**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет ПИиКТ**

**Дисциплина: Информатика**

**Лабораторная работа №4**

**Исследование протоколов, форматов обмена**

**информацией и языков разметки документов**

**Вариант 27**

Выполнил: Ткачев Денис Владимирович

Группа: Р3111

Преподаватель: доцент, кандидат технических наук

Малышева Татьяна Алексеевна

Санкт-Петербург 2024г.

Содержание

[Задание 3](#_Toc183007556)

[Основные этапы вычисления 5](#_Toc183007557)

[Исходное расписание и json-файл 5](#_Toc183007558)

[Задание 1 (Основное задание) 6](#_Toc183007559)

[Задание 2 (Дополнительное №1) 6](#_Toc183007560)

[Задание 3 (Дополнительное №2) 6](#_Toc183007561)

[Задание 4 (Дополнительное №3) 6](#_Toc183007562)

[Задание 5 (Дополнительное №4) 7](#_Toc183007563)

[Заключение 7](#_Toc183007564)

[Литература 7](#_Toc183007565)

# Задание

1. Определить номер варианта как остаток деления на 36 последних двух цифр своего идентификационного номера в ISU: например, **466727 % 36 = 27**. В случае, если в оба указнных дня 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, 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 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 45 процентов от максимального числа баллов БаРС за данную лабораторную): написать программу на языке Python 3.x или любом другом, 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).
10. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
11. Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
12. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
13. **Дополнительное задание №2** (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
14. Переписать исходный код, добавив в него использование регулярных выражений.
15. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
16. **Дополнительное задание №3** (позволяет набрать +25 процентов от максимального числа баллов БаРС за данную лабораторную).
17. 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.
18. Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.
19. Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
20. **Дополнительное задание №4** (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
21. Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
22. Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.
23. **Дополнительное задание №5** (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
24. 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.
25. Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.
26. Проверить, что все пункты задания выполнены и выполнены верно.
27. Написать отчёт о проделанной работе.
28. Подготовиться к устным вопросам на защите.

Таблица 1. Вариант 27

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер варианта** | **Исходный формат** | **Результирующий формат** | **Дни недели** |
| 27 | YAML | JSON | Среда |
| Пятница |

# Основные этапы вычисления

Исходное расписание и YAML-файл

Получив необходимые теоретические знания, было решено составить расписание конкретных для данного варианта дней в заданном формате. Получившийся файл input.yml имеет следующий вид:

group\_name: P3111  
schedule\_Wednesday:  
 - day\_id: 2  
 day\_name: Среда  
 even\_week: false  
 schedule:  
 - classroom: Ауд. 1216, ул.Ломоносова, д.9, лит.М  
 subject: Информатика  
 teacher: Балакшин Павел Валерьевич  
 start\_time: '8:20'  
 end\_time: '9:50'  
 class\_format: Очно-дистанционный  
 type: Лекция  
 - classroom: Ауд. 1216, ул.Ломоносова, д.9, лит.М  
 subject: Основы профессиональной деятельности  
 teacher: Клименков Сергей Викторович  
 start\_time: '10:00'  
 end\_time: '11:30'  
 class\_format: Очно-дистанционный  
 type: Лекция  
 - classroom: Ауд. 1338, Кронверкский пр., д.49, лит.А  
 subject: Информатика  
 teacher: Малышева Татьяна Алексеевна  
 start\_time: '13:30'  
 end\_time: '15:00'  
 class\_format: Очно-дистанционный  
 type: Лабораторная  
 - classroom: Ауд. 1338, Кронверкский пр., д.49, лит.А  
 subject: Информатика  
 teacher: Малышева Татьяна Алексеевна  
 start\_time: '15:20'  
 end\_time: '16:50'  
 class\_format: Очно-дистанционный  
 type: Лабораторная  
  
schedule\_Friday:  
 - day\_id: 4  
 day\_name: Пятница  
 even\_week: false  
 schedule:  
 - classroom: Ауд. 2407, Кронверкский пр., д.49, лит.А  
 subject: Математический анализ  
 teacher: Правдин Константин Валерьевич  
 start\_time: '10:00'  
 end\_time: '11:30'  
 class\_format: Очно-дистанционный  
 type: Лекция  
 - classroom: Ауд. 2414, Кронверкский пр., д.49, лит.А  
 subject: Математический анализ  
 teacher: Савченко Татьяна Владимировна  
 start\_time: '11:40'  
 end\_time: '13:10'  
 class\_format: Очный  
 type: Лекция  
 - classroom: Ауд. 2201, Кронверкский пр., д.49, лит.А  
 teacher: Ватутин Александр Дмитриевич  
 subject: Линейная алгебра  
 start\_time: '13:30'  
 end\_time: '15:00'  
 class\_format: Очный  
 type: Практика

Задание 1 (Основное задание)

Исходный файл yaml: <https://github.com/denict/lab_inf/blob/main/data/input.yml>

Исходный код программы: <https://github.com/denict/lab_inf/tree/main/task1>

Результат вывода программы: <https://github.com/denict/lab_inf/blob/main/data/out1_row.json>

Задание 2 (Дополнительное №1)

Исходный код программы: <https://github.com/denict/lab_inf/blob/main/task2/parser_lib.py>

Результат вывода программы: <https://github.com/denict/lab_inf/blob/main/data/out2_lib.json>

Готовые библиотеки: библиотека Python yaml для преобразования файл YAML формата в Словарь Python языка с помощью функции yaml.load(). Затем с помощью библиотеки json преобразуем питоновский словарь в JSON-формат.

