

**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 Р3115
Владимир Мацюк
Преподаватель:
Малышева Татьяна Алексеевна

г. Санкт-Петербург
2022г.

Содержание

1 Задание	1
1.1 Вариант	1
1.2 Исходный файл	1
1.3 Обязательное задание	3
1.4 Дополнительное задание №1	6
1.5 Дополнительное задание №2	8
1.6 Дополнительное задание №3	10
1.7 Дополнительное задание №4	11
1.8 Итог	14
2 Вывод	14

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 {
2   "tbody": {
3     "tr": [
4       {
5         "th": {
6           "span": "ЧТ"
7         },
8         "td": [
9           {
10            "span": "11:40-13:10",
11            "div": "2, 4, 6, 8, 10, 12, 14, 16",
12            "dd": {
13              "text": "2306/2 бывш(. 305) ауд."
14            },
15            "dt": {
16              "span": "Кронверкский пр., д.49, литА."
17            }
18          },
19          {
20            "dl": {
21              "dd": "2306/2 бывш(. 305) ауд.",
```

```

22         "dt": {
23             "span": "Кронверкский пр., д.49, литА."
24         }
25     },
26 },
27 {
28     "dl": {
29         "dd": "ИнформатикаЛаб()",
30         "dt": {
31             "b": "Малышева ТатьянаАлексеевна"
32         },
33         "text": "четная неделя"
34     }
35 },
36 {
37     "text": "Очно - дистанционный"
38 }
39 ]
40 },
41 {
42     "td": [
43         {
44             "span": "13:30-15:00",
45             "div": "2, 4, 6, 8, 10, 12, 14, 16",
46             "dd": {
47                 "text": "2306/2 бывш(. 305) ауд."
48             },
49             "dt": {
50                 "span": "Кронверкский пр., д.49, литА."
51             }
52         },
53         {
54             "dl": {
55                 "dd": "2306/2 бывш(. 305) ауд.",
56                 "dt": {
57                     "span": "Кронверкский пр., д.49, литА."
58                 }
59             }
60         },
61         {
62             "dl": {
63                 "dd": "ИнформатикаЛаб()",
64                 "dt": {
65                     "b": "Малышева ТатьянаАлексеевна"
66                 },
67                 "text": "четная неделя"
68             }
69         },
70         {
71             "text": "Очно - дистанционный"
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
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
62 def parse_tokens(it: Iterator[Token]):
63     i = next(it)
64     match i.type:
65         case TokenType.STR:
66             return i.val
67         case TokenType.BEGIN_OBJ:
68             res = {}
69             i = next(it)
70             while True:
71                 if i.type != TokenType.STR:
72                     raise RuntimeError('expected str key')
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:
81                     i = next(it)
82                 elif i.type == TokenType.END_OBJ:
83                     break
84                 else:
85                     raise RuntimeError('unexpected token')
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:
98                     raise RuntimeError('unexpected token')
99             return res
100
101
102 def parse_json(s: str):
103     tokens = tokenise(s)
104     res = parse_tokens(iter(tokens))
105     return res
106
107

```

```

108 def obj2xml(obj, deep=0, parent='root') -> str:
109     sp = ' ' * deep
110     match obj:
111         case str(): return obj
112         case dict():
113             res = ''
114             for (key, val) in obj.items():
115                 match val:
116                     case list():
117                         res += f'{obj2xml(val, deep, key)}'
118                     case str():
119                         res += f'{sp}<{key}>{val}</{key}>\n'
120                     case _:
121                         res += f'{sp}<{key}>\n{obj2xml(val, deep+1, key)}
122                             ↪ </{key}>\n'
123             return res
124         case list():
125             return ''.join(f'{sp}<{parent}>\n{obj2xml(val, deep+1)}{
126                 ↪ sp}</{parent}>\n' for val in obj)
127
128 def task1(s: str):
129     res = parse_json(s)
130     return obj2xml(res)

```

out1.xml

```

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

```

```

33     <td>
34         <text>Очно - дистанционный</text>
35     </td>
36 </tr>
37 <tr>
38     <td>
39         <span>13:30-15:00</span>
40         <div>2, 4, 6, 8, 10, 12, 14, 16</div>
41         <dd>
42             <text>2306/2 бывш(. 305) ауд.</text>
43         </dd>
44         <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

```

1 import json
2 from dict2xml import dict2xml
3
4
5 def parse_json(s: str):
6     return json.loads(s)
7

