Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа по информатике №4

Исследование протоколов, форматов обмена информацией и языков разметки документов

Вариант 4

Выполнил  
Пчелкин Илья Игоревич

P3106

Проверил

канд. техн. наук, доцент

Балакшин П. В.

Санкт-Петербург 2024

Оглавление

[Основные этапы вычисления 3](#_Toc183266150)

[Обязательное задание 5](#_Toc183266151)

[Дополнительное задание №1 6](#_Toc183266152)

[Дополнительное задание №2 6](#_Toc183266153)

[Дополнительное задание №3 7](#_Toc183266154)

[Дополнительное задание №4 9](#_Toc183266155)

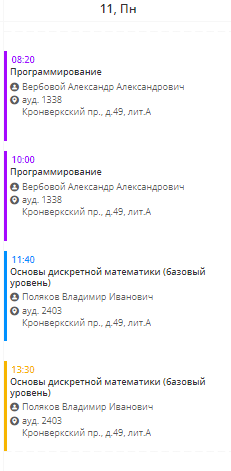
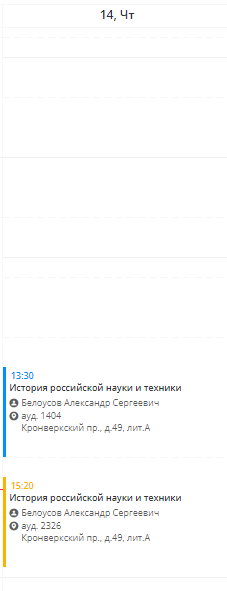
[Дополнительное задание №5 10](#_Toc183266156)

[Заключение 11](#_Toc183266157)

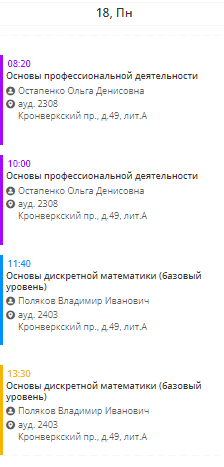
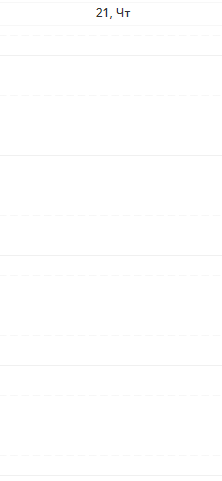
[Список использованных источников 12](#_Toc183266158)

# Основные этапы вычисления

Исходное расписание по нечётным неделям:

Исходное расписание по чётным неделям:

Исходный YAML файл:

schedule:  
 weeks:  
 -  
 week: Чётная  
 days:  
 -  
 day: Понедельник  
 disciplines:  
 -  
 discipline: Основы профессиональной деятельности  
 teacher: Остапенко Ольга Денисовна  
 building: Кронверкский пр., д.49, лит.А  
 classroom: "2308"  
 lessons:  
 -  
 lesson: Лабораторная  
 time: "8:20-9:50"  
 -  
 lesson: Лабораторная  
 time: "9:50-11:20"  
 -  
 discipline: Дискретная математика  
 teacher: Поляков Владимир Иванович  
 building: Кронверкский пр., д.49, лит.А  
 classroom: "2403"  
 lessons:  
 -  
 lesson: Лекция  
 time: "11:40-13:10"  
 -  
 lesson: Практика  
 time: "13:30-15:00"  
 -  
 day: Четверг  
 -  
 week: Нечётная  
 days:  
 -  
 day: Понедельник  
 disciplines:  
 -  
 discipline: Программирование  
 teacher: Вербовой Александр Александрович  
 building: Кронверкский пр., д.49, лит.А  
 classroom: "1338"  
 lessons:  
 -  
 lesson: Лабораторная  
 time: "8:20-9:50"  
 type: Лабораторная  
 -  
 lesson: Лабораторная  
 time: "9:50-11:20"  
  
 -  
 discipline: Дискретная математика  
 teacher: Поляков Владимир Иванович  
 building: Кронверкский пр., д.49, лит.А  
 classroom: "2403"  
 lessons:  
 -  
 lesson: Лекция  
 time: "11:40-13:10"  
 -  
 lesson: Практика  
 time: "13:30-15:00"  
 -  
 day: Четверг  
 disciplines:  
 -  
 discipline: История российской науки и техники  
 teacher: Белоусов Александр Сергеевич  
 building: Кронверкский пр., д.49, лит.А  
 classroom: "2326"  
 lessons:  
 -  
 lesson: Лекция  
 time: "13:30-15:00"  
 -  
 lesson: Практика  
 time: "15:20-16:50"