Файл результата отличается от файла результата основного задания тем, что парсит все типы данных из YAML в JSON.

Задание 3 (Дополнительное №2)

Исходный код программы: <https://github.com/denict/lab_inf/tree/main/task3>

Результат вывода программы: <https://github.com/denict/lab_inf/blob/main/data/out3_regex.json>

Файл результата не отличается от результата обязательного задания. Единственное изменение в коде программы – проверка условий операторов ветвления и замены происходят через регулярные выражения.

Задание 4 (Дополнительное №3)

Исходный код программы: <https://github.com/denict/lab_inf/tree/main/task4>

Результат вывода программы: <https://github.com/denict/lab_inf/edit/main/data/out4_parser.json>

Исходный код программы конвертирует файл формата YAML в файл JSON следующим образом: cначала рекурсивная функция parse\_object() производит парсинг объектов «ключ-значение», используя функцию parse\_key\_value(), которая производит парсинг «ключей» и «значений» для входной строки, возвращает ключ-значение. parse\_object() возвращает словарь объектов формата «ключ-значение». Функция yaml\_to\_dict() производит перевод строки YAML формата в словарь языка Python. Функция from\_dict\_py\_json() конвертирует словарь языка Python в JSON формат, конвертируя элементы словаря в соответствующие типы данных. Функция from\_list\_py\_to\_json() преобразует элементы списка языка в Python в соответствующий тип данных. Функция escaping\_special\_characters() экранирует служебные символы.

Файл результата отличается от файла результата основного задания тем, что парсит все типы данных из YAML в JSON и конвертирует соответствующие типы данных в JSON.

Задание 5 (Дополнительное №4)

Исходный код программы: <https://github.com/denict/lab_inf/blob/main/task%205/speed_test.py>

Результат вывода программы:

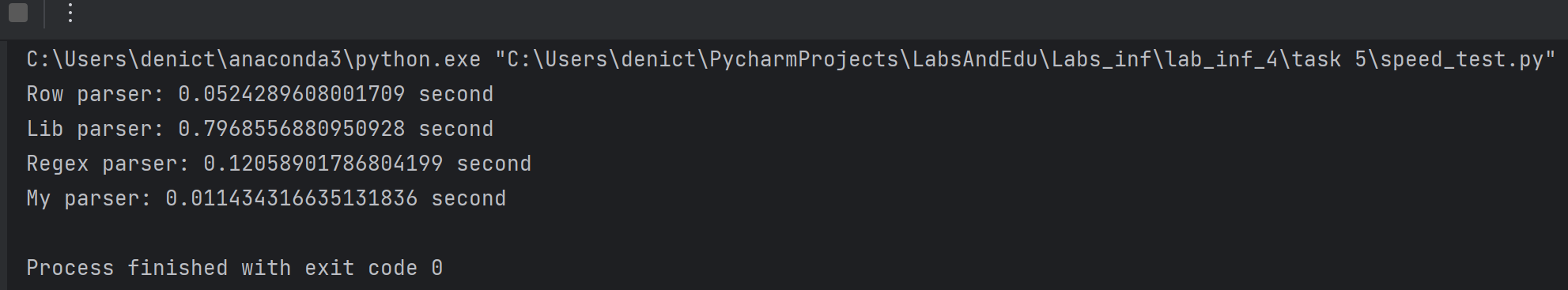


Рис. 1 Сравнение времени выполнения четырёх программ

Рекурсионный парсер-конвертор из YAML работает быстрее всего. Второе место занимает парсер исходного файла из осного задания, выполняющий построчную конвертацию. Скорость медленнее за счёт двойных циклов, в которых ищется окончание списков или вложенности.

Третье место по времени работы занимает исходный код с регулярными выражениями, работает медленнее, чем исходный файл с основного задания из-за того, что сначала программа компилирует регулярные выражения, а затем производит поиск по строке и замену, чем простая итерация проверки по строке.

Последнее место по скорости работы файла занимает файл со встроенными библиотеками yaml() и json(), за счёт того, что покрывает все возможные тесты, выполняя множество проверок внутри кода библиотек, сначала из YAML в словарь Python, а затем из словаря Python в JSON.

# Заключение

В ходе проделанной лабораторной работе были изучены новые аспекты, возможности и особенности языка программирования Python, было получено много нового интересного теоретического материала про формальные грамматики, было проведено ознакомление с такими языками разметки как YAML и JSON и были написаны различные методы по парсингу и конвертации файлов.

# Литература

1. Форма Бэкуса – Наура [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: <https://clck.ru/3EPqoV> (дата обращения: 04.11.2024).
2. Информатика: методическое пособие, 2020 [Электронный ресурс]: сост. и ред. Балакшин П.В., Соснин В.В., Машина Е.А. – Режим доступа: <https://vk.com/doc-31201840_566998093> (дата обращения: 04.11.2024).
3. Информатика: лабораторные работы и тесты, 2019 [Электронный ресурс]: Балакшин П.В., Соснин В.В., Калинин И.В., Малышева Т.А., Раков С.В., Рущенко Н.Г., Дергачев А.М. Редакционно-издательский отдел Университета ИТМО. – Режим доступа: [https://books.ifmo.ru/file/pdf/2464.pdf](https://books.ifmo.ru/file/pdf/2464.pdf%20) (дата обращения 04.11.2024).