«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники (ФПИиКТ)

Лабораторная работа

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 32

Выполнил

Григорьев Даниил Александрович

Группа 3116

Принял В.В. Пономарев

Санкт-Петербург 2024

Содержание

[Задание 3](#_Toc177734059)

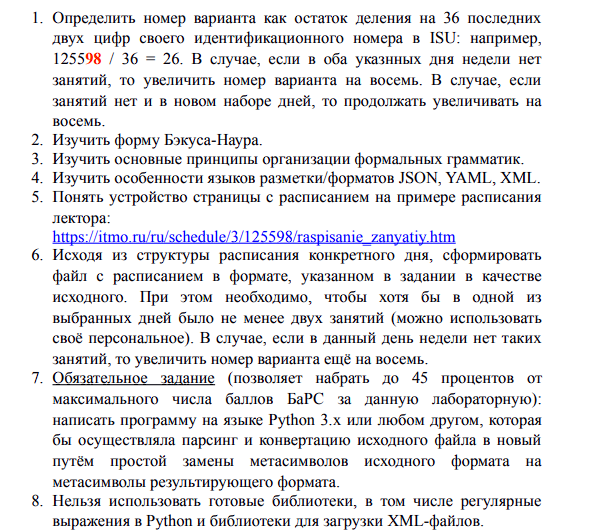
[Основные этапы вычисления 4](#_Toc177734060)

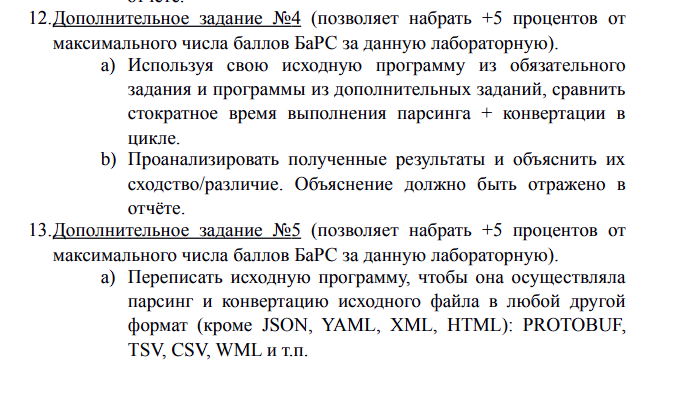
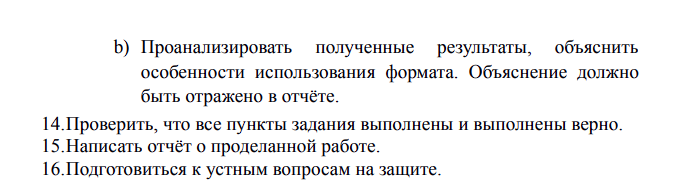
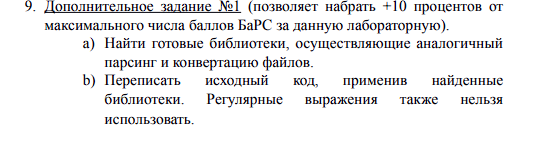
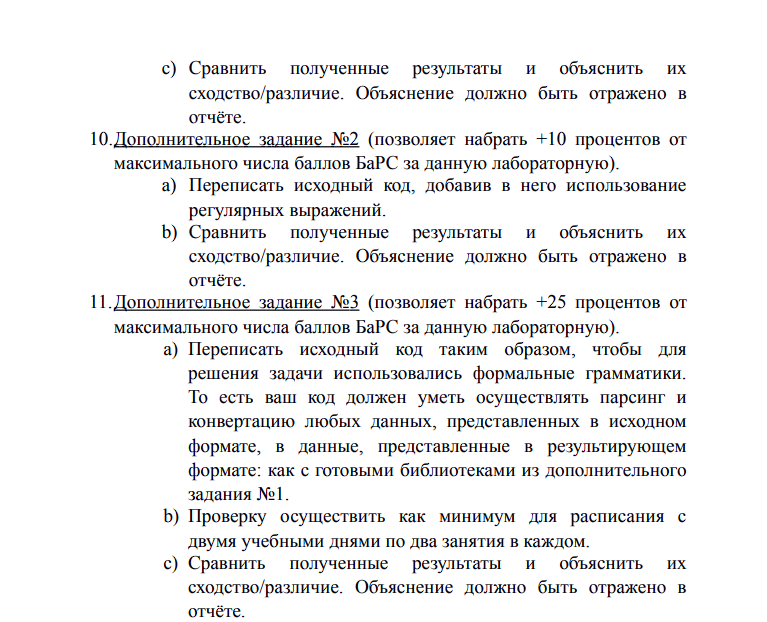
[Код программы для дополнительного задания №1: 7](#_Toc177734061)

