Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 86

Выполнил:
Жгилев Иван Игоревич
Студент группы P3130
Преподаватель:
Рыбаков Степан Дмитриевич

Санкт-Петербург
2025 г

Содержание

Задание

1. Изучить форму Бэкуса-Наура.
2. Изучить основные принципы организации формальных грамматик.
3. Изучить особенности языков разметки/форматов JSON, RON, HCL, YAML, TOML, INI, XML.
4. Понять устройство страницы с расписанием на примере расписания лектора: https://itmo.ru/ru/schedule/3/125598/raspisanie_zanyatiy.htm
5. 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.
6. 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.
7. 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:
   • 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.
8. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
   a) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов (десериализацию и сериализацию).
   b) Переписать исходный код и код из дополнительного задания №1, применив найденные библиотеки. Регулярные выражения также нельзя использовать.

c) Сравнить полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

9. Дополнительное задание №3 (позволяет набрать +20 процентов от максимального числа баллов БаРС за данную лабораторную). Переписать код из дополнительного задания №1, чтобы сериализации производилась в XML файл.
10. Дополнительное задание № 4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).
    a) Используя свою исходную программу из обязательного задания и программы из дополнительных заданий, сравнить стократное время выполнения парсинга + конвертации в цикле.
    b) Проанализировать полученные результаты и объяснить их сходство/различие. Объяснение должно быть отражено в отчёте.

По варианту 86 исходный файл формата ini, а конечный формата hcl.

Исходный файл с расписанием в формате ini можно посмотреть по ссылке.

Решение заданий

Для выполнения заданий были разработаны несколько классов, о каждом из них подробней в заданиях, где они используются.

Весь исходный код и файлы можно посмотреть по ссылке.

Также есть классы, которые обладают методами общего назначения, то есть их используют несколько классов.

Код по ссылке.

Классы:

- AnalyzerSyntaxis – для анализа символов и определения корректности содержимого файлов.
- BinaryConverter – для представления числа в 4 байтах и перевода 4 байт в число.

Обязательное задание

Для выполнения этого задания были созданы классы:

- INIParser (код по ссылке) читает .ini файл и получает данные из него.
- Section и Sections (код по ссылке) классы для хранения и работы с секциями из ini файла.
- Классы ошибок (код по ссылке) для обработки исключений возникающих при чтении ini файла.
- BinaryParser (код по ссылке) по полученным данным создает файл .bin.

Конечное решение с использованием этих классов по ссылке.

Дополнительное задание №1

Для выполнения этого задания были созданы классы:

- BinaryParser (код по ссылке) читает .bin файл и получает данные из него.
- HCLSerializer (код по ссылке) по полученным данным создает файл .hcl.

Конечное решение с использованием этих классов по ссылке.

Файл, который был сформирован, можно посмотреть по ссылке.

Дополнительное задание №2

Для выполнения этого задания был создан класс:

- LibraryConverter (код по ссылке) имеет два метода, один загружает данные из ini файла и создает файл bin, а второй читает bin файл и загружает данные в файл hcl.

Использованные библиотеки:

- configparser – для работы с ini файлом;
- pickle – для работы с bin файлом;
- python-hcl2 – для работы с hcl файлом;

Конечное решение по ссылке.

Файл, который был сформирован, можно посмотреть по ссылке.

Сравним его с файлом, полученным в **Дополнительное задание №1**.

Готовая библиотека записывает каждую секцию из ini файла в hcl файл как переменную типа map. Конечный файл содержит несколько таких словарей, и каждому словарю соответствует секция из ini файла. Разработанная программа же каждую секцию из ini файла преобразует в блок HCL, что является более структурированным для человека. Подход с блоками более универсальный, но сильной разницы при работе нет.

Дополнительное задание №3

Для выполнения этого задания был создан класс:

- XMLSerializer (код по ссылке) по полученным данным создает файл формата xml.

Конечное решение по ссылке.

Файл, который был сформирован, можно посмотреть по ссылке.

Дополнительное задание №4

Решение по ссылке.

