

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное учреждение высшего образования  
«Чувашский государственный университет И.Н. Ульянова»  
Факультет информатики и вычислительной техники  
Кафедра вычислительной техники

Кросс-платформенные средства разработки программного обеспечения  
Лабораторная работа 4  
«Работа с файлами и json-данными»

**Выполнил:**

Студент группы ИВТ-41-20  
Галкин Д.С.

**Проверил:**

Ковалев С.В.

## Цель работы:

1. Получение навыков работы с файлами с использованием языка Python
2. Ознакомление с форматами json для представления данных
3. Получение практических навыков использования json-данных в python-скриптах

### Задание для выполнения

В рамках лабораторной работы необходимо написать программу для помощи в изучении иностранного языка. Программа позволяет найти в какой-либо книге, представленной отдельным текстовым файлом, примеры использования иностранных слов.

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. Параметры поиска задаются во внешнем файле «request.json», который должен иметь, как минимум, следующую структуру

```
{
  "file name": "Harry Potter and the Sorcerer.txt", "words": [
    "Harry",
    "Ron",
    "Hermione"
  ],
  "example minimum length": 20,
  "example maximum length": 200,
  "number of examples": 5
}
```

4. Программа должна найти в тексте абзац требуемого размера, который все необходимые слова. Данный абзац и будет считаться примером использования
5. Программа должна найти требуемое кол-во примеров и записать работы в файле «response.json». Данный файл должен быть json-представлением, структуру которого необходимо проработать самостоятельно
6. В python-скрипте при работе со всеми файлами необходимо использовать оператор with

## Полный текст программы:

```
import re
import json

class Document:
    def __init__(self, file_name, target_words, example_minimum_length,
example_maximum_length, number_of_examples):
        self.file_name = file_name
        self.target_words = target_words
        self.example_minimum_length = example_minimum_length
        self.example_maximum_length = example_maximum_length
        self.number_of_examples = number_of_examples

# Функция, которая считывает параметры поиска из файла "request.json"
def read_json(filename):
    with open(filename, "r") as request_file:
        return json.load(request_file)

# Функция, которая конфигурирует параметры
def config_document(request_data):
    return Document(**request_data)

# Функция, которая читает текст с файла книги
def read_text(filename):
    with open(filename, "r", encoding="utf-8") as book:
        return book.read()

# Функция, которая записывает результат в response.json
def write_json(filename, data):
    with open(filename, "w", encoding="utf-8") as response:
        json.dump(data, response, ensure_ascii=False, indent=4)

    print("Примеры использовани найдены и сохранены в файле 'response.json!")

# Функция, которая ищет слова представленные с json
def find_words(text, document, filename):
    examples = []
    paragraph_pattern = r"(.*)[.!?]\s+"

    for word in document.target_words:
        word_examples = re.findall(paragraph_pattern, text, re.DOTALL)
        word_examples = [example.strip() for example in word_examples
            if document.example_minimum_length <= len(
                example) <= document.example_maximum_length and word in example]
        examples.extend(word_examples[:document.number_of_examples])

    response_data = {
        "examples": examples
```

```
}
```

```
write_json(filename, response_data)
```

```
r_filename = "request.json"
```

```
w_filename = "response.json"
```

```
document = config_document(read_json(r_filename))
```

```
book = read_text(document.file_name)
```

```
find_words(book, document, w_filename)
```