

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

Мукатанов А.В.

Преподаватель

Шевская Н.В.

Санкт-Петербург

2020

Цель работы.

Изучить основные управляющие конструкции языка Python и научиться работать с модулем 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.

Встроенные функции:

input() — считывает входные данные, возвращает строку

print() — выводит аргумент на консоль

len() — принимает на вход строку или список, возвращает целочисленную длину данного объекта

range() — создаёт список, содержащий арифметическую прогрессию с определенным шагом.

Функции модуля Wikipedia API:

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

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

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

3

Операторы:

`if: else:` - если значение выражения после оператора `if` - `true`, то выполняется блок кода в одинаковой табуляции после `if`, в случае `false` выполняется блок кода после `else:` `not` — инвертирует значение выражения `in` — если объект перед оператором является подстрокой или элементом объекта после оператора, то значение выражения — `true`, в противном случае — `false`

`break` — прерывает цикл

`continue` — цикл переходит на следующую итерацию

`return` — в функции возвращает значение

Методы:

`str.split(sep)` — метод класса `str`, принимает на вход разделитель (по умолчанию — пробел) и разбивает строку, к которой применён, на подстроки по разделителю и возвращает список этих подстрок.

`list.append()` — добавляет аргумент в конец списка `list`

Обращения к полям:

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

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

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

Выполнение работы.

Подключается модуль `wikipedia` с помощью инструкции `import wikipedia` . Далее мы считываем входные данные в переменную `tr` с использованием метода `split` для разделения строки на подстроки. Потом с помощью оператора `if` программа проверяет валидность языка, если проверка прошла , устанавливает

его основным и выполняет нижезаписанные подзадачи записанные , иначе программа выводит "по results" и прекращает выполнение.

Подзадачи:

- 1). В оператор `f` поступает значение функции `set_language(p[-1])`. Функция `set_language(language)` получает на вход строку `p[-1]`. И, если строка `p[-1]` является ключом для словаря `wikipedia.languages()`, то язык `p[-1]` устанавливается, как основной, а функция возвращает значение `True`, в противном случае возвращается значение `False`
- 2) Функция `max words(p)`, в которой с помощью оператора `if` и цикла `for` ищется страница с максимальным количеством слов в ее краткой записи. Функция возвращает максимальное количество слов в краткой записи и `title` этой страницы.
- 3) Функция `chain list(p)` принимает на вход список с именами страниц. После этого переменной `chain` присваивается элемент изначального массива с индексом 0.

Далее мы запускаем цикл `for` с нуля до количества элементов в массиве -1 (т.к мы присвоили нулевой элемент переменной `chain`). Потом мы присваиваем переменной `link` список ссылок страницы `p[i]` и проверяем, есть ли название следующей страницы в этом списке ссылок. Если есть, то мы добавляем `p[i+1]` в список `chain`. Если нет, то мы запускаем еще один цикл `for` , в котором мы проверяем валидность страницы из списка ссылок `link`, создаем новый список ссылок и проверяем, есть ли в этом списке элемент `p[i+1]`. Если есть, то мы добавляем промежуточную страницу `link[j]` и `p[i+1]` в список `chain`. Функция возвращает список `chain`.

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']	Верный ответ
2.	Айсберг, tBM, asd	no results	Верный ответ

Выводы.

В ходе выполнения работы были изучены основные управляющие конструкции языка Python и получен опыт работы с модулем Wikipedia API.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: cs_lb1.py

```
import wikipedia
```

```
def is_page_valid(page):
```

```
    try:
```

```
        wikipedia.page(page)
```

```
    except Exception:
```

```
        return False
```

```
    return True
```

```
def set_lang(language):
```

```
    if language in wikipedia.languages():
```

```
        wikipedia.set_lang(language)
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def max_words(p):
```

```
    max = -1001
```

```
    ti = "
```

```
    for i in range(len(p)):
```

```
        if len(wikipedia.page(p[i]).summary.split()) > max:
```

```
            max = len(wikipedia.page(p[i]).summary.split())
```

```
            ti = wikipedia.page(p[i]).title
```

```
    return max, ti
```

```
def chain_list(p):
```

```

chain = [p[0]]
for i in range(len(p) - 1):
    link = wikipedia.page(p[i]).links
    if p[i+1] in link:
        chain.append(p[i+1])
    else:
        for j in range(len(link)):
            if is_page_valid(link[j]):
                link_2 = wikipedia.page(link[j]).links
                if p[i + 1] in link_2:
                    chain.append(link[j])
                    chain.append(p[i+1])
                    break
return chain

```

```

p = input().split(' ')
if not set_lang(p[-1]):
    print('no results')
else:
    print(*max_words(p[:-1]))
    print(chain_list(p[:-1]))

```