# Обязательное задание

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

path = "schedule.yml"  
yaml\_file = open(path, "r", encoding="utf-8")  
yaml\_lines = [line.strip('-') for line in yaml\_file]  
i = 0  
s = ""  
def to\_yaml():  
 global s  
 global i  
 while i < len(yaml\_lines):  
 line\_split = yaml\_lines[i].strip().replace("\"", "").split(':', 1)  
 space = len(yaml\_lines[i]) - len(yaml\_lines[i].strip())  
 if yaml\_lines[i] == "":  
 pass  
 elif len(line\_split) == 2 and line\_split[1] == '': #начало списка  
 key = line\_split[0]  
 s += ' ' \* space + f'<{key}>\n'  
 i += 1  
 if yaml\_lines[i] == yaml\_lines[i+1]:  
 s += ' ' \* space + f'</{key}>\n'  
 to\_yaml()  
 to\_yaml()  
 s += ' ' \* space + f'</{line\_split[0]}>\n'  
 elif len(line\_split) == 2 and line\_split[1] != '': #элемент списка  
 key, value = line\_split  
 s += ' ' \* space + f'<{key}>{value.strip()}</{key}>\n'  
 i += 1  
 return s.strip()  
  
with open('schedule.xml', 'w', encoding='utf-8') as f:  
 result = to\_yaml()  
 f.write(result)  
yaml\_file.close()

# Дополнительное задание №1

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. Объяснение должно быть отражено в отчёте.

import yaml  
import dict2xml  
  
yaml\_file = open('empty.yml', 'r', encoding="utf-8")  
yaml\_dict = yaml.safe\_load(yaml\_file)  
with open('schedule.xml', 'w', encoding="utf-8") as xml\_file:  
 result = dict2xml.dict2xml(yaml\_dict)  
 xml\_file.write(result)

# Дополнительное задание №2

a)  Переписать исходный код, добавив в него использование регулярных выражений.

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

import re  
path = "schedule.yml"  
with open(path, "r", encoding="utf-8") as yaml\_file:  
 yaml\_lines = [line for line in yaml\_file]  
i = 0  
s = ""  
  
  
def to\_xml():  
 global i  
 global s  
 while i < len(yaml\_lines):  
 line = re.sub('\"', '', yaml\_lines[i])  
 space = len(line) - len(line.strip())  
 match\_list = re.fullmatch(r'(\w+):', line.strip()) # начало списка  
 match\_element = re.fullmatch(r'(\w+):( \w+.+)', line.strip())  
  
 if match\_list:  
 list\_name = match\_list.group(1)  
 s += space \* ' ' + f'<{list\_name}>\n'  
 i += 1  
 to\_xml()  
 s += space \* ' ' + f'</{list\_name}>\n'  
  
 elif match\_element: # элемент списка  
 element\_name = match\_element.group(1)  
 element\_value = match\_element.group(2).strip()  
 s += space \* ' ' + f'<{element\_name}>{element\_value}</{element\_name}>\n'  
 i += 1  
 return s.strip()  
  
with open('schedule.xml', 'w', encoding='utf-8') as xml\_file:  
 result = to\_xml()  
 xml\_file.write(result)

# Дополнительное задание №3

а) 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.

b) Проверку осуществить как минимум для расписания с двумя учебными днями по два занятия в каждом.

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

def parse\_yaml(yaml\_lines):  
 parsed\_lines = []  
 stack = []  
 i = 0  
 while i < len(yaml\_lines):  
 line = yaml\_lines[i].rstrip()  
 if line.strip() == "":  
 i += 1  
 continue  
 space\_count = indent\_level(line)  
 key, value = line\_split(line)  
 line\_dict ={"space\_count": space\_count, "key":key, "value": value, "children":[]}  
 while stack and stack[-1]["space\_count"] >= space\_count:  
 stack.pop()  
 if stack:  
 stack[-1]["children"].append(line\_dict)  
 else:  
 parsed\_lines.append(line\_dict)  
 stack.append(line\_dict)  
 i += 1  
 return parsed\_lines  
  
#функция для подсчёта уровня вложенности  
def indent\_level(line):  
 flag = 0  
 space\_count = 0  
 for i in range(len(line)):  
 if line[i] == " " and flag == 0:  
 space\_count += 1  
 if line[i] != " ":  
 flag = 1  
 return(space\_count)  
  
#функция для разбиения строки на ключ и значение  
def line\_split(line):  
  
 if ":" in line:  
 key, value = line.split(":", 1)  
 key = key.strip()  
 value = value.strip()  
 return key, value  
  
