Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №4**

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант №2

Группа: P3112

Выполнил: Балин А. А.

Проверил: к. т. н., преподаватель Белозубов А. В.

Оглавление

[Введение 3](#_Toc120022551)

[Порядок выполнения работы 4](#_Toc120022552)

[Выполнение заданий 5](#_Toc120022553)

[Заключение 16](#_Toc120022554)

[Список литературы 17](#_Toc120022555)

# Введение

В этой лабораторной работе я хочу научиться переводить данные из одного формата в другой разными способами. Также хочу сравнить разницу во времени между разными алгоритмами конвертации.

# Порядок выполнения работы

1. Определить номер варианта как остаток деления на 36 порядкового номера в списке группы в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.
2. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного. При этом необходимо, чтобы в выбранном дне было не менее двух занятий (можно использовать своё персональное). В случае, если в данный день недели нет таких занятий, то увеличить номер варианта ещё на восемь.
3. Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый (нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов).
4. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать. Сравнить полученные результаты и объяснить их сходство/различие.
5. Переписать исходный код, добавив в него использование регулярных выражений. Сравнить полученные результаты и объяснить их сходство/различие.
6. Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить стократное время выполнения парсинга + конвертации в цикле. Проанализировать полученные результаты и объяснить их сходство/различие.
7. Переписать исходную программу, чтобы она осуществляла парсинг и конвертацию исходного файла в любой другой формат (кроме JSON, YAML, XML, HTML): PROTOBUF, TSV, CSV, WML и т.п. Проанализировать полученные результаты, объяснить особенности использования формата.
8. Проверить, что все пункты задания выполнены и выполнены верно.

# Выполнение заданий

№1

Мой порядковый номер в списке: 2, остаток от деления 2 на 36: 2, следовательно, номер моего варианта: 2

№2

Для моего варианта исходный формат: JSON, результирующий формат: YAML, день недели: понедельник

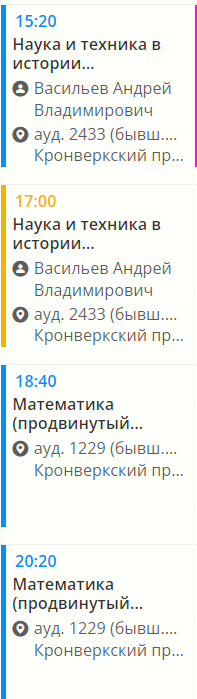


Рисунок 1. Моё расписание на понедельник

Сформируем объект JSON на основе расписания (Рисунок 1):

{

    "Первая пара": {

        "Название": "Наука и техника в истории цивилизации",

        "Преподаватель": "Васильев Андрей Владимирович",

        "Аудитория": "2433",

        "Время начала": "15:20",

        "Время окончания": "16:50",

        "Корпус": "Кронверкский проспект, 49",

        "Тип": "Лекция"

    },

    "Вторая пара": {

        "Название": "Наука и техника в истории цивилизации",

        "Преподаватель": "Васильев Андрей Владимирович",

        "Аудитория": "2433",

        "Время начала": "17:00",

        "Время окончания": "18:30",

        "Корпус": "Кронверкский проспект, 49",

        "Тип": "Практика"

    },

    "Третья пара": {

        "Название": "Математика (продвинутый уровень)",

        "Преподаватель": "Абжандадзе Зураб Леванович",

        "Аудитория": "1229",

        "Время начала": "18:40",

        "Время окончания": "20:10",

        "Корпус": "Кронверкский проспект, 49",

        "Тип": "Лекция"

    },

    "Четвертая пара": {

        "Название": "Математика (продвинутый уровень)",

        "Преподаватель": "Абжандадзе Зураб Леванович",

        "Аудитория": "1229",

        "Время начала": "20:20",

        "Время окончания": "21:50",

        "Корпус": "Кронверкский проспект, 49",

        "Тип": "Лекция"

    }

}

Этот объект будет лежать в файле Monday.json

№3

Код программы:

f = open('./Monday.json', 'r', encoding='utf-8')

