ФГБОУ ВО “Чувашский государственный университет им. И. Н. Ульянова”.

Факультет: Информатики и вычислительной техники.

Кафедра: Вычислительной техники.

Кросс-платформенные средства разработки программного обеспечения

**Лабораторная Работа №5**

Выполнил: студент группы ИВТ-41-18

Дружинин Артемий Константинович

Проверил: доцент Ковалев Сергей Васильевич

Чебоксары 2021 г.

**Задание**

В рамках лабораторной работы необходимо модифицировать программу из предыдущей лабораторной работы, которая предназначалась для помощи в изучении иностранного языка. Программа должна быть разбита на функции с целью улучшения читаемости кода. Кроме того, программа должна использовать модуль nltk (https://www.nltk.org/) для того, чтобы предоставить в качестве примера не весь абзац, а отдельное предложение. К программе предъявляются следующие требования:

1. Поиск примеров использования осуществляется во внешнем txt-файле, который должен быть представлен реальной книгой на любом иностранном языке.

2. Необходимо реализовать функцию для считывания содержимого файла и разбиения его на отдельные предложения. В качестве параметра функции выступает имя файла, результатом работы является список строк, где каждая строка содержит отдельное предложение.

3. Разбиение исходного текста на предложения осуществляется с помощью модуля nltk.

4. При необходимости, для улучшения логики работы программы и её читаемости, допускается создание дополнительных функций, например, функции, для проверки того, что какая-либо строка содержит все необходимые слова.

5. Параметры всех функций, а также тип возвращаемого значения для каждой функции, должны быть аннотированы.

6. Параметры поиска задаются во внешнем файле «request.json», который имеет туже структуру, что и в предыдущей лабораторной работе.

7. Результат работы необходимо сохранить в файле «response.json». Данный файл должен быть jsonпредставлением, структуру которого необходимо проработать самостоятельно.

**Полный текст программы**

import json  
  
from nltk import sent\_tokenize  
from setuptools.\_vendor.ordered\_set import is\_iterable  
  
path\_json\_out = r"D:\Study\University\Sem 7\КПРС ПО\Лабораторные\5\response.json"  
  
path\_json\_in = r"D:\Study\University\Sem 7\КПРС ПО\Лабораторные\5\request.json"  
  
  
def flatten(l):  
 """  
 Преобразование из списка списков в список  
 :param l: список списков  
 :return: список  
 """  
 if is\_iterable(l):  
 flat = []  
 for i in l:  
 if is\_iterable(i):  
 flat.extend(flatten(i))  
 else:  
 flat.append(i)  
 return flat  
 return [l]  
  
  
def open\_file(filename):  
 """  
 Чтение данных с текстового файла  
 :param filename: текстовый файл  
 :return: список, разбитый на предложения  
 """  
 with open(filename, encoding='utf-8') as file:  
 a = [sent\_tokenize(i) for i in file.readlines()]  
 return flatten(a)  
  
  
def read\_json(filename\_json):  
 """  
 Чтение данных с json файла  
 :param filename\_json: json файл  
 :return: словарь данных из файла  
 """  
 with open(filename\_json) as json\_file:  
 return json.load(json\_file)  
  
  
def search\_words():  
 """  
 Основная функция  
 :return:   
 """  
 json\_obj = read\_json(path\_json\_in)  
 s = json\_obj["file name"]  
 s = "D:/Study/University/Sem 7/КПРС ПО/Лабораторные/5/" + s  
 file\_read = open\_file(s)  
  
 proposal = []  
  
 for str\_read in file\_read:  
 if json\_obj["example minimum length"] <= len(str\_read) <= json\_obj["example maximum length"]:  
 fl\_search = True  
 for words in json\_obj["words"]:  
 if words not in str\_read:  
 fl\_search = False  
 break  
 if fl\_search:  
 proposal.append(str\_read)  
 if len(proposal) == json\_obj["number of examples"]:  
 break  
  
 # Запись в файл  
 with open(path\_json\_out, 'w') as json\_file:  
 if len(proposal) != 0:  
 cnt = 1  
 data = {}  
 for str\_s in proposal:  
 data[cnt] = str\_s  
 cnt += 1  
 json.dump(data, json\_file, indent=4, ensure\_ascii=False)  
 else:  
 json.dump({"Result": "The search failed"}, json\_file, indent=4)  
  
  
search\_words()