#функция для записи в XML  
def to\_xml(elements, level=0):  
 xml\_str = ""  
 indent = " " \* level  
 for element in elements:  
 key = element["key"]  
 value = element["value"]  
 children = element["children"]  
 if children:  
 xml\_str += f"{indent}<{key}>\n"  
 xml\_str += to\_xml(children, level + 1)  
 xml\_str += f"{indent}</{key}>\n"  
 else:  
 # Элементы без дочерних узлов  
 xml\_str += f"{indent}<{key}>{value}</{key}>\n"  
  
  
 return xml\_str  
  
  
with open('schedule.yml', 'r', encoding='utf-8') as file\_yml:  
 file = file\_yml.read().replace('\"', '').replace(" -", "")  
 lines = file.splitlines()  
  
 YAML\_parsed = parse\_yaml(lines)  
 xml\_data = to\_xml(YAML\_parsed)  
  
with open('schedule.xml', 'w', encoding='utf-8') as file:  
 file.write(xml\_data)

Получившийся XML файл:

<schedule>  
 <weeks>  
 <week>Чётная</week>  
 <days>  
 <day>Понедельник</day>  
 <disciplines>  
 <discipline>Основы профессиональной деятельности</discipline>  
 <teacher>Остапенко Ольга Денисовна</teacher>  
 <building>Кронверкский пр., д.49, лит.А</building>  
 <classroom>2308</classroom>  
 <lessons>  
 <lesson>Лабораторная</lesson>  
 <time>8:20-9:50</time>  
 <lesson>Лабораторная</lesson>  
 <time>9:50-11:20</time>  
 </lessons>  
 <discipline>Дискретная математика</discipline>  
 <teacher>Поляков Владимир Иванович</teacher>  
 <building>Кронверкский пр., д.49, лит.А</building>  
 <classroom>2403</classroom>  
 <lessons>  
 <lesson>Лекция</lesson>  
 <time>11:40-13:10</time>  
 <lesson>Практика</lesson>  
 <time>13:30-15:00</time>  
 </lessons>  
 </disciplines>  
 <day>Четверг</day>  
 </days>  
 <week>Нечётная</week>  
 <days>  
 <day>Понедельник</day>  
 <disciplines>  
 <discipline>Программирование</discipline>  
 <teacher>Вербовой Александр Александрович</teacher>  
 <building>Кронверкский пр., д.49, лит.А</building>  
 <classroom>1338</classroom>  
 <lessons>  
 <lesson>Лабораторная</lesson>  
 <time>8:20-9:50</time>  
 <lesson>Лабораторная</lesson>  
 <time>9:50-11:20</time>  
 </lessons>  
 <discipline>Дискретная математика</discipline>  
 <teacher>Поляков Владимир Иванович</teacher>  
 <building>Кронверкский пр., д.49, лит.А</building>  
 <classroom>2403</classroom>  
 <lessons>  
 <lesson>Лекция</lesson>  
 <time>11:40-13:10</time>  
 <lesson>Практика</lesson>  
 <time>13:30-15:00</time>  
 </lessons>  
 </disciplines>  
 <day>Четверг</day>  
 <disciplines>  
 <discipline>История российской науки и техники</discipline>  
 <teacher>Белоусов Александр Сергеевич</teacher>  
 <building>Кронверкский пр., д.49, лит.А</building>  
 <classroom>2326</classroom>  
 <lessons>  
 <lesson>Лекция</lesson>  
 <time>13:30-15:00</time>  
 <lesson>Практика</lesson>  
 <time>15:20-16:50</time>  
 </lessons>  
 </disciplines>  
 </days>  
 </weeks>  
</schedule>

# Дополнительное задание №4

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.

import time  
import subprocess  
  
start\_time = time.perf\_counter()  
for i in range(100):  
 subprocess.run(["/usr/local/bin/python3.12", "required\_task.py"])  
end\_time = time.perf\_counter()  
execution\_time = end\_time - start\_time  
print(f"Время выполнения required\_task: {round(execution\_time, 3)} секунд")  
  
  
  
start\_time = time.perf\_counter()  
for i in range(100):  
 subprocess.run(["/usr/local/bin/python3.12", "additional\_task\_1.py"])  
end\_time = time.perf\_counter()  
execution\_time = end\_time - start\_time  
print(f"Время выполнения additional\_task\_1: {round(execution\_time, 3)} секунд")  
  
  
start\_time = time.perf\_counter()  
for i in range(100):  
 subprocess.run(["/usr/local/bin/python3.12", "additonal\_task\_2.py"])  
end\_time = time.perf\_counter()  
execution\_time = end\_time - start\_time  
print(f"Время выполнения additonal\_task\_2: {round(execution\_time, 3)} секунд")  
  
  
start\_time = time.perf\_counter()  
for i in range(100):  
 subprocess.run(["/usr/local/bin/python3.12", "additional\_task\_3.py"])  
end\_time = time.perf\_counter()  
execution\_time = end\_time - start\_time  
print(f"Время выполнения additional\_task\_3: {round(execution\_time, 3)} секунд")

Вывод программы:

Время выполнения required\_task: 1.194 секунд

Время выполнения additional\_task\_1: 2.674 секунд

Время выполнения additonal\_task\_2: 1.487 секунд

Время выполнения additional\_task\_3: 1.122 секунд

additional\_task\_1.py медленнее всех, потому что в нём используются дополнительные библиотеки, требующие больше времени на импорт и выполнение. Остальные выполнялись примерно одинаковое количество времени, т. к. при ручном парсинге мы не используем лишние условия, которые не нужны в данном случае.

