Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»
Факультет Программной Инженерии и Компьютерной Техники

Вариант №18
Лабораторная работа №4
по дисциплине
Информатика

Выполнил Студент группы P3115
Владимир Мацюк
Преподаватель:
Малышева Татьяна Алексеевна

г. Санкт-Петербург
2022г.

Содержание

1 Задание

1.1 Вариант

1. Определить номер варианта как остаток деления на 36 порядкового номера в списке группы в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.

18	JSON	XML	Четверг

2. Изучить форму Бэкуса-Наура.
3. Изучить особенности языков разметки/форматов JSON, YAML, XML.
4. Понять устройство страницы с расписанием для своей группы: http://itmo.ru/ru/schedule/0/P3110/schedule.htm

1.2 Исходный файл

5. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного. При этом необходимо, чтобы в выбранном дне 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.

```
{
  "tbody": {
    "tr": [
      {
        "th": {
          "span": "Чт"
        },
        "td": [
          {
            "span": "11:40-13:10",
            "div": "2, 4, 6, 8, 10, 12, 14, 16",
            "dd": {
              "text": "2306/2 бывш(. 305) ауд."
            },
            "dt": {
              "span": "Кронверкский пр., д.49, литА."
            }
          },
          {
            "dl": {
              "dd": "2306/2 бывш(. 305) ауд.",
              "dt": {
                "span": "Кронверкский пр., д.49, литА."
              }
            }
          },
          {
            "dl": {
              "dd": "ИнформатикаЛаб()",
              "dt": {
                "b": "Малышева ТатьянаАлексеевна"
              },
              "text": "четная неделя"
```

```
          }
        },
        {
          "text": "Очно - дистанционный"
        }
      ]
    },
    {
      "td": [
        {
          "span": "13:30-15:00",
          "div": "2, 4, 6, 8, 10, 12, 14, 16",
          "dd": {
            "text": "2306/2 бывш(. 305) ауд."
          },
          "dt": {
            "span": "Кронверкский пр., д.49, литА."
          }
        },
        {
          "dl": {
            "dd": "2306/2 бывш(. 305) ауд.",
            "dt": {
              "span": "Кронверкский пр., д.49, литА."
            }
          }
        },
        {
          "dl": {
            "dd": "ИнформатикаЛаб()",
            "dt": {
              "b": "Малышева ТатьянаАлексеевна"
            },
            "text": "четная неделя"
          }
        },
        {
          "text": "Очно - дистанционный"
        }
      ]
    }
  ]
 }
}
```

1.3 Обязательное задание

7. Обязательное задание (позволяет набрать до 65 процентов от максимального числа баллов БаРС за данную лабораторную): написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.

8. Нельзя использовать готовые библиотеки, в том числе регулярные выражения в Python и библиотеки для загрузки XML-файлов.

