**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**](http://e.moevm.info/mod/quiz/view.php?id=76).

|  |  |  |
| --- | --- | --- |
| Студент гр. 1304 |  | Басыров В.А. |
| Преподаватель |  | [Берленко](https://etu.ru/ru/fakultety/fkti/sostav/kafedra-moevm/rukovodstvo-sostav-kafedry/berlenko-tatyana-andreevna) Т.А. |

Санкт-Петербург

2021

## Цель работы.

Изучение основных управляющих конструкция языка Python.Изучение модуля Wikipedia API,изучение основ функционального работа.Освоение работы с библиотеками.

## 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 больше ничего не делает. В случае, если язык есть, устанавливает его как язык запросов в текущей программе и выполняет еще два действия:

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 *(данный пример актуализирован к состоянию страниц wikipedia на 2021 год)*:

Айсберг, IBM, ru

В числе ссылок страницы с названием "Айсберг",  есть страница с названием , которая содержит ссылку на страницу с названием "1959 год", у которой есть ссылка на страницу с названием "IBM" -- это и есть цепочка с промежуточным звеном в виде страницы "1959 год".

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.

If<условие> :

<тело>

else:

<тело>-условный оператор

for <переменная счетчик> in range(леввое значение,правое значение>):  
 <тело> -цикл

w.page(<название страницы>)-поиск всех ссылок в википедии.

set\_lang(<язык>)-устанавливает язык запросов.

Методы википедии:

languages-все языки wikipedia

summary-подчсчет слов

title-заголовок страницы

links-все ссылки,содержащиеся на странице

print(),input()-операторы вывода и ввода

метод split(<разделитель>)-запись в список без разделителя.

Break-прекращение работы цикла.

del(индекс)-удаление элемента массива

## Выполнение работы.

Все функции выполнены в соответствии с уловием задачи.

Функция language(строка входных данных,ее длина)-функция проверка на наличие языка:

lang-возможный язык wikipedia

Функция max\_word\_in\_str(строка,ее длина)-функция нахождения максимального числа слов в кратком содержании страниц:

max- Максимальноечисло слов

tit-название страницы,содержащее максимальное число слов.

pager-страница википедии

words-промежуточное количество слов

Функция page\_chepochka(s,l)-функция,строющая цепочку из промежуточных звеньев и страниц:

s1-результирующая промежуточная строка

now-страница википедии

now\_links-список ссылок,содержащихся в странице.

l1-длина массива now\_links

middle\_page-все ссылки промежуточных страниц

Основная часть:

s-список,которую вводят

l-длина списка

a-максимальное число слов

b-название страницы

Разработанный программный код см. в приложении А.

Результаты тестирования см. в приложении Б.

## Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | Айсберг, IBM, ru | 115 IBM  ['Айсберг','1959 год', 'IBM'] | Пример из условия |
|  |  |  |  |
|  |  |  |  |
| 4 |  |  |  |

## Выводы.

Были изучены основные управляющие конструкции языка питон,а также модуль wikipedia,изучены основы функционального программирования.

Разработана программа, выполняющая считывание с клавиатуры исходных данных и выполняющая сразу несколько функция одновременно,такие как проверка языка википедии,поиск промежуточных ссылок и количество строк в каждой строке.Чтобы все реалзовать мне понадобились основные управляющие инструкции языки python,а также основные методы работы с википедией.

# Приложение А Исходный код программы

Название файла: 1304\_Basyrov

import wikipedia as w

def is\_page\_valid(title):

try:

w.page(title)

except Exception:

return False

return True

def language(s, l): # (1 подзадача)

lang = s[l - 1]

if not (lang in w.languages()):

return False

else:

return True

def max\_word\_in\_str(s, l): # 2 задача

max = 0 # Максимальное количество слов

tit=""

for i in range(l): # По всем указзанным в вводе страницам

pager = w.page(s[i]) # Страница

words = len(pager.summary.split())

if words >= max: # Проверка на максимум

max = words

tit = pager.title

return max, tit

def page\_chepochka(s,l):#3 задача

s1=[s[0]]

for i in range(l-1):

now=w.page(s[i])

now\_links=now.links

if not(s[i+1] in now\_links):

l1=len(now\_links)

for j in range(l1):

if is\_page\_valid(now\_links[j]):

middle\_page=w.page(now\_links[j]).links

if s[i+1] in middle\_page:

s1.append(now\_links[j])

break

s1.append(s[i+1])

return s1

s = list(input().split(', '))

l = len(s)

if language(s, l):

w.set\_lang(s[l - 1])

del s[l - 1]

l -= 1

a, b = max\_word\_in\_str(s, l)

print(a, b)

print(page\_chepochka(s,l))

else:

print("no results")