F = f.readlines()

f.close()

s = ''

for S in F:

    s += S.strip()

result = ''

tab = -1

i = 0

name = True

mas = False

mastab = [-3]

while i < len(s):

    if mastab[-1] == tab:

        name = False

    if s[i] == '[':

        i += 1

        mas = True

        if mastab[-1]==-3:

            mastab[-1] = tab

        else:

            mastab.append(tab)

        name = False

        result += '\n'

    if s[i] == '{':

        tab += 1

        if not(mas and tab - mastab[-1]<2):

            name = True

        i += 1

        if tab != 0:

            if result[-1] != '\n':

                result += '\n'

    elif s[i] == '\"':

        if name:

            t = ''

            i += 1

            while s[i] != '\"':

                t += s[i]

                i += 1

            i += 1

            if mas and tab == mastab[-1]:

                result += "  "\*tab + '- ' + t

            else:

                result += "  "\*tab + t

            name = False

        else:

            t = ''

            i += 1

            while s[i] != '\"':

                t += s[i]

                i += 1

            i += 1

            q = True

            for x in t:

                if x.isalpha():

                    q = False

                    break

            if q:

                if mas and mastab[-1] == tab:

                    if result[-1] != '\n':

                        result += '\n'

                    result += mastab[-1]\*'  '+'- '+"\'"+t+"\'"

                else:

                    result += "\'"+t+"\'"

                    name = True

            else:

                if mas and tab - mastab[-1]<2:

                    if result[-1] != '\n':

                        result += '\n'

                    result += mastab[-1]\*'  '+'- '+t

                else:

                    result += t

                    name = True

    elif s[i] == ':':

        result += ': '

        i += 1

    elif s[i] == ',':

        result += '\n'

        i += 1

    elif s[i] == '}':

        tab -= 1

        i += 1

    elif s[i] == ']':

        if len(mastab)==1:

            mas = False

        i += 1

        name = True

        if len(mastab)==1:

            mastab[-1]=-3

        else:

            del(mastab[-1])

    else:

        i += 1

f = open("./Monday.yaml", 'w', encoding='utf-8')

f.write(result)

f.close()

И результат программы (содержимое файла Monday.yaml):

Первая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время начала: '15:20'

  Время окончания: '16:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Вторая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время начала: '17:00'

  Время окончания: '18:30'

  Корпус: Кронверкский проспект, 49

  Тип: Практика

Третья пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время начала: '18:40'

  Время окончания: '20:10'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Четвертая пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время начала: '20:20'

  Время окончания: '21:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Немного изменим входной файл, объединив время начала и конца в новый объект «время», например:

"Время": {

      "Начало": "15:20",

      "Конец": "16:50"

    }

Вывод программы:

Первая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время:

    Начало: '15:20'

    Конец: '16:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Вторая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время:

    Начало: '17:00'

    Конец: '18:30'

  Корпус: Кронверкский проспект, 49

  Тип: Практика

Третья пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время:

    Начало: '18:40'

    Конец: '20:10'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Четвертая пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время:

    Начало: '20:20'

    Конец: '21:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

В JSON есть массивы, реализуем их:

"Первая пара": {

    "Название": "Наука и техника в истории цивилизации",

    "Преподаватель": "Васильев Андрей Владимирович",

    "Аудитория": "2433",

    "Время": ["15:20", "16:50"],

    "Корпус": "Кронверкский проспект, 49",

    "Тип": "Лекция"

  }

И вывод:

Первая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время:

  - '15:20'

  - '16:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

№4

В своей программе для этого задания я использовал библиотеки json и pyyaml. Код программы:

import json, yaml

js = json.loads(open('./Monday.json',encoding='utf8').read())

s = yaml.safe\_dump(js, default\_flow\_style=False, allow\_unicode=True, sort\_keys=False)

open('./Monday\_lib.yaml', 'w', encoding='utf8').write(s)

Вывод программы (содержимое файла Monday\_lib.yaml):

Первая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время:

  - '15:20'

