

Факультет Программной Инженерии и Компьютерной техники

**Информатика**

**Лабораторная работа №4**

**Вариант №2**

Выполнил:

*Барсуков Максим Андреевич*

*Группа Р3115*

Преподаватели:

*Балакшин Б. В.*

*Малышева Т. А.*

Оглавление

Задание..... 3

Основные этапы выполнения ..... 4

    1. Обязательное задание ..... 4

    2. Дополнительное задание №1 ..... 4

    3. Дополнительное задание №2 ..... 4

    4. Дополнительное задание №3 ..... 5

    5. Дополнительное задание №4 ..... 5

Вывод..... 6

Список литературы ..... 6

## 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:

<https://github.com/maxbarsukov/itmo/blob/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/data/in.json>

Исходный код:

<https://github.com/maxbarsukov/itmo/tree/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/task0>

Результат:

<https://github.com/maxbarsukov/itmo/blob/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/data/out.yaml>

### 2. Дополнительное задание №1

Исходный код:

<https://github.com/maxbarsukov/itmo/tree/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/dop1>

Результат:

<https://github.com/maxbarsukov/itmo/blob/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/data/out-dop1.yaml>

Готовые библиотеки: стандартная библиотека Python json для парсинга json и ruyaml для дампа словаря в файл yaml.

Файл результата не отличается от результата обязательного задания, кроме как тем, что нет triple-dash'a в начале (---), означающего начало документа (он не обязателен, но часто рекомендуется). Во остальном файлы совпадают. Код программы стал значительно проще – теперь он состоит из одной функции и нескольких строк.

### 3. Дополнительное задание №2

Исходный код:

<https://github.com/maxbarsukov/itmo/tree/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/dop2>

Результат:

<https://github.com/maxbarsukov/itmo/blob/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/data/out-dop2.yaml>

Файл результата полностью совпадает с файлом результата обязательного задания.

Регулярные выражения были использованы для более простого парсинга литералов JSON: строк, чисел, булевых значений и null.

#### 4. Дополнительное задание №3

1) Время работы программы для обязательного задания:

0.8610539436340332 секунды

2) Время работы программы для доп. задания №1 (программа использует библиотеку ruyaml):

0.4934084415435791 секунды

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

0.555591344833374 секунды

Возможное объяснение результатов:

Дольше всего работает программа (1), т.к. она производит полноценный парсинг JSON, средствами Python, в то время как две остальные программы используют нативные решения: так, движок для регулярных выражения в стандартной библиотеке Python написан на С и хорошо оптимизирован, поэтому код, использующий его вместо прохода по строке, который написан на самом Python, выполняется быстрее. Библиотека PyYaml работает быстрее всего также благодаря тому, что значительная её часть написана на Cython (язык для написания C/C++ модулей для Python). Однако, так как размер данных невелик, разница несущественная. Использование regex позволило значительно увеличить производительность программы (1 → доп. 2), заменив всего несколько методов.

#### 5. Дополнительное задание №4

Исходный код:

<https://github.com/maxbarsukov/itmo/tree/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/dop4>

Результат:

<https://github.com/maxbarsukov/itmo/blob/master/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4/tasks/data/out-dop4>

Protobuf – бинарный формат обмена структурированными данными. Это означает, что он, в отличие от JSON (который текстовый формат) обладает очень плохой читаемостью. Тем не менее, эта его особенность позволяет ему сериализоваться и десериализоваться быстрее и занимать меньше места сообщению, тем самым делая его передачу по сети быстрее. Данные для передачи должны быть структурированы, для описания структуры сообщения используется язык proto. Средство protoc помогает сгенерировать парсер для многих языков из файлов proto. Кроме того, используя уникальный номер поля, Protobuf обеспечивает превосходную обратную совместимость. Также Protobuf проверяет тип значения заданных данных во время кодирования и декодирования, обеспечивая целостность данных во время передачи данных (легко заметить, что proto типизирован). Однако, Protobuf требует привязку модели данных к схеме, что не всегда может быть удобно.

## Вывод

Во время выполнения лабораторной работы я узнал о языках разметки JSON и YAML, 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 ruyaml; узнал о Protobuf и научился с ним работать.

## 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>
2. 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>