**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«Санкт-Петербургский**

**национальный исследовательский университет**

**информационных технологий, механики и оптики»**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## 

**Лабораторная работа №4**

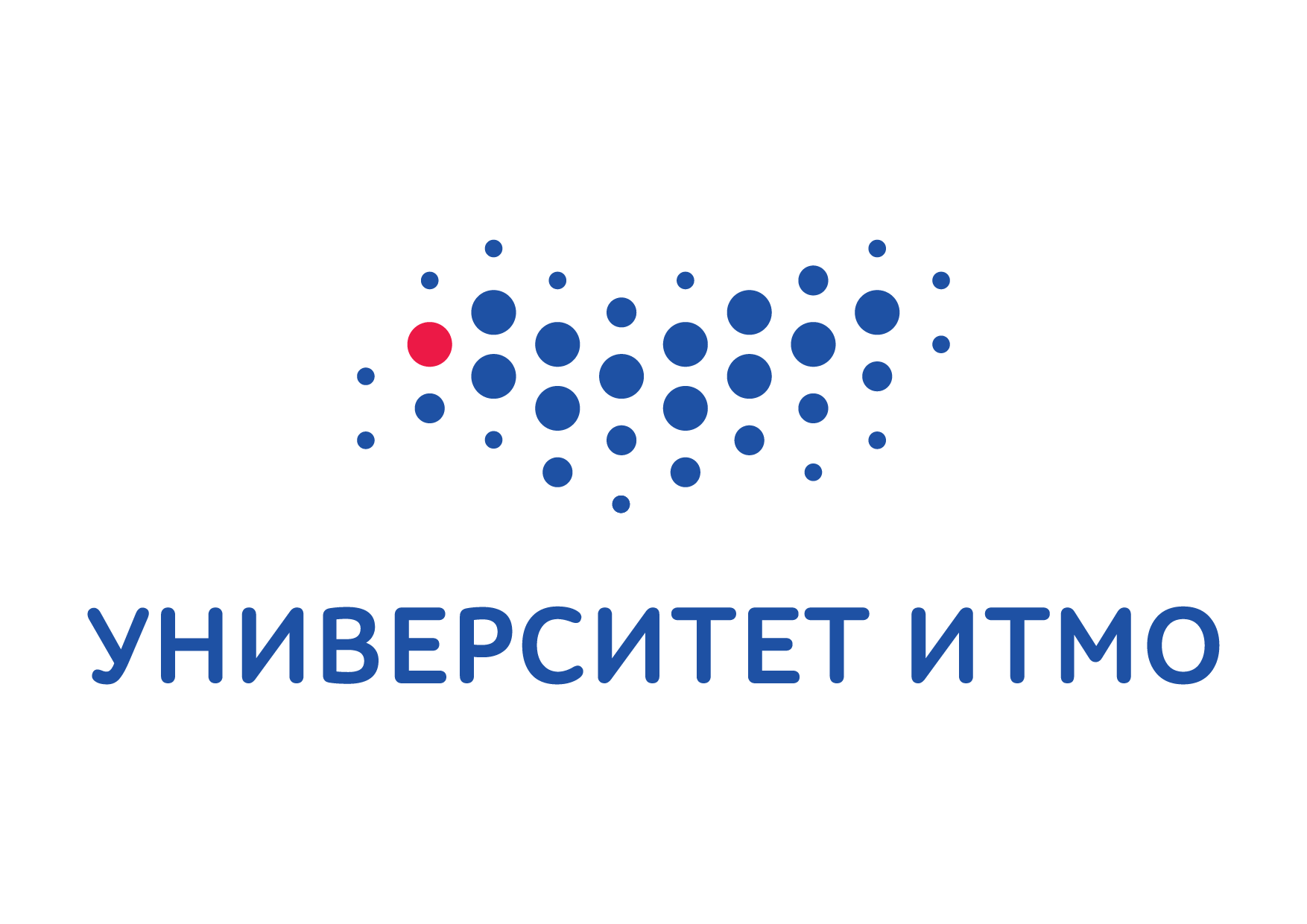
**по Информатике**

Вариант: 9

Группа: Р3110

Выполнил: Данилов Павел Юрьевич

Преподаватель: Балакшин Павел Валерьевич

****

**САНКТ-ПЕТЕРБУРГ**

**2020**

## Лабораторная работа № 4. Исследование языков разметки документов.

### **Цель задания:**

Овладеть знаниями о различных современных языках разметки документов и форматах данных, навыками обработки данных с помощью языка Python 3.x.

