

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Введение в ИТ»**  
**Тема: «Основные управляющие конструкции»**

Студент гр. 9303

Максимов Е.А.

Преподаватель

Берленко Т. А.

Санкт-Петербург

2019

## **Цель работы.**

1. Изучить основные управляющие конструкции языка Python;
2. Изучить функции модуля «Wikipedia».

## **Задание.**

Используя инструменты модуля 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 "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.**

1. Управляющие конструкции – специальные конструкции Python, которые позволяют управлять порядком вычислений в программе.

Простейшей из управляющих конструкций является условный оператор *if-elif-else*. Его синтаксис:

```
if <условие 1>:  
    <действие 1>  
elif <условие 2>:  
    <действие 2>  
else:  
    <действие 3>
```

Данная конструкция проверяет истинность набора условий и в зависимости от того, какие из них выполняются, переводит управление к соответствующим блокам кода. В случае, если не выполнено *<условие 1>*, то проверяется *<условие 2>*, если *<условие 2>* также не выполнено, то выполняется *<действие 3>*. Условием может быть любое выражение, которое можно привести к типу *bool*.

2. Для выполнения повторяющихся действий используются операторы циклов. В Python это цикл для перебора значений *for* и цикл с условием *while*.

Синтаксис цикла *for*:

```
for <переменная> in <коллекция>:  
    <действие 1>  
else:  
    <действие 2>
```

Здесь *<переменная>* участвует в функции *in*, которая используется для итерирования объектов из элементов *<коллекция>*, в качестве которой могут выступать списки, функция *range()* и др.

Синтаксис цикла *while*:

```
while <условие>:  
    <действие 1>  
else:  
    <действие 2>
```

Здесь блок *<действие 1>* выполняется до тех пор, пока *<условие>* не станет ложно, тогда выполнится *<действие 2>*.

3. Метод *split()* позволяет разбить строку на части, возвращая список элементов-строк, полученный путем деления исходной строки по символу или строке, переданному в качестве аргумента в метод. Сам разделитель при этом в эти строки не входит.

Синтаксис метода `split`:

```
<строка>.split(<разделитель>)
```

Если разделитель не указан, то аргументом метода по умолчанию будут пробельные символы.

4. Списки – изменяемый тип данных, представляющий собой упорядоченную последовательность элементов произвольного типа. Списки в Python обозначаются квадратными скобками. К данному типу относятся некоторые методы и функции Python.

5. Wikipedia – модуль Python, разработанный для взаимодействия со страницами Wikipedia.

В модуле Wikipedia используются следующие функции:

*page(title)* – объект класса `WikipediaPage`, который представляет собой страничку сервиса Wikipedia, название которой - строка *title*;

*languages()* – словарь, ключами которого являются сокращенные названия языков, а значениями – названия;

*set\_lang(lang)* – установить язык *lang*, как язык запросов в текущей программе.

В модуле Wikipedia представлены атрибуты класса `WikipediaPage`:

*page.summary* – краткое содержание страницы *page*;

*page.title* – название страницы *page*;

*page.links* – список названий страниц, ссылки на которые содержит страница *page*.

**Выполнение работы.**

В программе использовались следующие переменные:

1. Целого типа (integer):

- a. *i* – переменная-счётчик, участвующая в цикле проверки ссылок, в поиске ссылки с наибольшим количеством слов в кратком описании страницы и в создании списка-цепочки;
- b. *j* – переменная-счётчик, участвующая в цикле создания списка-цепочки;
- c. *max\_words* – максимальное количество слов среди кратких описаний страниц;
- d. *words* – количество слов в кратком содержании страницы текущей итерации;
- e. *max\_number* – номер страницы в списке, в которой максимальное количество слов в её кратком содержании.

2. Списки (list):

- a. *pages* – список ссылок, которые были указаны пользователем и язык запросов программы (в дальнейшем язык запросов удаляется среди элементов списка);
- b. *chain* – список-цепочка, создаваемый из списка страниц, указанных пользователем, и промежуточных страниц, если такие имеются;
- c. *links* – список страниц, на которые ведёт страница из *pages* в данной итерации (ссылки первого порядка);
- d. *sublinks* – список страниц, на которые ведёт страница из *links* в данной итерации (ссылки второго порядка).

Программа работает согласно алгоритму ниже:

1. Импортируется модуль *wikipedia*, функция *is\_page\_valid* из файла *help\_wiki\_function*. Считываются данные, которые вводит пользователь – строка, представляющая собой список ссылок и название языка;

данная строка сразу преобразовывается к списку *pages* с помощью метода с аргументами *split(, )*;

2. Программа использует последний элемент списка *pages* в качестве языка запросов программы и проверяет наличие данного языка среди элементов словаря, который вызывается с помощью функции *wikipedia.languages()*. Если введенный пользователем язык присутствует в словаре, то он устанавливается в качестве языка запросов программы с помощью функции *wikipedia.set\_lang()* и удаляется из списка *pages* с посредством инструкции *del*, иначе на экран выводится сообщение “*no results*” и работа программы немедленно завершается с помощью функции *exit()*.

