Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

img/itmo.png

Вариант №18 Лабораторная работа №4 по дисциплине Информатика

> Выполнил Студент группы Р3115 Владимир Мацюк Преподаватель: Малышева Татьяна Алексеевна

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

5. 1.2 Исходный файл

6. 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.

```
{
 1
     "tbody": {
 2
 3
       "tr": [
 4
          "th": {
 5
            "span": "Чт"
 6
          },
"td": [
 7
 8
 9
             "span": "11:40-13:10",
10
             "div": "2, 4, 6, 8, 10, 12, 14, 16",
11
             "dd": {
12
               "text": "2306/2 бывш(. 305) ауд."
13
14
             },
             "dt": {
15
               "span": "Кронверкский пр., д.49, литА."
16
             }
17
            },
18
19
             "dl": {
20
               "dd": "2306/2 бывш(. 305) ауд.",
21
               "dt": {
22
23
                 "span": "Кронверкский пр., д.49, литА."
24
25
             }
            },
{
26
27
             "dl": {
28
               "dd": "ИнформатикаЛаб()",
29
               "dt": {
30
                "b": "Малышева ТатьянаАлексеевна"
31
32
               "text": "четная неделя"
33
```

```
34
              }
            },
35
            {
36
              "text": "Очно - дистанционный"
37
38
            }
           ]
39
40
         },
41
           "td": [
42
43
            {
              "span": "13:30-15:00",
"div": "2, 4, 6, 8, 10, 12, 14, 16",
44
45
              "dd": {
46
                "text": "2306/2 бывш(. 305) ауд."
47
48
              },
              "dt": {
49
                "span": "Кронверкский пр., д.49, литА."
50
51
52
            },
53
            {
              "dl": {
54
                "dd": "2306/2 бывш(. 305) ауд.",
55
                "dt": {
56
57
                 "span": "Кронверкский пр., д.49, литА."
                }
58
              }
59
            },
60
            {
61
              "dl": {
62
                "dd": "ИнформатикаЛаб()",
63
                "dt": {
64
                 "b": "Малышева ТатьянаАлексеевна"
65
66
                "text": "четная неделя"
67
68
              }
            },
69
70
              "text": "Очно - дистанционный"
71
72
            }
73
74
         }
75
       ]
76
      }
77
    }
```

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

- 1 from dataclasses import dataclass
- 2 from enum import Enum, auto
- 3 from pathlib import Path

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

```
62 def parse_tokens(it: Iterator[Token]):
 63
        i = next(it)
 64
       match i.type:
 65
           case TokenType.STR:
              return i.val
 66
 67
           case TokenType.BEGIN_OBJ:
              res = \{\}
 68
 69
              i = next(it)
 70
              while True:
 71
                 if i.type ≠ TokenType.STR:
                    raise RuntimeError('expected str key')
 72
 73
                 key = i.val
 74
                 i = next(it)
 75
                 if i.type ≠ TokenType.COL:
 76
                    raise RuntimeError('expected ':'')
 77
                 val = parse_tokens(it)
 78
                 res[key] = val
 79
                 i = next(it)
 80
                 if i.type = TokenType.COMMA:
                    i = next(it)
 81
 82
                 elif i.type = TokenType.END_OBJ:
 83
 84
                 else:
                    raise RuntimeError('unexpected token')
 85
 86
              return res
 87
           case TokenType.BEGIN_ARR:
 88
              res = []
 89
              while True:
 90
                 val = parse_tokens(it)
 91
                 res.append(val)
 92
                 i = next(it)
 93
                 if i.type = TokenType.COMMA:
 94
                    continue
 95
                 elif i.type = TokenType.END_ARR:
 96
                    break
 97
                 else:
                    raise RuntimeError('unexpected token')
 98
 99
              return res
100
101
102 def parse_json(s: str):
       tokens = tokenise(s)
103
104
       res = parse_tokens(iter(tokens))
105
       return res
106
107
    def obj2xml(obj, deep=0, parent='root') -> str:
108
109
       sp = ' ' * deep
110
       match obj:
111
           case str(): return obj
112
           case dict():
              res = ''
113
              for (key, val) in obj.items():
114
115
                 match val:
116
                    case list():
                       res += f'{obj2xml(val, deep, key)}'
117
118
                    case str():
                       res += f'{sp}<{key}>{val}</{key}>\n'
119
```

```
120
                     case _:
121
                        res += f'{sp}<{key}>\n{obj2xml(val, deep+1, key)
                           \hookrightarrow }{sp}</{key}>\n'
122
              return res
123
           case list():
              return ''.join(f'{sp}<{parent}>\n{obj2xml(val, deep+1)}{
124
                  → sp}</{parent}>\n' for val in obj)
125
126
127 def task1(s: str):
128
        res = parse_json(s)
129
        return obj2xml(res)
```