  - '16:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Вторая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время:

    Начало: '17:00'

    Конец: '18:30'

  Корпус: Кронверкский проспект, 49

  Тип: Практика

Третья пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время:

    Начало: '18:40'

    Конец: '20:10'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Четвертая пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время:

    Начало: '20:20'

    Конец: '21:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Как можно заметить, внешне результаты №3 и №4 неотличимы, однако на самом деле при использовании библиотек в конце файла получается пустая строка, чего нет при использовании моего алгоритма. Разница заключается лишь в условиях переноса строки: я это делаю после запятой, библиотека ставит перенос после символа «}». Сходство объясняется тем, что при конвертации я старался соблюсти все требования разметки YAML.

№5

Исходный код с регулярными выражениями:

import re

S = open('./Monday.json', encoding='utf8').readlines()

newlevel = r'.\*\{'

newobj = r'\".\*\"'

newsimp = r'.\*[\,]\*'

simple = r'\"[^\"]\*\"'

endoflevel = r'.\*\}[\,]\*'

result = ''

tab = -1

for s in S:

    s = s.rstrip()

    if re.fullmatch(newlevel, s):

        if s != '{':

            result += tab\*'  ' + re.findall(newobj, s)[0].replace('"', '')+':'+'\n'

        tab += 1

    elif re.fullmatch(endoflevel, s):

        tab -= 1

        if result[-1] != '\n':

            result += '\n'

    elif re.match(newsimp, s):

        q = re.findall(simple, s)

        name = q[0]

        value = q[1]

        e = False

        for x in value:

            if x.isalpha():

                e = True

                break

        if e:

            result += tab\*'  ' + name.replace('"', '') + ': ' + value.replace('"', '') + '\n'

        else:

            result += tab\*'  ' + name.replace('"', '') + ': ' + value.replace('"', '\'') + '\n'

f = open('Monday\_reg.yaml', 'w', encoding='utf8')

f.write(result)

f.close()

Вывод программы:

Первая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время: '15:20'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Вторая пара:

  Название: Наука и техника в истории цивилизации

  Преподаватель: Васильев Андрей Владимирович

  Аудитория: '2433'

  Время:

    Начало: '17:00'

    Конец: '18:30'

  Корпус: Кронверкский проспект, 49

  Тип: Практика

Третья пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время:

    Начало: '18:40'

    Конец: '20:10'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Четвертая пара:

  Название: Математика (продвинутый уровень)

  Преподаватель: Абжандадзе Зураб Леванович

  Аудитория: '1229'

  Время:

    Начало: '20:20'

    Конец: '21:50'

  Корпус: Кронверкский проспект, 49

  Тип: Лекция

Вывод алгоритма с регулярными выражениями не отличается от вывода в №4. Опять же, при написании алгоритма я, как и создатели библиотеки, придерживался правил разметки YAML, поэтому результаты одинаковы.

№6

Для более точного измерения среднего времени исполнения каждого алгоритма, запустим алгоритм в цикле 100 раз и разделим время на 100.

Посчитаем среднее время для каждого алгоритма, используя следующий код:

from time import time

u = time()

for \_ in range(100):

    #...

    #Code

    #...

print((time() – u)/100)

Время для моего алгоритма (без регулярных выражений и библиотек): 0.00335447072982788119 секунды

Время для алгоритма с использованием библиотек: 0.00994575738906860368 секунды

Время для алгоритма с использования регулярных выражений: 0.002288749217987060334 секунды

Как можно заметить, разница между первым и третьим алгоритмами в среднем 1 миллисекунда. Это объясняется тем, что различие заключатся лишь в разных подходах к парсингу строк, к тому же в своём алгоритме я не старался сильно оптимизировать код, а библиотека для регулярных выражений хорошо оптимизирована.

Среднее время второго алгоритма при этом почти в 3 раза больше, чем первого. Это тоже легко объяснить: хоть библиотеки и оптимизированы, для конвертации мне пришлось использовать объект Python, который парсился из файла JSON, что занимает явно больше времени, чем построчный парсинг.