3. С помощью цикла *for* и импортированной функции *is\_page\_valid* проверяются по всей длине списка с помощью функции *range* ссылки *pages* на корректность.

Функция *is\_page\_valid* представляет собой обработку исключения *Exception* посредством конструкции *try-except*. В случае возникновения исключения функция возвращает булево значение *False*.

В случае, если одна из ссылок некорректна, на экран выводится сообщение “*no results*” и работа программы немедленно посредством функции *exit()*.

4. Производится поиск страницы с максимальным количеством слов в кратком описании с помощью цикла *for* по всей длине списка *pages* посредством функции *range*. Переменная *words* вызывается для каждого элемента списка с помощью добавления атрибута класса *summary* и метода *split*. Переменные *max\_words* и *max\_number* обновляются, если переменная *words* больше или равно *max\_words*. После цикла переменная *max\_number* используется для определения элемента списка *pages*. Далее выводится значение *max\_words* и название страницы с наибольшим количеством слов в кратком описании посредством функции *page* и класса *title*.

5. Производится создание списка-цепочки. Поскольку первый и последний элементы списка однозначно определены по условию, и ими являются первый и последний элементы списка *pages*, то списку *chain* присваивается значение первого элемента списка *pages*.

Далее программа на всей длине списка с помощью функции *range*, кроме последнего элемента, проверяет наличие следующего элемента среди списка ссылок текущего элемента ( $len(pages)-1$ , т.к. у последнего элемента нет следующего), который присвоен списку ссылок первого порядка *links* посредством функции *page*.

Если он присутствует, то в список-цепочку *chain* добавляется искомый элемент с помощью метода списков *append* и программа переходит к следующей итерации посредством *continue*.

Если он отсутствует, тогда программа с помощью цикла *for* для каждого элемента списка *links* создаёт список ссылок второго порядка – *sublinks*. Теперь программа ищет проверяемый элемент среди списка ссылок второго порядка. Если программа не находит его среди ссылок второго порядка, то цикл переходит к следующей ссылке первого порядка. Это продолжается до тех пор, пока не обнаруживается искомый элемент среди ссылок второго порядка. В этом случае программа добавляет итерируемую ссылку первого порядка и искомый проверяемый элемент в список-цепочку *chain* с помощью метода списков *append*, и программа выходит из внутреннего цикла во внешний посредством инструкции *break*.

В результате программа печатает полученный список-цепочку *chain*.

Разработанный программный код см. в приложении А.

**Результаты тестирования.**

№ п/п	Входные данные	Выходные данные	Комментарии
1.	2002, 2003, qwerty	no results	Тест пройден
2.	Канада, Валюта ячѐ, Тенке, cv	32 Литас [“Канада”, “Валюта ячѐ”, “ISO 4217”, “Тенке”]	Тест пройден
3.	Айсберг, IBM, ru	115 IBM [“Айсберг”, “Буран”, “IBM”]	Тест пройден

### **Выводы.**

В ходе лабораторной работы были изучены: структура программы языка программирования Python, основные управляющие конструкции, модуль Wikipedia. Была разработана программа, которая принимает на вход исходные данные в виде строки, состоящей из заголовков страниц и языка запросов, проверяет их на корректность, и в случае успешной проверки выводит число, обозначающее максимальное количество слов среди корректных страниц, её название и список-цепочку из страниц согласно условию задания. Для обработки данных использовались метод строк *split()*, метод списков *append()*, условные операторы *if-else*, операторы циклов *while*, *for*, функции и методы модуля Wikipedia.



## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файлов: main.py, help\_wiki\_function.py

#### 1. Файл main.py:

```
import wikipedia
from help_wiki_function import is_page_valid
pages = input().split(', ')

if pages[-1] in wikipedia.languages():
    wikipedia.set_lang(pages[-1])
    del pages[-1]
else:
    print('no results')
    exit()

for i in range(len(pages)):
    if not is_page_valid(i):
        print('no results')
        exit()

max_words, max_number = 0, 0
for i in range(len(pages)):
    words = len(wikipedia.summary(pages[i]).split())
    if words >= max_words:
        max_words = words
        max_number = i
print(max_words, wikipedia.page(pages[max_number]).title)

chain = [pages[0]]
for i in range(len(pages) - 1):
    links = wikipedia.page(pages[i]).links
    if pages[i + 1] in links:
        chain.append(pages[i + 1])
        continue
    else:
        for j in range(len(links)):
            sublinks = wikipedia.page(links[j]).links
            if pages[i + 1] in sublinks:
                chain.append(links[j])
                chain.append(pages[i + 1])
                break
print(chain)
```

#### 2. Файл help\_wiki\_function.py:

```
import wikipedia

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