### **Задание**

1. Определить номер варианта как остаток деления на 35 порядкового номера в списке группы в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.
2. Изучить форму Бэкуса-Наура.
3. Изучить особенности языков разметки/форматов JSON, YAML, XML, PROTOBUF.
4. Понять устройство страницы с расписанием для своей группы: http://www.ifmo.ru/ru/schedule/0/P3110/schedule.htm
5. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного.
6. Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.
7. Нельзя использовать готовые библиотеки, кроме re (регулярные выражения в Python) и библиотеки для загрузки XML-файлов.
8. Необязательное задание для получения оценки «4» и «5» (позволяет набрать от 75 до 89 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
   2. Сравнить полученные результаты и объяснить их сходство/различие.
9. Необязательное задание для получения оценки «5» (позволяет набрать от 90 до 100 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. Используя свою программу и найденные готовые библиотеки, сравнить десятикратное время выполнения парсинга + конвертации в цикле.
   2. Проанализировать полученные результаты и объяснить их сходство/различие.
10. Проверить, что все пункты задания выполнены и выполнены верно.
11. Написать отчёт о проделанной работе.
12. Подготовиться к устным вопросам на защите.s

|  |  |  |  |
| --- | --- | --- | --- |
| **№ варианта** | **Исходный формат** | **Результирующий формат** | **День недели** |
| 9 | YAML | XML | Вторник |

### **Исходный формат:**

**Timetable**:  
 **Day**:  
 **Tuesday**:  
 **Subject1**:  
 **Time**: **'10:00-11:30'  
 Room**: **'550'  
 Building**: **'Биржевая линия, д.16, лит.А'  
 Lesson**: **'ФИЗИКА(ЛЕК)'  
 Teacher**: **'Коробков Максим Петрович'  
 Week**: **'Все'  
 Subject2**:  
 **Time**: **'11:40-13:10'  
 Room**: **'545'  
 Building**: **'Биржевая линия, д.16, лит.А'  
 Lesson**: **'ФИЗИКА(ЛАБ)'  
 Teacher**: **'Коробков Максим Петрович'  
 Week**: **'Четная неделя'  
 Subject3**:  
 **Time**: **'11:40-13:10'  
 Room**: **'545'  
 Building**: **'Биржевая линия, д.16, лит.А'  
 Lesson**: **'ФИЗИКА(ЛАБ)'  
 Teacher**: **'Коробков Максим Петрович'  
 Week**: **'Нечетная неделя'**

### **Код программы:**

**class** Subject:  
 **def** \_\_init\_\_(self):  
 self.Time = **None** self.Room = **None** self.Building = **None** self.Lesson = **None** self.Teacher = **None** self.Week = **None  
  
  
class** Timetable:  
 **def** \_\_init\_\_(self, subjects):  
 self.subjects = subjects  
  
  
**class** Parser:  
 **def** parse(self, line, subj):  
 values = line.split(**":"**)  
 key = values[0]  
 val = values[1].replace(**"\n"**, **""**)  
 val = val.replace(**"\’"**, **""**)  
 val = val[1:]  
 **if** key.count(**'Time'**) > 0:  
 subj.Time = val + **':'** + values[2].replace(**"\n"**, **""**) + **':'** + values[3].replace(**"\n"**, **""**).replace(**"\’"**, **""**)  
 **elif** key.count(**'Room'**) > 0:  
 subj.Room = val  
 **elif** key.count(**'Building'**) > 0:  
 subj.Building = val  
 **elif** key.count(**'Lesson'**) > 0:  
 subj.Lesson = val  
 **elif** key.count(**'Teacher'**) > 0:  
 subj.Teacher = val  
 **elif** key.count(**'Week'**) > 0:  
 subj.Week = val  
 **return** subj  
  
 **def** start(self, inp):  
 strings = inp.readlines()  
 subjects = [Subject(), Subject(), Subject()]  
 ind = -1  
 **for** string **in** strings:  
 string = string.replace(**"\n"**, **""**)  
 *# string = string.replace(" ", "")* **if** string.count(**"Subject"**) > 0:  
 ind += 1  
 **elif** string.count(**"Timetable"**) + string.count(**"Day"**) + string.count(**"Tuesday"**) == 0:  
 subjects[ind] = self.parse(string, subjects[ind])  
 table = Timetable(subjects)  
 **return** table  
  
