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

Студент гр. 9383 _____ Ноздрин В.Я.

Преподаватель _____ Размочаева Н.В.

Санкт-Петербург
2019

Цель работы.

Целью данной работы является изучение основных управляющих конструкций языка **Python**.

Задание.

Используя библиотеку ***wikipedia***, 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 и выводит на экран это максимальное количество и название страницы, 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. Каждая страница содержит в себе ссылку на следующий элемент списка-цепочки.

Выполнение работы.

В ходе решения представленных задач используется предложенная библиотека ***Wikipedia***.

Переменные:

user_input - Список, введенная пользователем строка, разбитая по ", "

lang - последний элемент списка пользовательского ввода, строка с короткой записью языка

wp_titles_list - часть пользовательского ввода с названиями используемых страниц

WikipediaPage_list - список ***wp_titles_list***, преобразованный в объекты типа ***WikipediaPage***

max_words_amount, max_words_pagename - наибольшее количество слов в статье и ее название

shortest_WikipediaPage_chain - искомая цепочка страниц

Функции:

try_set_lang() - функция ставит указанный язык, как язык работы с библиотекой ***Wikipedia*** возвращает ***True***, если язык существует и ***False***, если язык неверно указан и его не удалось поставить

list_max_words() - выполняет задачу 2 для списка объектов типа ***WikipediaPage***

make_chain() - выполняет задачу 3 для списка названий страниц

Также была использована функция ***is_page_valid()*** из преложенного модуля ***help_wiki_function***.

Разработанный программный код см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1	Айсберг, IBM, ru	115 IBM ['Айсберг', 'Буран', 'IBM']
2	Канада, Валюта ячё, Тенке, cv	32 Валюта ячё ['Канада', 'Валюта ячё', 'ISO 4217', 'Тенке']

5 Выводы.

Были изучены основные управляющие конструкции языка ***Python***. Были использованы условные операторы ***if-else*** и циклы ***for***.

Была разработана программа, выполняющая считывание с клавиатуры исходных данных и обрабатывающая их согласно условию. Для обработки данных использовались условные операторы ***if-else*** и циклы ***for***. Так же использовалась библиотека ***Wikipedia*** и модуль ***Help_wiki_function***.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
from wikipedia import *
from help_wiki_function import is_page_valid


def try_set_lang(language):
    if language in languages():
        set_lang(language)
        return True
    else:
        return False


def list_max_words(arg_WikipediaPage_list):
    max_summary_words_amount = 0
    max_summary_words_pagename = ''
    for a_page in arg_WikipediaPage_list:
        words_amount = len(a_page.summary.split())
        if words_amount >= max_summary_words_amount:
            max_summary_words_amount = words_amount
            max_summary_words_pagename = a_page.title
    return max_summary_words_amount, max_summary_words_pagename


def make_chain(wp_list):
    result_chain = [wp_list[0]]
    for i in range(len(wp_list)-1):
        if wp_list[i+1] in page(wp_list[i]).links:
            result_chain.append(wp_list[i+1])
        else:
```

```
                for intermediate in page(wp_list[i]).links:
                    if is_page_valid(intermediate):
                        if wp_list[i+1] in page(intermediate).links:
                            result_chain.append(intermediate)
                            result_chain.append(wp_list[i+1])
                            break
            return result_chain


        # if __name_ == "__main__":
        user_input = input().split(', ')
        lang = user_input[-1]
        if try_set_lang(lang):
            wp_titles_list = user_input[:-1]
            WikipediaPage_list = map(page, wp_titles_list)
                                    max_words_amount,      max_words_pagename      =
list_max_words(WikipediaPage_list)
            shortest_WikipediaPage_chain = make_chain(wp_titles_list)
            print(max_words_amount, max_words_pagename)
            print(shortest_WikipediaPage_chain)
        else:
            print('no results')
```