[Заключение 8](#_Toc177734062)

[Список использованных источников 9](#_Toc177734063)

Задание



Основные этапы вычисления

Основное задание:

def parse\_input\_file():

file = open("Informatics/fourth\_lab/main\_task/input.json", "r", encoding="utf-8")

yaml\_data = file.read()

*# Заменяем ключи на YAML-формат (отступы и двоеточия)*

yaml\_data = yaml\_data.replace("{", "").replace("}", "")

yaml\_data = yaml\_data.replace("[", "").replace("]", "")

yaml\_data = yaml\_data.replace(",", "")

yaml\_data = yaml\_data.replace(":", ":\n ")

yaml\_data = yaml\_data.replace("\"", "") *# Убираем кавычки*

yaml\_data = yaml\_data.replace("\n\n", "\n") *# Убираем лишние пустые строки*

lines = yaml\_data.split("\n")

*# print(lines)*

yaml\_lines = []

for line in lines:

if line.strip() == "":

continue

indent\_level = line.find(line.strip()) *# находим уровень отступа*

yaml\_lines.append(" " \* indent\_level + line.strip())

final\_var = ""

reserved\_words = ["date", "first\_day", "second\_day", "subject", "lesson", "lecture", "time", "room", "place", "schedule"]

print(yaml\_lines)

for line in yaml\_lines:

exist\_flag = False

for i in reserved\_words:

if i in line:

final\_var += "\n" + line

exist\_flag = True

break

if exist\_flag == False:

final\_var += " " + line.strip()

*# print(final\_var)*

*# print(yaml\_lines)*

output\_file = open("Informatics/fourth\_lab/main\_task/output.yaml", "w", encoding="utf-8")

output\_file.write(final\_var)

output\_file.close()

def main():

parse\_input\_file()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Пример работы:

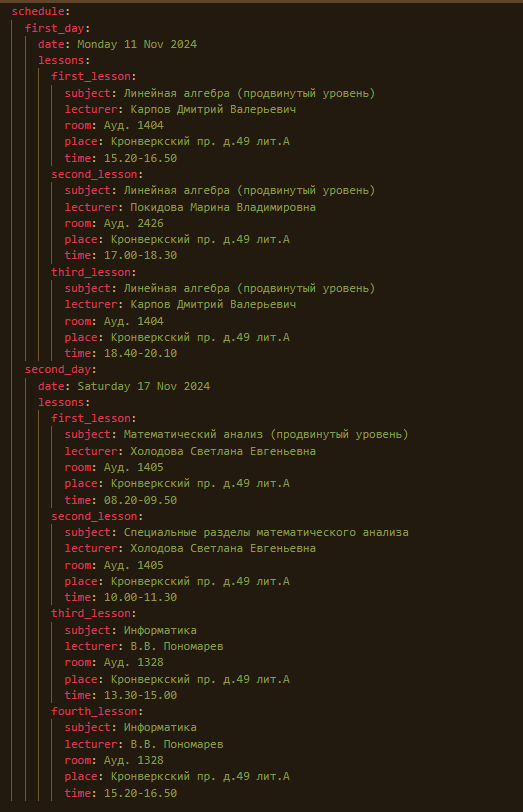
Входной файл input.json:

Первая половина файла



Вторая половина файла

Выходной файл после работы алгоритма:



Дополнительное задание №1

Код алгоритма:

import json

import yaml *# используем pyyaml*

input\_file = open("Informatics/fourth\_lab/first\_option\_task/input.json", "r", encoding="utf-8").read()

json\_data = json.loads(input\_file) *# парсим json из строки*

yaml\_data = yaml.dump(json\_data, default\_flow\_style=False, allow\_unicode="True") *# конвертируем json в yaml*

output\_file = open("Informatics/fourth\_lab/first\_option\_task/output.yaml", "w", encoding="utf-8")

output\_file.write(yaml\_data)

output\_file.close()

Входные файлы для обязательного задания и дополнительного задания №1 ничем не отличаются. В выходных файлах глобальных отличий нет, единственное отличие – в выходном коде после выполнения алгоритма дополнительного задания №1 “ключи” сортируются по названию в алфавитном порядке.

Заключение

Список использованных источников

1. [Балакшин, П. В. Информатика / П. В. Балакшин, В. В. Соснин, Е. А. Машина. – СПБ : Университет ИТМО, 2020. – 143 с.]
2. [ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» : дата введения 2017 25 09. – Москва : Стандартинформ, 2017. – 32 с.]
3. ГОСТ 7.1 — 2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления» : дата введения 01.07.2004. – Москва : ИПК Издательство стандартов, 2003. – 57 с.