for* с логическими операторами *if:else:*.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

**Название файла:** wiki.py

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

def put_lang(lang):
    if lang in wikipedia.languages():
        wikipedia.set_lang(lang)
        return True
    else:
        return False

def maxCW(nameTitle):
    maX = 0
    title_max = ''
    for i in nameTitle:
        if len(wikipedia.page(i).summary.split()) >= maX:
            maX = len(wikipedia.page(i).summary.split())
            title_max = wikipedia.page(i).title
    return maX, title_max

def get_chain(nameTitle):
    chain = [nameTitle[0]]
    sign = True
    for i in range(len(nameTitle)-1):
        sign = True
        if nameTitle[i+1] in wikipedia.page(nameTitle[i]).links:
            chain.append(nameTitle[i+1])
        else:
            for j in wikipedia.page(nameTitle[i]).links:
                if sign == True:
                    if nameTitle[i+1] in wikipedia.page(j).links:
                        chain.append(j)
                        chain.append(nameTitle[i+1])
                        sign = False
    return chain

nameTitle_val = input().split(', ')

if put_lang(nameTitle_val[-1]):
    nameTitle_val.pop()
    print(maxCW(nameTitle_val)[0], maxCW(nameTitle_val)[1])
    print(get_chain(nameTitle_val))
else:
    print("no results")
```