out1.xml

```
1
2
    3
      4
       <span>4T</span>
5
     >
6
7
       <span>11:40-13:10
       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
8
9
10
        <text>2306/2 бывш(. 305) ауд.</text>
       </dd>
11
12
       <dt>
13
        <span>Кронверкский пр., д.49, литА.
14
       </dt>
15
      16
      17
       <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
18
19
20
          <span>Кронверкский пр., д.49, литА.
21
        </dt>
22
       </dl>
23
      24
      25
       <dl>
        <dd>ИнформатикаЛаб()</dd>
26
27
28
          <b>Малышева ТатьянаАлексеевна</b>
29
        </dt>
30
        <text>четная неделя</text>
31
       </dl>
      32
33
      34
       <text>0чно - дистанционный</text>
35
      36
    37
    38
      39
       <span>13:30-15:00</span>
40
       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
41
        <text>2306/2 бывш(. 305) ауд.</text>
42
43
       </dd>
44
       <dt>
```

```
45
        <span>Кронверкский пр., д.49, литА.
46
       </dt>
47
      48
      49
       <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
50
51
          <span>Кронверкский пр., д.49, литА.
52
53
        </dt>
       </dl>
54
55
      56
     57
       <dl>
        <dd>ИнформатикаЛаб()</dd>
58
59
          <b>>Малышева ТатьянаАлексеевна</b>
60
61
        </dt>
62
        <text>четная неделя</text>
63
       </dl>
      64
      65
       <text>0чно - дистанционный</text>
66
      67
68
    69
```

1.4 Дополнительное задание №1

- 9. Дополнительное задание №1 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
 - (а) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
 - (b) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
 - (с) Сравнить полученные результаты и объяснить их сходство/различие.

task2.py

```
import json
   from dict2xml import dict2xml
 3
 4
 5
   def parse_json(s: str):
 6
       return json.loads(s)
 7
 8
 9
   def obj2xml(s: str):
       return dict2xml(s)
10
11
12
13 def task2(s: str):
14
       res = parse_json(s)
       return obj2xml(res)
15
```

out2.xml

1

```
2
    3
     4
       <dd>
5
        <text>2306/2 бывш(. 305) ауд.</text>
6
       </dd>
       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
7
8
9
        <span>Кронверкский пр., д.49, литА.
10
       </dt>
       <span>11:40-13:10
11
12
     >
13
14
       <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
15
16
17
          <span>Кронверкский пр., д.49, литА.
18
        </dt>
19
       </dl>
20
     >
21
       <dl>
22
        <dd>ИнформатикаЛаб()</dd>
23
24
        <dt>
25
          <b>Малышева ТатьянаАлексеевна</b>
        </dt>
26
27
        <text>четная неделя</text>
28
       </dl>
29
     30
     31
       <text>0чно - дистанционный</text>
32
     33
     34
       <span>4T</span>
35
     36
    37
    38
     39
       <dd>
        <text>2306/2 бывш(. 305) ауд.</text>
40
41
       </dd>
       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
42
43
       <dt>
44
        <span>Кронверкский пр., д.49, литА.
45
       </dt>
46
       <span>13:30-15:00
     47
     48
       <dl>
49
        <dd>2306/2 бывш(. 305) ауд.</dd>
50
51
52
          <span>Кронверкский пр., д.49, литА.
53
        </dt>
       </dl>
54
     55
     56
57
       <dl>
        <dd>ИнформатикаЛаб()</dd>
58
59
        <dt>
```

```
60
         <b>Малышева ТатьянаАлексеевна</b>
61
        </dt>
62
        <text>четная неделя</text>
63
       </dl>
     64
     65
       <text>0чно - дистанционный</text>
66
     67
68
    69
```

1.5 Дополнительное задание №2

- 10. Дополнительное задание №2 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
 - (а) Переписать исходный код, добавив в него использование 2 регулярных выражений.
 - (b) Сравнить полученные результаты и объяснить их сходство/различие.

task3.py

```
1 import re
   from task1 import Token, TokenType, obj2xml, parse_tokens
 3
 5
   def tokenise(s: str):
 6
       return [
 7
          Token(match.group(match.lastindex), TokenType(match.
              \hookrightarrow lastindex + 1))
          for match in re.finditer('|'.join([
 8
             r'(\{)',
 9
             r'(\})',
10
             r'(\[)'
11
             r'(\])',
12
             r'"((?:\.|[^"])*)"',
13
14
             r'(:)',
15
             r'(,)',
16
          ]), s)
       ]
17
18
19
20 def parse_json(s: str):
       tokens = tokenise(s)
21
22
       res = parse_tokens(iter(tokens))
23
       return res
24
25
26 def task3(s: str):
27
       res = parse_json(s)
28
       return obj2xml(res)
```

