



УНИВЕРСИТЕТ ИТМО

Факультет программной инженерной и компьютерной техники

Информатика

Лабораторная работа №4

Вариант 2

Преподаватель: Малышева Татьяна Алексеевна

Выполнил: Альхимович Арсений Дмитриевич

Р3110

Санкт-Петербург, 2023

Задание	2
Исходный файл json	4
Конечные файлы	4
Основное Задание	6
Дополнительные задания	6
Вывод	9
Список литературы.....	10

Задание

1. Определить номер варианта как остаток деления на 36 последних двух цифр своего идентификационного номера в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.
2. Изучить формулу Бэкуса-Наура.
3. Изучить основные принципы организации формальных грамматик.
4. Изучить особенности языков разметки/форматов JSON, YAML, XML.
5. Понять устройство страницы с расписанием на примере расписания лектора:
https://itmo.ru/ru/schedule/3/125598/raspisanie_zanyatiy.htm
6. 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.
7. Обязательное задание (позволяет набрать до 45 процентов от максимального числа баллов БаРС за данную лабораторную): написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый путём простой замены метасимволов исходного формата на метасимволы результирующего формата.
8. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

9. Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). а) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов. б) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать. в) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

10. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную). а) Переписать исходный код, добавив в него использование регулярных выражений. б) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

11. Дополнительное задание №3 (позволяет набрать +25 процентов от максимального числа баллов БаРС за данную лабораторную). а) Переписать исходный код таким образом, чтобы для решения задачи использовались формальные грамматики. То есть ваш код должен уметь осуществлять парсинг и конвертацию любых данных, представленных в исходном формате, в данные, представленные в результирующем формате: как с готовыми библиотеками из дополнительного задания №1. б) Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом. в) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

12. Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную). а) Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле. б) Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

13. Дополнительное задание №5 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную). а) 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. б) Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.

Исходный файл json



Рисунок1. Исходный файл json

Конечные файлы



Рисунок2. Конечный файл yaml Понедельник

```
---
group_name: P3110
day: Tuesday
lesson1:
  title: Дискретка
  class_format: Очно
  type: Лекция
  campus: Кронва
  from_time: 11:40
  to_time: 13:10
  auditory: 2337
  teacher: Поляков Владимир Иванович

lesson2:
  title: Дискретка
  class_format: Очно
  type: Практика
  campus: Кронва
  from_time: 13:30
  to_time: 15:00
  auditory: 2337
  teacher: Поляков Владимир Иванович
```

Рисунок3. Конечный файл yaml Вторник

```
group_name,day,lesson1/title,lesson1/class_format,lesson1/type,lesson1/campus
,lesson1/from_time,lesson1/to_time,lesson1/auditory,lesson1/teacher,lesson2/t
itle,lesson2/class_format,lesson2/type,lesson2/campus,lesson2/from_time,lesso
n2/to_time,lesson2/auditory,lesson2/teacher,lesson3/title,lesson3/class_forma
t,lesson3/type,lesson3/campus,lesson3/from_time,lesson3/to_time,lesson3/audit
ory,lesson3/teacher,lesson4/title,lesson4/class_format,lesson4/type,lesson4/c
ampus,lesson4/from_time,lesson4/to_time,lesson4/auditory,lesson4/teacher
P3110, Monday, Матан, Очно-дистанционный, Лекция, Кронва, 08, 09, 1404,
Рванова Алла, Матан, Очно-дистанционный, Практика, Кронва, 10, 11, 2412,
Исаева Татьяна Тимофеевна, Иностранный язык, Очно, Практика, Лома, 15, 16,
3210, Левшинова Екатерина Андреевна, Иностранный язык, Очно, Практика, Лома,
17, 18, 3210, Левшинова Екатерина Андреевна
```