task1.py

```python
from dataclasses import dataclass
from enum import Enum, auto
from pathlib import Path
```

```
 4  import pprint
 5  from typing import Iterator
 6
 7
 8  class TokenType(Enum):
 9     NONE = auto()
10     BEGIN_OBJ = auto()
11     END_OBJ = auto()
12     BEGIN_ARR = auto()
13     END_ARR = auto()
14     STR = auto()
15     COL = auto()
16     COMMA = auto()
17
18
19  @dataclass(init=True)
20  class Token:
21     val: str
22     type: TokenType
23
24
25  def tokenise(s: str):
26     tokens: list[Token] = []
27
28     it = iter(s)
29     while True:
30        try:
31           i = next(it)
32        except StopIteration:
33           break
34        cur = Token(i, TokenType.NONE)
35        match i:
36           case '{': cur.type = TokenType.BEGIN_OBJ
37           case '}': cur.type = TokenType.END_OBJ
38           case '[': cur.type = TokenType.BEGIN_ARR
39           case ']': cur.type = TokenType.END_ARR
40           case ':': cur.type = TokenType.COL
41           case ',': cur.type = TokenType.COMMA
42           case '"':
43              cur.type = TokenType.STR
44              cur.val = ''
45              while True:
46                 try:
47                    i = next(it)
48                 except StopIteration:
49                    break
50                 if i == "\\":
51                    i = next(it)
52                 elif i == '"':
53                    break
54                 cur.val += i
55
56        if cur.type ≠ TokenType.NONE:
57           tokens.append(cur)
58
59     return tokens
60
61
```

```
def parse_tokens(it: Iterator[Token]):
   i = next(it)
   match i.type:
      case TokenType.STR:
         return i.val
      case TokenType.BEGIN_OBJ:
         res = {}
         i = next(it)
         while True:
            if i.type != TokenType.STR:
               raise RuntimeError('expected str key')
            key = i.val
            i = next(it)
            if i.type != TokenType.COL:
               raise RuntimeError('expected ':'')
            val = parse_tokens(it)
            res[key] = val
            i = next(it)
            if i.type == TokenType.COMMA:
               i = next(it)
            elif i.type == TokenType.END_OBJ:
               break
            else:
               raise RuntimeError('unexpected token')
         return res
      case TokenType.BEGIN_ARR:
         res = []
         while True:
            val = parse_tokens(it)
            res.append(val)
            i = next(it)
            if i.type == TokenType.COMMA:
               continue
            elif i.type == TokenType.END_ARR:
               break
            else:
               raise RuntimeError('unexpected token')
         return res


def parse_json(s: str):
   tokens = tokenise(s)
   res = parse_tokens(iter(tokens))
   return res


def obj2xml(obj, deep=0, parent='root') -> str:
   sp = ' ' * deep
   match obj:
      case str(): return obj
      case dict():
         res = ''
         for (key, val) in obj.items():
            match val:
               case list():
                  res += f'{obj2xml(val, deep, key)}'
               case str():
                  res += f'{sp}<{key}>{val}</{key}>\n'
```

```
                case _:
                    res += f'{sp}<{key}>\n{obj2xml(val, deep+1, key)
                        ↪ }{sp}</{key}>\n'
            return res
        case list():
            return ''.join(f'{sp}<{parent}>\n{obj2xml(val, deep+1)}{
                ↪ sp}</{parent}>\n' for val in obj)


def task1(s: str):
    res = parse_json(s)
    return obj2xml(res)
```

out1.xml

```
<tbody>
  <tr>
    <th>
      <span>Чт</span>
    </th>
    <td>
      <span>11:40-13:10</span>
      <div>2, 4, 6, 8, 10, 12, 14, 16</div>
      <dd>
        <text>2306/2 бывш(. 305) ауд.</text>
      </dd>
      <dt>
        <span>Кронверкский пр., д.49, литА.</span>
      </dt>
    </td>
    <td>
      <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
        <dt>
          <span>Кронверкский пр., д.49, литА.</span>
        </dt>
      </dl>
    </td>
    <td>
      <dl>
        <dd>ИнформатикаЛаб()</dd>
        <dt>
          <b>Малышева ТатьянаАлексеевна</b>
        </dt>
        <text>четная неделя</text>
      </dl>
    </td>
    <td>
      <text>Очно - дистанционный</text>
    </td>
  </tr>
  <tr>
    <td>
      <span>13:30-15:00</span>
      <div>2, 4, 6, 8, 10, 12, 14, 16</div>
      <dd>
        <text>2306/2 бывш(. 305) ауд.</text>
      </dd>
      <dt>
```

```
45        <span>Кронверкский пр., д.49, литА.</span>
46      </dt>
47    </td>
48    <td>
49      <dl>
50        <dd>2306/2 бывш(. 305) ауд.</dd>
51        <dt>
52          <span>Кронверкский пр., д.49, литА.</span>
53        </dt>
54      </dl>
55    </td>
56    <td>
57      <dl>
58        <dd>ИнформатикаЛаб()</dd>
59        <dt>
60          <b>Малышева ТатьянаАлексеевна</b>
61        </dt>
62        <text>четная неделя</text>
63      </dl>
64    </td>
65    <td>
66      <text>Очно - дистанционный</text>
67    </td>
68  </tr>
69 </tbody>
```

1.4 Дополнительное задание №1

9. Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

   (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.

task2.py

```python
import json
from dict2xml import dict2xml


def parse_json(s: str):
    return json.loads(s)


def obj2xml(s: str):
    return dict2xml(s)


def task2(s: str):
    res = parse_json(s)
    return obj2xml(res)
```