Результат времени представлен на Рис 1

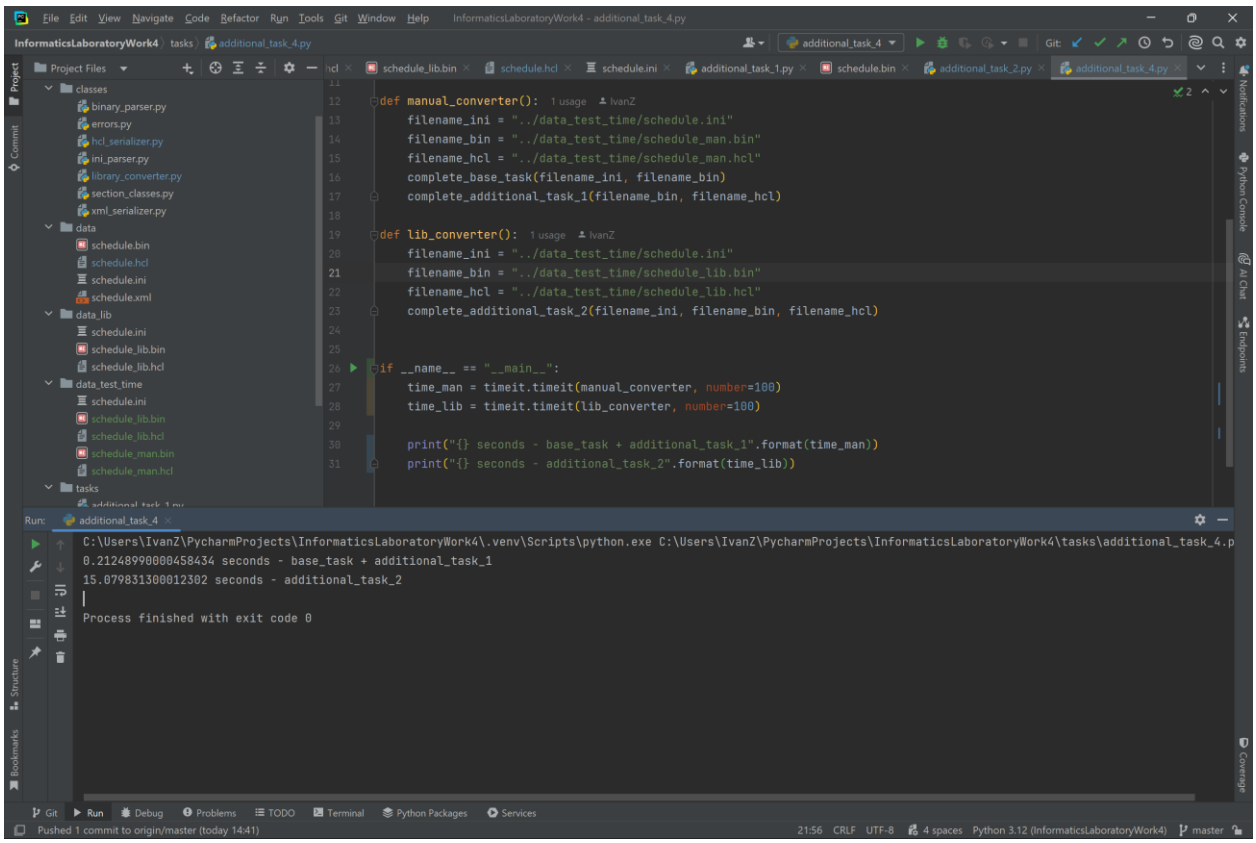

Рис 1 - Результат работы программы

0.21248990000458434 секунд – время выполнения совокупности программ из **Решение заданий**

Для выполнения заданий были разработаны несколько классов, о каждом из них подробней в заданиях, где они используются.

Весь исходный код и файлы можно посмотреть по ссылке.

Также есть классы, которые обладают методами общего назначения, то есть их используют несколько классов.

Код по ссылке.

Классы:

- AnalyzerSyntaxis – для анализа символов и определения корректности содержимого файлов.
- BinaryConverter – для представления числа в 4 байтах и перевода 4 байт в число.

Обязательное задание и **Дополнительное задание №1**

15.079831300012302 секунд – время выполнения программы из **Дополнительное задание №2**

Дольше работает программа с готовыми библиотеками (больше всего времени занимает создание строки формата hcl для записи в файл) так как она реализует полноценный парсинг с полной обработкой входных и выходных файлов.

Вывод

Во время выполнения лабораторной работы я узнал о формальных грамматиках, также об особенностях языков INI, HCL, XML и научился с ними работать, переводить файл из одного формата в другой. Полученные знания я применил на практике.

Список литературы

1. Формальная грамматика / [Электронный ресурс] // WIKI2 : [сайт]. — URL: Формальная грамматика — Википедия с видео // WIKI 2 (дата обращения: 01.11.2025).
2. Что такое XML / [Электронный ресурс] // Хабр : [сайт]. — URL: https://habr.com/ru/articles/524288/?ysclid=mhl0dpzx40787708928#xml_elems (дата обращения: 01.11.2025).