№8

Я выбрал парсинг из JSON и конвертацию в CSV. CSV файл можно представить как таблицу значений, поэтому не понятно, как определить массив в файле CSV или сделать больше 2 уровней вложенности. Я буду работать с самой первой версией файла:

{

  "Первая пара": {

    "Название": "Наука и техника в истории цивилизации",

    "Преподаватель": "Васильев Андрей Владимирович",

    "Аудитория": "2433",

    "Начало": "15:20",

    "Конец": "16:50",

    "Корпус": "Кронверкский проспект, 49",

    "Тип": "Лекция"

  },

   ...

}

Код программы:

import json, csv

js = json.loads(open('./Monday\_cs.json',encoding='utf8').read())

wr = csv.writer(open('./Monday\_cs.csv','w',encoding='utf8'),delimiter=',',skipinitialspace=False)

headers = []

for x in js:

    headers.append(x)

columns = ['']+list(js[headers[0]].keys())

wr.writerow(columns)

for i in range(len(headers)):

    t = [headers[i]]

    for j in range(1,len(columns)):

        t.append(js[headers[i]][columns[j]])

    wr.writerow(t)

Вывод:

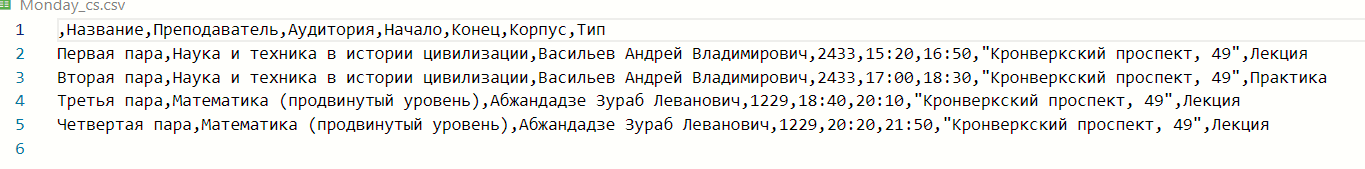
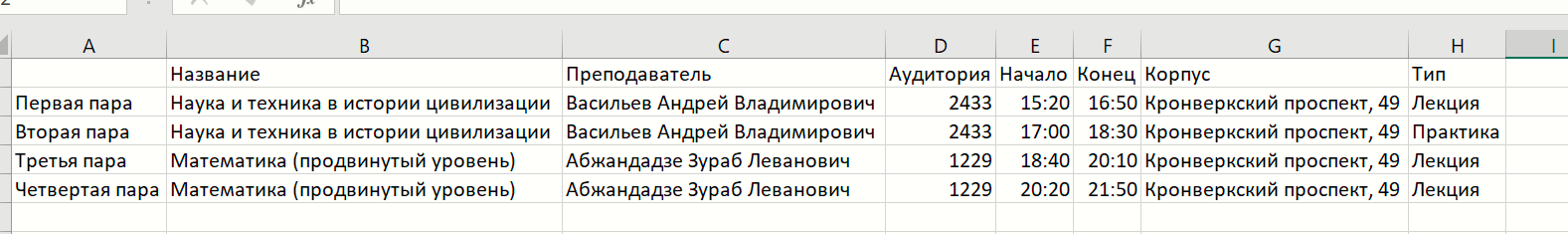


Рисунок 2. Вывод (CSV файл)

В таблице (Excel):

Рисунок 3. CSV файл в таблице Excel

№8

С помощью онлайн конвертеров я проверял выходной файл YAML (или CSV) и исправлял код, пока не пришёл к верному результату.

# Заключение

В этой лабораторной работе я научился писать собственную программу для парсинга из JSON и конвертации в YAML или CSV. Было тяжело соблюсти все требования разметки, используя только функцию записи в файл.

# Список литературы

**JSON to CSV** [В Интернете]. - https://csvjson.com/json2csv.

**JSON to YAML** [В Интернете]. - https://www.json2yaml.com/no-ads-or-analytics.