



Вариант №109
Лабораторная работа №4
По дисциплине Информатика

Выполнил студент группы Р3131:
Эллити Мохамед Эмад Ахмед Авад

Преподаватель:
Балакшин Павел Валерьевич
Марухленко Даниил Сергеевич

Санкт-Петербург 2025 г.

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

Задание:

1. Определить номер варианта как остаток деления на 132 своего идентификационного номера в ISU: например, $125598 / 132 = 16$. Вслучае, если в обоих указанных днях недели нет занятий, то увеличить номер варианта на восемь. В случае, если занятий нет и в новом наборе дней, то продолжать увеличивать на восемь.
2. Изучить форму Бэкуса-Наура.
3. Изучить основные принципы организации формальных грамматик.
4. Изучить особенности языков разметки/форматов JSON, RON, HCL, YAML, TOML, INI, 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 хотя бы в одной из выбранных дней 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 50 процентов от максимального числа баллов БаРС за данную лабораторную). Написать программу на языке Python 3.x или любом другом, которая:
 - осуществляет парсинг и конвертацию исходного файла в бинарный объект (=десериализацию);
 - для решения задачи использует 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 №2);
 - не использует готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки файлов.
8. Дополнительное задание №1 (позволяет набрать +15 процентов от максимального числа баллов БаРС за данную лабораторную). Написать программу на языке Python 3.x или любом другом, которая: 1b)
 - осуществляет парсинг и конвертацию бинарного объекта, полученного в обязательном задании, в новый формат (=сериализацию);
 - для решения задачи использует формальные грамматики;
 - не использует готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки файлов.
9. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
 - а) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов (десериализацию и сериализацию).
 - б) Переписать исходный код и код из дополнительного задания №1, 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 №3 (позволяет набрать +20 процентов от максимального числа баллов БаРС за данную лабораторную).

Переписать код из дополнительного задания №1, чтобы сериализация производилась в XML файл.
11. 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.
12. Проверить, что все пункты задания выполнены и выполнены верно.
13. Написать отчёт о проделанной работе.
14. Подготовиться к устным вопросам на защите.

Основные этапы выполнения

1. Обязательное задание

Исходный файл INI:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/data/Schedule.json>

Исходный код:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/main/task1.py>

Результат:

https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/output/Task1_Result.txt

2. Дополнительное задание № 1

Исходный код:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/main/taskExtra1.py>

Результат:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/output/taskExtra1Result.yaml>

3. Дополнительное задание № 2

Исходный код:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/main/taskExtra2.py>

Результат:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/output/taskExtra2Result.yaml>

4. Дополнительное задание № 3

Исходный код:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/main/taskExtra3.py>

Результат:

https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/output/taskExtra3_Result.xml

5. Дополнительное задание № 4

Исходный код:

<https://github.com/mohamedellithyyyy/ITMO-Hub/blob/main/Computer-Science/лабораторные/Lab4/main/taskExtra4.py>

Результат:

```
Manual parser + YAML: 0.08205890655517578 sec
YAML library: 0.17187714576721191 sec
Manual parser + XML: 0.08003902435302734 sec
```

В ходе эксперимента были измерены времена выполнения трёх операций:

1. Ручной парсинг JSON + генерация YAML
2. Парсинг YAML с помощью библиотеки PyYAML
3. Ручной парсинг JSON + генерация XML

Полученные результаты показали:

- **Самый быстрый** — ручной парсер + YAML (~0.082 s)
Это объясняется тем, что собственный парсер и конвертер реализованы максимально просто и работают только с ограниченным форматом данных. У них нет сложной логики, проверки типов и поддержки расширенных возможностей, поэтому они выполняются быстрее.
- **Самый медленный** — библиотека PyYAML (~0.172 s)
PyYAML — это универсальный парсер, который должен корректно разбирать YAML любого уровня сложности. В нём много дополнительной логики: проверка структуры, обработка типов, безопасность. Это приводит к заметному увеличению времени выполнения.
- **Средний по скорости** — ручной парсер + XML (~0.080 s)
Генерация XML требует больше операций, чем генерация YAML (теги, закрывающие элементы, экранирование символов), но при этом всё равно проще, чем полноценный парсинг YAML библиотекой.

Итог: различия во времени объясняются сложностью используемых инструментов. Простая ручная реализация работает быстрее, универсальная библиотека — медленнее, а XML находится примерно посередине по объёму операций, но в вашем эксперименте её выполнение оказалось даже чуть быстрее ручного YAML, что связано с малым объёмом данных.

Вывод

Во время выполнения лабораторной работы я узнал о языках разметки JSON, YAML, TOML и XML и научился с ними работать, включая конвертацию данных из одного формата в другой с помощью Python. Я освоил работу с библиотеками для парсинга, такими как json, ruyaml и toml, а также научился самостоятельно реализовывать сериализацию и десериализацию данных без использования готовых библиотек. Кроме того, я научился работать с бинарными объектами и применять формальные грамматики для обработки данных.

Список литературы

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/catalog/2019/catalog_2024.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>