  
file = open(**r"C:\Users\admin\Documents\Timetable.yaml"**, **"r"**, encoding=**"utf-8"**)  
ans = open(**r"C:\Users\admin\Documents\Timetable.xml"**, **"w"**, encoding=**"utf-8"**)  
parser = Parser()  
timetable = parser.start(file)  
ans.write(**"<Timetable>\n"**)  
ans.write(**"\t<Day>\n"**)  
ans.write(**"\t\t<Tuesday>\n"**)  
**for** i **in** range(len(timetable.subjects)):  
 subject = timetable.subjects[i]  
 ans.write(**"\t\t\t<Subject{}>\n"**.format(i + 1))  
 dict = timetable.subjects[i].\_\_dict\_\_  
 **for** j **in** dict:  
 ans.write(**"\t\t\t\t<{}>{}</{}>\n"**.format(j, dict[j], j))  
 ans.write(**"\t\t\t</Subject{}>\n"**.format(i + 1))  
ans.write(**"\t\t</Tuesday>\n"**)  
ans.write(**"\t</Day>\n"**)  
ans.write(**"</Timetable>\n"**)

### **Результат:**

<**Timetable**>  
 <**Day**>  
 <**Tuesday**>  
 <**Subject1**>  
 <**Time**>10:00-11:30</**Time**>  
 <**Room**>550</**Room**>  
 <**Building**>Биржевая линия, д.16, лит.А</**Building**>  
 <**Lesson**>ФИЗИКА(ЛЕК)</**Lesson**>  
 <**Teacher**>Коробков Максим Петрович</**Teacher**>  
 <**Week**>Все</**Week**>  
 </**Subject1**>  
 <**Subject2**>  
 <**Time**>11:40-13:10</**Time**>  
 <**Room**>545</**Room**>  
 <**Building**>Биржевая линия, д.16, лит.А</**Building**>  
 <**Lesson**>ФИЗИКА(ЛАБ)</**Lesson**>  
 <**Teacher**>Коробков Максим Петрович</**Teacher**>  
 <**Week**>Четная неделя</**Week**>  
 </**Subject2**>  
 <**Subject3**>  
 <**Time**>11:40-13:10</**Time**>  
 <**Room**>545</**Room**>  
 <**Building**>Биржевая линия, д.16, лит.А</**Building**>  
 <**Lesson**>ФИЗИКА(ЛАБ)</**Lesson**>  
 <**Teacher**>Коробков Максим Петрович</**Teacher**>  
 <**Week**>Нечетная неделя</**Week**>  
 </**Subject3**>  
 </**Tuesday**>  
 </**Day**>  
</**Timetable**>