out2.xml

```
<tbody>
```

```
  <tr>
    <td>
      <dd>
        <text>2306/2 бывш(. 305) ауд.</text>
      </dd>
      <div>2, 4, 6, 8, 10, 12, 14, 16</div>
      <dt>
        <span>Кронверкский пр., д.49, литА.</span>
      </dt>
      <span>11:40-13:10</span>
    </td>
    <td>
      <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
        <dt>
          <span>Кронверкский пр., д.49, литА.</span>
        </dt>
      </dl>
    </td>
    <td>
      <dl>
        <dd>ИнформатикаЛаб()</dd>
        <dt>
          <b>Малышева ТатьянаАлексеевна</b>
        </dt>
        <text>четная неделя</text>
      </dl>
    </td>
    <td>
      <text>Очно - дистанционный</text>
    </td>
    <th>
      <span>Чт</span>
    </th>
  </tr>
  <tr>
    <td>
      <dd>
        <text>2306/2 бывш(. 305) ауд.</text>
      </dd>
      <div>2, 4, 6, 8, 10, 12, 14, 16</div>
      <dt>
        <span>Кронверкский пр., д.49, литА.</span>
      </dt>
      <span>13:30-15:00</span>
    </td>
    <td>
      <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
        <dt>
          <span>Кронверкский пр., д.49, литА.</span>
        </dt>
      </dl>
    </td>
    <td>
      <dl>
        <dd>ИнформатикаЛаб()</dd>
        <dt>
```

```
60          <b>Малышева ТатьянаАлексеевна</b>
61        </dt>
62        <text>четная неделя</text>
63      </dl>
64    </td>
65    <td>
66      <text>Очно - дистанционный</text>
67    </td>
68  </tr>
69 </tbody>
```

1.5 Дополнительное задание №2

10. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

    (a) Переписать исходный код, добавив в него использование 2 регулярных выражений.

    (b) Сравнить полученные результаты и объяснить их сходство/различие.

task3.py

```python
import re
from task1 import Token, TokenType, obj2xml, parse_tokens


def tokenise(s: str):
    return [
        Token(match.group(match.lastindex), TokenType(match.
            ↪ lastindex + 1))
        for match in re.finditer('|'.join([
            r'(\{)',
            r'(\})',
            r'(\[)',
            r'(\])',
            r'"((?:\.|[^"])*)"',
            r'(:)',
            r'(,)',
        ]), s)
    ]


def parse_json(s: str):
    tokens = tokenise(s)
    res = parse_tokens(iter(tokens))
    return res


def task3(s: str):
    res = parse_json(s)
    return obj2xml(res)
```

out3.xml

```
<tbody>
  <tr>
    <th>
      <span>Чт</span>
    </th>
```

```
    <td>
      <span>11:40-13:10</span>
      <div>2, 4, 6, 8, 10, 12, 14, 16</div>
      <dd>
        <text>2306/2 бывш(. 305) ауд.</text>
      </dd>
      <dt>
        <span>Кронверкский пр., д.49, литА.</span>
      </dt>
    </td>
    <td>
      <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
        <dt>
          <span>Кронверкский пр., д.49, литА.</span>
        </dt>
      </dl>
    </td>
    <td>
      <dl>
        <dd>ИнформатикаЛаб()</dd>
        <dt>
          <b>Малышева ТатьянаАлексеевна</b>
        </dt>
        <text>четная неделя</text>
      </dl>
    </td>
    <td>
      <text>Очно - дистанционный</text>
    </td>
  </tr>
  <tr>
    <td>
      <span>13:30-15:00</span>
      <div>2, 4, 6, 8, 10, 12, 14, 16</div>
      <dd>
        <text>2306/2 бывш(. 305) ауд.</text>
      </dd>
      <dt>
        <span>Кронверкский пр., д.49, литА.</span>
      </dt>
    </td>
    <td>
      <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
        <dt>
          <span>Кронверкский пр., д.49, литА.</span>
        </dt>
      </dl>
    </td>
    <td>
      <dl>
        <dd>ИнформатикаЛаб()</dd>
        <dt>
          <b>Малышева ТатьянаАлексеевна</b>
        </dt>
        <text>четная неделя</text>
      </dl>
```

```
    </td>
    <td>
      <text>Очно - дистанционный</text>
    </td>
  </tr>
</tbody>
```

1.6 Дополнительное задание №3

11. Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).

    (a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания №1 и программу из дополнительного задания №2, сравнить стократное время выполнения парсинга + конвертации в цикле.