Рисунок4. Исходный файл csv

Основное Задание

```
"""
Понедельник В-2 (408138 38-36 = 2)
json -> yaml
"""

def main():
    json_file = open("text.json", "r").read().split('\n') #разбиение файла по строкам
    yaml_file = open("text.yaml", "w+")
    yaml_file.write("---")

    SpaceStatus = -1 #кол-во пробелов

    for line in json_file:
        line = line.strip() #убираем лишние пробелы
        newString = [] #кастыль для удаления пробелов и тд
        for sym in line:
            if sym != ',' and sym != ' ':
                newString.append(sym)
        line = ''.join(newString)
        yaml_file.write(" " * SpaceStatus) # кол-во отступов
        if "{" in line:
            SpaceStatus += 1
            yaml_file.write(line[:-1])
        elif "}" in line:
            SpaceStatus -= 1
        else:
            yaml_file.write(line)
        yaml_file.write("\n") # переход на новую строку
    return go(f, seed, [])
```

Рисунок5. Основное задание на Python

Дополнительные задания

```
import json, yaml

def task1():
    json_text= open("text.json", "r").read()
    python_dict=json.loads(json_text)
    ymal_text=yaml.dump(python_dict, allow_unicode=True)
    yaml_file = open("text.yaml", "w+")
    yaml_file.write(ymal_text)
```

Рисунок6. Доп задание №1 на Python

Видна существенная разница в размере кода между Основным заданием и доп заданием №1 тк мы используем уже готовую библиотеку

```

import re

def task2():
    json_text= open("text.json", "r").read()
    json_text = re.split(",", json_text)
    yaml_file = open("text.yaml", "w+")
    yaml_file.write("---")

    for s in json_text:
        s = re.sub(r'\}', '}', s)
        s = re.sub(r'\}', '}', s)
        s = re.sub("\n", '\n', s) #удаляем лишние переносы
        s = re.sub(r'\{', '\n', s) #добавляем переносы на новую строку после {
        yaml_file.write(s)

```

Рисунок 7. Доп задание №2 на Python

Видна существенная разница в размере кода между Основным заданием тк мы используем регулярные выражения, которые помогают упрощать код. В сравнении с доп заданием №1, код получился немного больше. Из сходств, я так же разбил текст файла на строки и пройдясь по каждой заменил символы на символы нового формата.

```

def task3():
    def main(json_file, yaml_file):
        yaml_file.write("---")
        SpaceStatus = -1 # кол-во пробелов
        for line in json_file:
            line = line.strip() # убираем пробелы
            line = line.replace(",", ' ') # убираем запятые
            line = line.replace("'", '"') # убираем кавычки
            yaml_file.write(" " * SpaceStatus) # кол-во отступов
            if "{" in line:
                SpaceStatus += 1
                yaml_file.write(line[:-1])
            elif "}" in line:
                SpaceStatus -= 1
            else:
                yaml_file.write(line)
            yaml_file.write("\n") # переход на новую строку

        json_file = open("text.json", "r").read().split('\n') #разбиение файла по строкам
        yaml_file = open("text.yaml", "w+")
        json_file1 = open("text1.json", "r").read().split('\n')
        yaml_file1 = open("text1.yaml", "w+")

        main(json_file, yaml_file)
        main(json_file1, yaml_file1)

```

Рисунок8. Доп задание №3 на Python

Код доп задания№3 это гибрид из кода основного задания и доп задания№2, с помощью регулярных выражений я смог убрать некоторые кастыли, чем упростил читабельность кода, а так код основного и доп задания№3 однотипны.



```
import time, main, task1, task2, task3

start_time = time.time() # время начала выполнения
for _ in range(100):
    main.main()

end_time = time.time() # время окончания выполнения
execution_time = end_time - start_time # вычисляем время выполнения
print(f"Время выполнения основного задания x100: {execution_time} секунд")

start_time = time.time() # время начала выполнения
for _ in range(100):
    task1.task1()

end_time = time.time() # время окончания выполнения
execution_time = end_time - start_time # вычисляем время выполнения
print(f"Время выполнения доп1 x100: {execution_time} секунд")

start_time = time.time() # время начала выполнения
for _ in range(100):
    task2.task2()

end_time = time.time() # время окончания выполнения
execution_time = end_time - start_time # вычисляем время выполнения
print(f"Время выполнения доп2 x100: {execution_time} секунд")

start_time = time.time() # время начала выполнения
for _ in range(100):
    task3.task3()

end_time = time.time() # время окончания выполнения
execution_time = end_time - start_time # вычисляем время выполнения
print(f"Время выполнения доп3 x100: {execution_time} секунд")
```