# Дополнительное задание №5

a) 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.

b) Проанализировать полученные результаты, объяснить особенности использования формата. Объяснение должно быть отражено в отчёте.

import yaml  
import csv  
  
with open('schedule.yml', 'r', encoding='utf-8') as yaml\_file:  
 yaml\_dict = yaml.safe\_load(yaml\_file)  
  
def to\_csv(yaml\_dict):  
 rows = []  
 for week\_data in yaml\_dict.get('schedule', {}).get('weeks', []):  
 week = week\_data.get('week', '')  
  
 for day\_data in week\_data.get('days', []):  
 day = day\_data.get('day', '')  
  
 for discipline\_data in day\_data.get('disciplines', []):  
 discipline = discipline\_data.get('discipline', '')  
 teacher = discipline\_data.get('teacher', '')  
 building = discipline\_data.get('building', '')  
 classroom = discipline\_data.get('classroom', '')  
  
 for lesson\_data in discipline\_data.get('lessons', []):  
 lesson = lesson\_data.get('lesson', '')  
 time = lesson\_data.get('time', '')  
  
 rows.append([week, day, discipline, teacher, building, classroom, lesson, time])  
 return rows  
  
  
rows = to\_csv(yaml\_dict)  
  
with open('schedule.csv', 'w', newline='', encoding='utf-8') as file:  
 writer = csv.writer(file)  
  
 writer.writerow(['Week', 'Day', 'Discipline', 'Teacher', 'Building', 'Classroom', 'Lesson', 'Time'])  
  
 writer.writerows(rows)

Получившийся CSV файл:

Week,Day,Discipline,Teacher,Building,Classroom,Lesson,Time  
Чётная,Понедельник,Основы профессиональной деятельности,Остапенко Ольга Денисовна,*"*Кронверкский пр., д.49, лит.А*"*,2308,Лабораторная,8:20-9:50  
Чётная,Понедельник,Основы профессиональной деятельности,Остапенко Ольга Денисовна,*"*Кронверкский пр., д.49, лит.А*"*,2308,Лабораторная,9:50-11:20  
Чётная,Понедельник,Дискретная математика,Поляков Владимир Иванович,*"*Кронверкский пр., д.49, лит.А*"*,2403,Лекция,11:40-13:10  
Чётная,Понедельник,Дискретная математика,Поляков Владимир Иванович,*"*Кронверкский пр., д.49, лит.А*"*,2403,Практика,13:30-15:00  
Нечётная,Понедельник,Программирование,Вербовой Александр Александрович,*"*Кронверкский пр., д.49, лит.А*"*,1338,Лабораторная,8:20-9:50  
Нечётная,Понедельник,Программирование,Вербовой Александр Александрович,*"*Кронверкский пр., д.49, лит.А*"*,1338,Лабораторная,9:50-11:20  
Нечётная,Понедельник,Дискретная математика,Поляков Владимир Иванович,*"*Кронверкский пр., д.49, лит.А*"*,2403,Лекция,11:40-13:10  
Нечётная,Понедельник,Дискретная математика,Поляков Владимир Иванович,*"*Кронверкский пр., д.49, лит.А*"*,2403,Практика,13:30-15:00  
Нечётная,Четверг,История российской науки и техники,Белоусов Александр Сергеевич,*"*Кронверкский пр., д.49, лит.А*"*,2326,Лекция,13:30-15:00  
Нечётная,Четверг,История российской науки и техники,Белоусов Александр Сергеевич,*"*Кронверкский пр., д.49, лит.А*"*,2326,Практика,15:20-16:50

CSV — это очень удобный формат для работы с простыми табличными данными.

Особенности формата:  
Наиболее распространённый разделитель — запятая, но могут быть использованы и другие символы, такие как точка с запятой, табуляция, пробел и другие.

# Заключение

При выполнении данной лабораторной работы я познакомился с различными форматами файлов и научился конвертировать их.

# Список использованных источников

1. Презентация лекции по информатике № 6. Электронный источник — <https://t.me/balakshin_students/266>
2. Статья «Пишем изящный парсер на Питоне» – <https://habr.com/ru/post/309242/>