out3.xml

```
1 
2 
3 4 <span>4T</span>
5
```

```
>
6
7
       <span>11:40-13:10
       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
8
9
10
        <text>2306/2 бывш(. 305) ауд.</text>
       </dd>
11
12
       <dt>
13
        <span>Кронверкский пр., д.49, литА.
14
       </dt>
      15
16
      17
       <dl>
        <dd>2306/2 бывш(. 305) ауд.</dd>
18
19
20
          <span>Кронверкский пр., д.49, литА.
21
        </dt>
22
       </dl>
      23
24
      >
25
       <dl>
        <dd>ИнформатикаЛаб()</dd>
26
27
          <b>Малышева ТатьянаАлексеевна</b>
28
29
        </dt>
30
        <text>четная неделя</text>
31
       </dl>
      32
33
      34
       <text>0чно - дистанционный</text>
35
      36
    37
    38
      <span>13:30-15:00
39
       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
40
41
        <text>2306/2 бывш(. 305) ауд.</text>
42
43
       </dd>
44
       <dt>
45
         <span>Кронверкский пр., д.49, литА.
46
       </dt>
47
      48
      49
       <dl>
50
        <dd>2306/2 бывш(. 305) ауд.</dd>
51
52
          <span>Кронверкский пр., д.49, литА.
53
        </dt>
54
       </dl>
55
      56
57
       <dl>
        <dd>ИнформатикаЛаб()</dd>
58
59
        <dt>
60
          <b>>Малышева ТатьянаАлексеевна</b>
61
         </dt>
        <text>четная неделя</text>
62
63
       </dl>
```

```
64 
65 
66 <text>Очно - дистанционный</text>
67 
68 
69
```

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 N1 и программу из дополнительного задания N2, сравнить стократное время выполнения парсинга + конвертации в цикле.
 - (b) Проанализировать полученные результаты и объяснить их сходство/различие.

task4.py

```
from pathlib import Path
 2
 3
 4
    def read():
 5
       return open(Path(__file__).with_name('input.json')).read()
 7
 8
    def write(idx: int, out: str, ext='xml'):
 9
       open(Path(__file__).with_name(f'out{idx}.{ext}'), mode="w+").
           → write(out)
10
11
12
   def task4():
       from task1 import task1
13
14
       from task2 import task2
       from task3 import task3
15
       from timeit import timeit
16
17
       s = read()
18
       write(1, task1(s))
19
20
       write(2, task2(s))
       write(3, task3(s))
21
22
       res = '\n'.join(
23
          s + str(t) for (s, t) in
24
25
              ('no lib + no regex: ', timeit("task1(s)", globals=locals
26
                 \hookrightarrow (), number=100)),
              ('lib: ', timeit("task2(s)", globals=locals(), number
27
                 \rightarrow =100)),
              ('regex: ', timeit("task3(s)", globals=locals(), number
28
                 \rightarrow =100))
29
          ]
30
       )
       open(Path(__file__).with_name(f'out4.txt'), mode="w+").write(
31
           \hookrightarrow res)
       print(res)
32
```

out4.txt

```
1 no lib + no regex: 0.06759362399952806
2 lib: 0.026628636000168626
3 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

```
from task2 import parse_json
   from task4 import read, write
 3
 4
 5
   def val2toml(item: any):
 6
      match item:
 7
          case str():
 8
             return item.__repr__()
 9
          case list():
             return f'[{", ".join(val2toml(i) for i in item)}]'
10
11
          case _:
             raise RuntimeError('forbidden type')
12
13
14
   def obj2toml(item: dict, parent=''):
15
16
      res = []
17
18
      def tables_at_end(item):
19
          _, value = item
20
          return isinstance(value, dict)
21
22
       for (key, val) in sorted(item.items(), key=tables_at_end):
          path = f"{parent}.{key}" if parent else key
23
24
          match val:
25
             case dict():
                res.append(f'\n[{path}]\n{obj2toml(val, path)}')
26
27
             case list():
28
                small = True
                if len(val) \neq 0:
29
30
                   if val[0] is dict:
                      small = False
31
32
                if small:
                   for i in val:
33
                      res.append(f'\n[[{path}]]\n{obj2toml(i, path)}')
34
35
                else:
                   res.append(f'\n[[{path}]]\n{obj2toml(val, path)}')
36
37
                res.append(f'{key} = {val2toml(val)}')
38
39
      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]
   [[tbody.tr]]
 6
   [[tbody.tr.td]]
   span = '11:40-13:10'
 7
 8 div = '2, 4, 6, 8, 10, 12, 14, 16'
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'
33 [[tbody.tr.td]]
34 text = 'Очно - дистанционный'
35
36 [tbody.tr.th]
37 \text{ span} = ' 4 \text{T}'
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'
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 .