Рисунок9. Доп задание №4 на Python

Вывод программы:

Время выполнения основного задания x100: 0.026165485382080078 секунд

Время выполнения доп1 x100: 0.11299920082092285 секунд

Время выполнения доп2 x100: 0.02224135398864746 секунд

Время выполнения доп3 x100: 0.03168034553527832 секунд

Как видно время основного, доп задания№2 и доп задания№3 примерно одинаковы(доп 3 больше ~2 раза тк там два файла, а не один, но по факту время выполнения одинаково). В свою очередь время выполнения того же задания, но уже с готовой библиотекой заняло намного больше времени, те если увеличить масштабы, где кол-во файлов будет по несколько тысяч и размеры файлов будут гораздо больше программы написанные руками покажут себя намного продуктивнее.

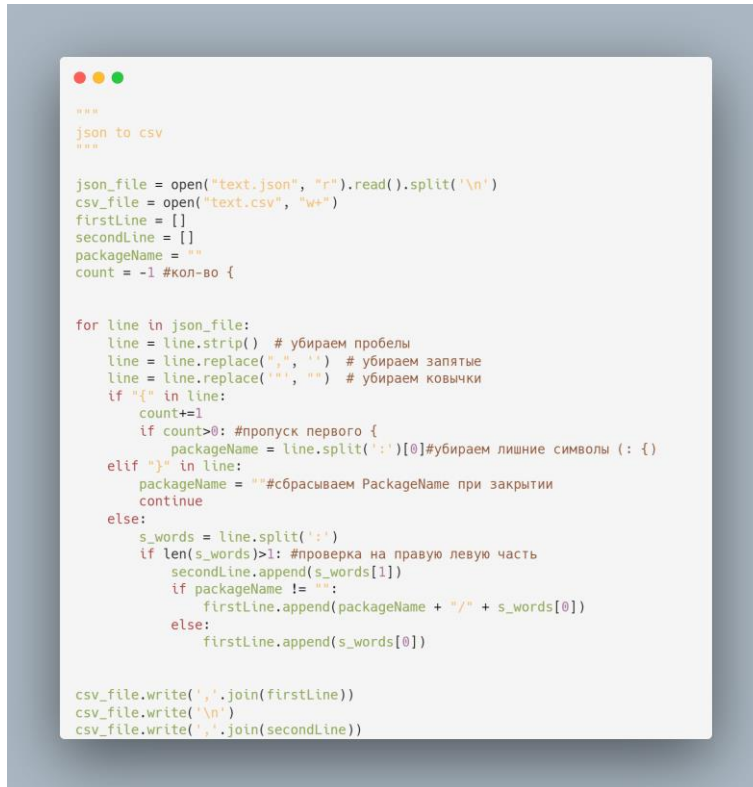


Рисунок10. Доп задание №5 на Python

Особенность формата CSV в том, что у нас имеется две строки. В первом мы пишем наши переменные, в этом формате вместе {} используются / как бы разбивая на пакеты нашу информацию. Во второй строчке пишется значения этих переменных так, что n-ому элементу 1 строчки соответствует его n-ое значение во второй. Такой формат удобен для хранения большой информации, но очень не удобен тем, что очень не читабельный и человеку, а не компьютеру скорее всего не разобраться если этих переменных будет сотни и тысячи.

Вывод

Во время лабораторной работы я создал парсер с json в yaml и с json в CSV. Повторил регулярные выражения и поработал над обновлением уже готового кода используя более простое решение каких-либо проблем. Поработал с готовыми библиотеками Python json и yaml, которые оказались очень полезными при маленьких масштабах, но не эффективными при больших.

Список литературы

Видео урон на площадке ютуб "[Python с нуля. Урок 12 | Регулярные выражения. Часть 1](#)"