```

```

8
9 def obj2xml(s: str):
10     return dict2xml(s)
11
12
13 def task2(s: str):
14     res = parse_json(s)
15     return obj2xml(res)

```

out2.xml

```

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

```

1.5 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.
 - (б) Сравнить полученные результаты и объяснить их сходство/различие.

task3.py

```

1 import re
2 from task1 import Token, TokenType, obj2xml, parse_tokens
3
4
5 def tokenise(s: str):
6     return [
7         Token(match.group(match.lastindex), TokenType(match.
8             ↪ lastindex + 1))
9         for match in re.finditer('|'.join([
10             r'(\{)',
11             r'(\})',
12             r'(\[)',
13             r'(\])',
14             r'"((?:\.|"[^"])*)"',
15             r'(:)',
16             r'(',')',
17         ]), s)
18     ]
19
20 def parse_json(s: str):
21     tokens = tokenise(s)
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 <tbody>
2   <tr>
3     <th>
4       <span>ЧТ</span>
5     </th>
6     <td>
7       <span>11:40-13:10</span>
8       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
9       <dd>
10        <text>2306/2 бывш(. 305) ауд.</text>
11      </dd>
12      <dt>
13        <span>Кронверкский пр., д.49, литА.</span>
14      </dt>
15    </td>
16    <td>
17      <dl>
18        <dd>2306/2 бывш(. 305) ауд.</dd>
19        <dt>
20          <span>Кронверкский пр., д.49, литА.</span>
21        </dt>
22      </dl>
23    </td>
24    <td>
25      <dl>
26        <dd>ИнформатикаЛаб()</dd>
27        <dt>
28          <b>Малышева ТатьянаАлексеевна</b>
29        </dt>
30        <text>четная неделя</text>
31      </dl>
32    </td>
33    <td>
34      <text>Очно - дистанционный</text>
35    </td>
36  </tr>
37  <tr>
38    <td>
39      <span>13:30-15:00</span>
40      <div>2, 4, 6, 8, 10, 12, 14, 16</div>
41      <dd>
42        <text>2306/2 бывш(. 305) ауд.</text>
43      </dd>
44      <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>

```

1.6 Дополнительное задание №3

11. Дополнительное задание №3 (позволяет набрать +10 процентов от максимального числа баллов БаРС за данную лабораторную).
 - (а) 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.
 - (б) Проанализировать полученные результаты и объяснить их сходство/различие.

task4.py

```

1 from pathlib import Path
2
3
4 def read():
5     return open(Path(__file__).with_name('input.json')).read()
6
7
8 def write(idx: int, out: str, ext='xml'):
9     open(Path(__file__).with_name(f'out{idx}.{ext}'), mode="w+").
10        ↪ write(out)
11
12 def task4():
13     from task1 import task1
14     from task2 import task2
15     from task3 import task3
16     from timeit import timeit
17     s = read()
18
19     write(1, task1(s))
20     write(2, task2(s))
21     write(3, task3(s))
22
23     res = '\n'.join(
24         s + str(t) for (s, t) in
25         [
26             ('no lib + no regex: ', timeit("task1(s)", globals=locals
27                 ↪ (), number=100)),

```

```

27         ('lib: ', timeit("task2(s)", globals=locals(), number
           ↪ =100)),
28         ('regex: ', timeit("task3(s)", globals=locals(), number
           ↪ =100))
29     ]
30 )
31 open(Path(__file__).with_name(f'out4.txt'), mode="w+").write(
    ↪ res)
32 print(res)

```

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

```

1 from task2 import parse_json
2 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():
10            return f'[{", ".join(val2toml(i) for i in item)}]\'
11        case _:
12            raise RuntimeError('forbidden type')
13
14
15 def obj2toml(item: dict, parent=''):
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):
23         path = f'{parent}.{key}' if parent else key
24         match val:
25             case dict():
26                 res.append(f'\n[{path}]\n{obj2toml(val, path)}\'
27             case list():
28                 small = True
29                 if len(val) != 0:

```

```

30         if val[0] is dict:
31             small = False
32         if small:
33             for i in val:
34                 res.append(f'\n[{path}]\n{obj2toml(i, path)}')
35         else:
36             res.append(f'\n[{path}]\n{obj2toml(val, path)}')
37         case _:
38             res.append(f'{key} = {val2toml(val)}')
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]
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 .