**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе дисциплины   
“Информатика”

Лабораторная работа №4   
вариант 13

**Автор**:

Чимирев Игорь Олегович

**Группа**:

P3115

**Преподаватель**:

Белокон Юлия Алексеевна

Санкт-Петербург, 2024

Оглавление

[Задание 3](#_Toc117980261)

[Основные этапы выполнения 4](#_Toc117980262)

[1. Обязательное задание 4](#_Toc117980263)

[2. Дополнительное задание №1 4](#_Toc117980264)

[3. Дополнительное задание №2 4](#_Toc117980265)

[4. Дополнительное задание №3 5](#_Toc117980266)

[5. Дополнительное задание №4 5](#_Toc117980267)

[Вывод 6](#_Toc117980268)

[Список литературы 6](#_Toc117980269)

# Задание

**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.** Изучить форму Бэкуса-Наура.

**3.** Изучить особенности языков разметки/форматов JSON, YAML, XML.

**4.** Понять устройство страницы с расписанием для своей группы:

<http://itmo.ru/ru/schedule/0/P3110/schedule.htm>

**5.** 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 данный день недели нет таких занятий, то увеличить номер варианта ещё на восемь.

**6.** Обязательное задание (позволяет набрать до 65 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.

**7.** Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

**8.** Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

**a)** Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.

**b)** Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.

**c)** Сравнить полученные результаты и объяснить их сходство/различие.

**9.** Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

**a)** Переписать исходный код, добавив в него использование регулярных выражений.

**b)** Сравнить полученные результаты и объяснить их сходство/различие.

**10.** Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

**a)** 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.

**b)** Проанализировать полученные результаты и объяснить их сходство/различие.

**11.** Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

**c)** 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.

**d)** Проанализировать полученные результаты, объяснить особенности использования формата.

**12.** Проверить, что все пункты задания выполнены и выполнены верно.

**13.** Написать отчёт о проделанной работе.

**14.** Подготовиться к устным вопросам на защите

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | JSON | YAML | Понедельник |

# Основные этапы выполнения

## 1. Обязательное задание

Исходный файл json: q.json <https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/q.json>

Исходный код: main1.py

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/tasks/main1.py>

Результат: out.xml

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/out.xml>

## 2. Дополнительное задание №1

Исходный код: dop1.py <https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/q.json>

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/tasks/dop1.py>

Результат: outdop1.xml <https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/outdop1.xml>

Готовые библиотеки: стандартные библиотеки Python json ,xml.etree.ElementTree и dicttoxml для парсинга из json в xml.

Файл результата не отличается от результата обязательного задания, кроме как тем, что нет отступов между тегами в файле XML. Во остальном файлы совпадают. Код программы стал значительно проще – теперь он состоит из одной функции и нескольких строк.

## 3. Дополнительное задание №2

Исходный код: dop2.py <https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/q.json>

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/tasks/dop2.py>

Результат: outdop2.xml

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/outdop2.xml>

Файл результата полностью совпадает с файлом результата обязательного задания.

Регулярные выражения были использованы для замены недопустимых символов на подчеркивание, булевых значений и null.

## 4. Дополнительное задание №3

Исходный код: dop3.py <https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/q.json>

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/tasks/dop3.py>

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/tasks/dop3(2).py>

Результат: outdop3.xml

<https://github.com/IgorChimirev/itmo/blob/main/%D0%B8%D0%BD%D1%84%D0%B0/lab4/InOutFiles/outdop3.xml>

## Формальные используются грамматики для описания структуры JSON-данных, которые затем разбираются и преобразуются в XML-формат. Формальные грамматики помогают установить правила и структуру для разбора и преобразования данных от одного формата (JSON) к другому (XML), а код реализует эти правила через рекурсию и обработку типов данных.

## 5. Дополнительное задание №4

**1)** Время выполнения: 0.0005841255187988281 секунд

**2)** Время работы программы для доп. задания №1 (программа использует библиотеки: json, xml.etree.ElementTree, dicttoxml):

Время выполнения: 0.002729177474975586 секунд

**3)** Время работы программы для доп. задания №2 (программа использует библиотеку для регулярных выражений):

Время выполнения: 0.0008931159973144531 секунд

**4)** Время работы программы для доп. задания №3 (программа использует библиотеки: json, dicttoxml, xml.etree.ElementTree ):

Время выполнения: 0.06732010841369629 секунд

# Вывод

Во время выполнения лабораторной работы я узнал о языках разметки JSON и XML, научился с ними работать и переводить один в другой с помощью средств языка Python. Научился работать с некоторыми библиотеками для парсинга, такими как json и dicttoxml.

# Список литературы

1. Балакшин П.В., Соснин В.В., Калинин И.В., Малышева Т.А., Раков С.В., Рущенко Н.Г., Дергачев А.М. Информатика: лабораторные работы и тесты: Учебно-методическое пособие / Рецензент: Поляков В.И. - Санкт-Петербург: Университет ИТМО, 2019. - 56 с. - экз. - Режим доступа:

<https://books.ifmo.ru/book/2248/informatika:_laboratornye_raboty_i_testy:_uchebno-metodicheskoe_posobie_/_recenzent:_polyakov_v.i..htm>

1. Грошев А.С. Г89 Информатика: Учебник для вузов / А.С. Грошев. – Архангельск, Арханг. гос. техн. ун-т, 2010. -470с. -Режим доступа <https://narfu.ru/university/library/books/0690.pdf>