    (b) Проанализировать полученные результаты и объяснить их сходство/различие.

task4.py

```python
from pathlib import Path


def read():
    return open(Path(__file__).with_name('input.json')).read()


def write(idx: int, out: str, ext='xml'):
    open(Path(__file__).with_name(f'out{idx}.{ext}'), mode="w+").
        ↪ write(out)


def task4():
    from task1 import task1
    from task2 import task2
    from task3 import task3
    from timeit import timeit
    s = read()

    write(1, task1(s))
    write(2, task2(s))
    write(3, task3(s))

    res = '\n'.join(
        s + str(t) for (s, t) in
        [
            ('no lib + no regex: ', timeit("task1(s)", globals=locals
                ↪ (), number=100)),
            ('lib: ', timeit("task2(s)", globals=locals(), number
                ↪ =100)),
            ('regex: ', timeit("task3(s)", globals=locals(), number
                ↪ =100))
        ]
    )
    open(Path(__file__).with_name(f'out4.txt'), mode="w+").write(
        ↪ res)
    print(res)
```

out4.txt

```
no lib + no regex: 0.06759362399952806
lib: 0.026628636000168626
regex: 0.032346663000680564
```

1.7 Дополнительное задание №4

12. Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов БаРС за данную лабораторную).

    (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.

task5.py

```python
from task2 import parse_json
from task4 import read, write


def val2toml(item: any):
    match item:
        case str():
            return item.__repr__()
        case list():
            return f'[{", ".join(val2toml(i) for i in item)}]'
        case _:
            raise RuntimeError('forbidden type')


def obj2toml(item: dict, parent=''):
    res = []

    def tables_at_end(item):
        _, value = item
        return isinstance(value, dict)

    for (key, val) in sorted(item.items(), key=tables_at_end):
        path = f"{parent}.{key}" if parent else key
        match val:
            case dict():
                res.append(f'\n[{path}]\n{obj2toml(val, path)}')
            case list():
                small = True
                if len(val) != 0:
                    if val[0] is dict:
                        small = False
                if small:
                    for i in val:
                        res.append(f'\n[[{path}]]\n{obj2toml(i, path)}')
                else:
                    res.append(f'\n[[{path}]]\n{obj2toml(val, path)}')
            case _:
                res.append(f'{key} = {val2toml(val)}')
    return '\n'.join(res)
```

```
40
41
42  def task5():
43     s = read()
44     d = parse_json(s)
45     res = obj2toml(d)
46     write(5, res, 'toml')
```

out5.toml

```
 1
 2  [tbody]
 3
 4  [[tbody.tr]]
 5
 6  [[tbody.tr.td]]
 7  span = '11:40-13:10'
 8  div = '2, 4, 6, 8, 10, 12, 14, 16'
 9
10  [tbody.tr.td.dd]
11  text = '2306/2 бывш(. 305) ауд.'
12
13  [tbody.tr.td.dt]
14  span = 'Кронверкский пр., д.49, литА.'
15
16  [[tbody.tr.td]]
17
18  [tbody.tr.td.dl]
19  dd = '2306/2 бывш(. 305) ауд.'
20
21  [tbody.tr.td.dl.dt]
22  span = 'Кронверкский пр., д.49, литА.'
23
24  [[tbody.tr.td]]
25
26  [tbody.tr.td.dl]
27  dd = 'ИнформатикаЛаб()'
28  text = 'четная неделя'
29
30  [tbody.tr.td.dl.dt]
31  b = 'Малышева ТатьянаАлексеевна'
32
33  [[tbody.tr.td]]
34  text = 'Очно - дистанционный'
35
36  [tbody.tr.th]
37  span = 'Чт'
38
39  [[tbody.tr]]
40
41  [[tbody.tr.td]]
42  span = '13:30-15:00'
43  div = '2, 4, 6, 8, 10, 12, 14, 16'
44
45  [tbody.tr.td.dd]
46  text = '2306/2 бывш(. 305) ауд.'
47
48  [tbody.tr.td.dt]
49  span = 'Кронверкский пр., д.49, литА.'
```

```
50
51 [[tbody.tr.td]]
52
53 [tbody.tr.td.dl]
54 dd = '2306/2 бывш(. 305) ауд.'
55
56 [tbody.tr.td.dl.dt]
57 span = 'Кронверкский пр., д.49, литА.'
58
59 [[tbody.tr.td]]
60
61 [tbody.tr.td.dl]
62 dd = 'ИнформатикаЛаб()'
63 text = 'четная неделя'
64
65 [tbody.tr.td.dl.dt]
66 b = 'Малышева ТатьянаАлексеевна'
67
68 [[tbody.tr.td]]
69 text = 'Очно - дистанционный'
```

1.8 Итог

13. Проверить, что все пункты задания выполнены и выполнены верно.
14. Написать отчёт о проделанной работе.
15. Подготовиться к устным вопросам на защите

2 Вывод

Я познакомился с форматами файлов JSON, XML и написал парсер JSON на Python .