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 _____ Куликов М.Д.

Преподаватель _____ Шевская Н.В

Санкт-Петербург

2020

Цель работы.

Изучение основных управляющих конструкций языка Python и модуля Wikipedia API.

Задание.

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 данной работе были использованы такие конструкции языка Python:

Функции:

print() - функция для вывода в консоль

len() - функция,позволяющая узнать длину списка или строки

input() - считывание данных с консоли, возвращает строку

range() - создает список чисел с определенным шагом

Методы:

str.split() - разбивает строку на подстроки по разделителю и возвращает список с подстроками

list.append() - добавляет элемент в конец списка

Операторы:

if: else: - если значение выражения после оператора if - true, то выполняется блок кода в одинаковой табуляцией после if, в случае false выполняется блок кода после else: 🎬

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

Обращение к полям:

page.summary – поле класса page модуля Wikipedia, которое возвращает строку, содержащую краткое содержание страницы page 🎬

page.title – поле класса page модуля Wikipedia, которое возвращает строку, содержащую краткое содержание страницы page 🎬

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

Выполнение работы.

В начале подключается библиотека wikipedia с помощью строки import wikipedia.

Потом считываются входные данные в переменную inp с использованием метода split для разделения строки на подстроки.

После этого проверяется валидность введенного пользователя языка и, в случае успешной установки языка, выполняем подзадачи с помощью написанных заранее функций (в случае ,если такой язык отсутствует, программа выведет на консоль «no results» и прекратит выполнение)

Функции для подзадач:

1) Функция set_lang(language) в которой проверяется валидность языка с помощью оператора if и возвращается и устанавливается этот язык для запросов. В случае успешной установки языка функция возвращает 0, в случае неудачного 1.

2) Функция max_words(inp) 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(inp) принимает на вход список с именами страниц. После этого переменной chain присваивается элемент изначального массива с индексом 0. Далее запускается цикл for с нуля до количества элементов в массиве -1 (т.к был присвоен нулевой элемент переменной chain). Далее присвается переменой links список ссылок страницы inp[i] и проверяется, есть ли название следующей страницы в этом списке ссылок. Если есть, то добавляется inp[i+1] в список chain. Если нет, то запускается еще один цикл for , в котором проверяется валидность страницы из списка ссылок links, создаем новый список ссылок и проверяем, есть ли в этом списке элемент

inp[i+1]. Если есть, то добавляется промежуточная страница links[j] и inp[i+1] в список chain. Функция возвращает список chain.

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']	Корректная работа программы
2.	Айсберг, IBM, 1231	no results	Корректная работа программы

Выводы.

В ходе работы были изучены основные управляющие конструкции языка Python и модуль wikipedia.

Для решения каждой подзадачи были написаны отдельные функции, которые в последствии были использованны в теле нашей программы и с помощью встроенной функции printf мы вывели нужные данные на экран.

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

def max_words(inp):
    max_amount = 0
    page = ''
    for i in range(len(inp)):
        if len(wikipedia.page(inp[i]).summary.split()) > max_amount:
            max_amount = len(wikipedia.page(inp[i]).summary.split())
            page = wikipedia.page(inp[i]).title
    return max_amount, page

def chain_list(inp):
    chain = [inp[0]]
    for i in range(len(inp) - 1):
        links = wikipedia.page(inp[i]).links
        if inp[i + 1] in links:
            chain.append((inp[i + 1]))
        else:
            for j in range(len(links)):
                if is_page_valid(links[j]):
                    trans_links = wikipedia.page(links[j]).links
                    if inp[i + 1] in trans_links:
                        chain.append(links[j])
                        chain.append(inp[i + 1])
                        break
    return chain

inp = input().split(', ')
if set_lang(inp[-1]) == 0:
    inp.pop(-1)
    print(max_words(inp)[0], max_words(inp)[1])
    print(chain_list(inp))
else:
    print('no results')
```