### **Осуществить парсинг и конвертацию используя готовые библиотеки и объяснить полученные сходства и различия**

**Результат:**

<?xml version=”1.0” encoding=”UTF-8” ?>

<root><Timetable type=»dict»><Day type=»dict»><Tuesday type=»dict»><Subject1 type=»dict»><Time type=»str»>10:00-11:30</Time><Room type=»str»>550</Room><Building type=»str»>\xd0\x91\xd0\xb8\xd1\x80\xd0\xb6\xd0\xb5\xd0\xb2\xd0\xb0\xd1\x8f \xd0\xbb\xd0\xb8\xd0\xbd\xd0\xb8\xd1\x8f, \xd0\xb4.16, \xd0\xbb\xd0\xb8\xd1\x82.\xd0\x90</Building><Lesson type=»str»>\xd0\xa4\xd0\x98\xd0\x97\xd0\x98\xd0\x9a\xd0\x90(\xd0\x9b\xd0\x95\xd0\x9a)</Lesson><Teacher type=»str»>\xd0\x9a\xd0\xbe\xd1\x80\xd0\xbe\xd0\xb1\xd0\xba\xd0\xbe\xd0\xb2 \xd0\x9c\xd0\xb0\xd0\xba\xd1\x81\xd0\xb8\xd0\xbc \xd0\x9f\xd0\xb5\xd1\x82\xd1\x80\xd0\xbe\xd0\xb2\xd0\xb8\xd1\x87</Teacher><Week type=»str»>\xd0\x92\xd1\x81\xd0\xb5</Week></Subject1><Subject2 type=»dict»><Time type=»str»>11:40-13:10</Time><Room type=»str»>545</Room><Building type=»str»>\xd0\x91\xd0\xb8\xd1\x80\xd0\xb6\xd0\xb5\xd0\xb2\xd0\xb0\xd1\x8f \xd0\xbb\xd0\xb8\xd0\xbd\xd0\xb8\xd1\x8f, \xd0\xb4.16, \xd0\xbb\xd0\xb8\xd1\x82.\xd0\x90</Building><Lesson type=»str»>\xd0\xa4\xd0\x98\xd0\x97\xd0\x98\xd0\x9a\xd0\x90(\xd0\x9b\xd0\x90\xd0\x91)</Lesson><Teacher type=»str»>\xd0\x9a\xd0\xbe\xd1\x80\xd0\xbe\xd0\xb1\xd0\xba\xd0\xbe\xd0\xb2 \xd0\x9c\xd0\xb0\xd0\xba\xd1\x81\xd0\xb8\xd0\xbc \xd0\x9f\xd0\xb5\xd1\x82\xd1\x80\xd0\xbe\xd0\xb2\xd0\xb8\xd1\x87</Teacher><Week type=»str»>\xd0\xa7\xd0\xb5\xd1\x82\xd0\xbd\xd0\xb0\xd1\x8f \xd0\xbd\xd0\xb5\xd0\xb4\xd0\xb5\xd0\xbb\xd1\x8f</Week></Subject2><Subject3 type=»dict»><Time type=»str»>11:40-13:10</Time><Room type=»str»>545</Room><Building type=»str»>\xd0\x91\xd0\xb8\xd1\x80\xd0\xb6\xd0\xb5\xd0\xb2\xd0\xb0\xd1\x8f \xd0\xbb\xd0\xb8\xd0\xbd\xd0\xb8\xd1\x8f, \xd0\xb4.16, \xd0\xbb\xd0\xb8\xd1\x82.\xd0\x90</Building><Lesson type=»str»>\xd0\xa4\xd0\x98\xd0\x97\xd0\x98\xd0\x9a\xd0\x90(\xd0\x9b\xd0\x90\xd0\x91)</Lesson><Teacher type=»str»>\xd0\x9a\xd0\xbe\xd1\x80\xd0\xbe\xd0\xb1\xd0\xba\xd0\xbe\xd0\xb2 \xd0\x9c\xd0\xb0\xd0\xba\xd1\x81\xd0\xb8\xd0\xbc \xd0\x9f\xd0\xb5\xd1\x82\xd1\x80\xd0\xbe\xd0\xb2\xd0\xb8\xd1\x87</Teacher><Week type=»str»>\xd0\x9d\xd0\xb5\xd1\x87\xd0\xb5\xd1\x82\xd0\xbd\xd0\xb0\xd1\x8f \xd0\xbd\xd0\xb5\xd0\xb4\xd0\xb5\xd0\xbb\xd1\x8f</Week></Subject3></Tuesday></Day></Timetable></root>

**Вывод:**

**Первое очевидное различие:** вывод символов русского алфавита осуществляется при помощи кодов каждого из символов.

**Второе различие:** у каждого элемента разметки указан тип хранимого элемента: словарь(пример: <Subject1 type=»dict»>) или строка: (пример: <Time type=»str»>10:00-11:30)

**Третье различие:** отсутствует табуляция. Этим, вероятно, объясняется указание типов хранимых элементов у каждого из элементов разметки(иначе разметка была бы просто нечитаема).

**Сходства:** в остальном разметка идентична и подчиняется правилам языка XML: <key1></key1> и т. п.

### **Осуществить десятикратный парсинг и конвертацию в цикле библиотеками и своей программой; объяснить полученные сходства и различия во времени работы программ**

Время моей программы(10 циклов): 0.010936498641967773 seconds

Время программы с библиотеками(10 циклов): 0.07580900192260742 seconds

**Вывод:**

Уже до запуска десятикратного исполнения программ было понятно, что время исполнения программы с библиотеками будет больше. Это происходит из-за того, что в своей программе я реализовал парсер и конвертер только для заданного файла, а в библиотеках парсинг и конвертация осуществляются для общего случая. В конвертации для общего случая как минимум проверяется валидность текста, что требует временных затрат. Также в библиотеке dicttoxml(которую я использовал) осуществляется структурирование данных словаря, на что также тратится время. В моей же программе структура заранее понятна и задана статично в классе Subject. Вследствие этих, а также иных факторов, вызванных конвертацией и парсингом для общего случая, время программы с библиотеками увеличивается относительно моей программы.

### **Вывод:**

В результате выполнения работы я вспомнил основы языка Python и научился реализовывать классы в этом языке. Также я познакомился с основами синтаксиса языков разметки YAML и XML и научился преобразовывать файл из одного формата в другой и понял некоторые недостатки использования готовых библиотек в проектах.