

**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

<b>1 Задание</b>	<b>1</b>
1.1 Вариант	1
1.2 Исходный файл	1
1.3 Обязательное задание	3
1.4 Дополнительное задание №1	7
1.5 Дополнительное задание №2	9
1.6 Дополнительное задание №3	11
1.7 Дополнительное задание №4	12
1.8 Итог	14
<b>2 Вывод</b>	<b>14</b>

## 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           "@class": "day",
7           "span": "ЧТ"
8         },
9         "td": [
10          {
11            "@class": "time",
12            "span": "11:40-13:10",
13            "div": "2, 4, 6, 8, 10, 12, 14, 16",
14            "dd": {
15              "@class": "rasp_aud_mobile",
16              "#text": "2306/2 бывш(. 305) ауд."
17            },
18            "dt": {
19              "@class": "rasp_corp_mobile",
20              "i": {
21                "@class": "fa fa-map-marker"
```

```

22         },
23         "span": "Кронверкский пр., д.49, литА."
24     }
25 },
26 {
27     "@class": "room",
28     "dl": {
29         "dd": "2306/2 бывш(. 305) ауд.",
30         "dt": {
31             "i": {
32                 "@class": "fa fa-map-marker"
33             },
34             "span": "Кронверкский пр., д.49, литА."
35         }
36     }
37 },
38 {
39     "@class": "lesson",
40     "dl": {
41         "dd": "ИнформатикаЛаб()",
42         "dt": {
43             "b": "Малышева ТатьянаАлексеевна"
44         },
45         "#text": "четная неделя"
46     }
47 },
48 {
49     "@class": "lesson-format",
50     "#text": "Очно - дистанционный"
51 }
52 ]
53 },
54 {
55     "th": {
56         "@class": "day"
57     },
58     "td": [
59         {
60             "@class": "time",
61             "span": "13:30-15:00",
62             "div": "2, 4, 6, 8, 10, 12, 14, 16",
63             "dd": {
64                 "@class": "rasp_aud_mobile",
65                 "#text": "2306/2 бывш(. 305) ауд."
66             },
67             "dt": {
68                 "@class": "rasp_corp_mobile",
69                 "i": {
70                     "@class": "fa fa-map-marker"
71                 },
72                 "span": "Кронверкский пр., д.49, литА."
73             }
74         },
75         {
76             "@class": "room",
77             "dl": {
78                 "dd": "2306/2 бывш(. 305) ауд.",
79                 "dt": {

```

```

80         "i": {
81             "@class": "fa fa-map-marker"
82         },
83         "span": "Кронверкский пр., д.49, литА."
84     }
85 }
86 },
87 {
88     "@class": "lesson",
89     "dl": {
90         "dd": "ИнформатикаЛаб()",
91         "dt": {
92             "b": "Малышева ТатьянаАлексеевна"
93         },
94         "#text": "четная неделя"
95     }
96 },
97 {
98     "@class": "lesson-format",
99     "#text": "Очно - дистанционный"
100 }
101 ]
102 }
103 ]
104 }
105 }

```

---

### 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 += obj2xml(val, deep+1, 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)}\n'
122             return res
123         case list():
124             return ''.join(f'{sp}<{parent}>{obj2xml(val, deep+1)}</{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 <tbody>
2   <tr> <th>
3     <@class>day</@class>

```

```

4      <span>Чт</span>
5  </th>
6      <td> <@class>time</@class>
7          <span>11:40-13:10</span>
8          <div>2, 4, 6, 8, 10, 12, 14, 16</div>
9          <dd>
10             <@class>rasp_aud_mobile</@class>
11             <#text>2306/2 бывш(. 305) ауд.</#text>
12         </dd>
13         <dt>
14             <@class>rasp_corp_mobile</@class>
15             <i>
16                 <@class>fa fa-map-marker</@class>
17             </i>
18             <span>Кронверкский пр., д.49, литА.</span>
19         </dt>
20 </td>
21 <td> <@class>room</@class>
22     <dl>
23         <dd>2306/2 бывш(. 305) ауд.</dd>
24         <dt>
25             <i>
26                 <@class>fa fa-map-marker</@class>
27             </i>
28             <span>Кронверкский пр., д.49, литА.</span>
29         </dt>
30     </dl>
31 </td>
32 <td> <@class>lesson</@class>
33     <dl>
34         <dd>ИнформатикаЛаб()</dd>
35         <dt>
36             <b>Малышева ТатьянаАлексеевна</b>
37         </dt>
38         <#text>четная неделя</#text>
39     </dl>
40 </td>
41 <td> <@class>lesson-format</@class>
42     <#text>Очно - дистанционный</#text>
43 </td>
44 </tr>
45 <tr> <th>
46     <@class>day</@class>
47 </th>
48 <td> <@class>time</@class>
49     <span>13:30-15:00</span>
50     <div>2, 4, 6, 8, 10, 12, 14, 16</div>
51     <dd>
52         <@class>rasp_aud_mobile</@class>
53         <#text>2306/2 бывш(. 305) ауд.</#text>
54     </dd>
55     <dt>
56         <@class>rasp_corp_mobile</@class>
57         <i>
58             <@class>fa fa-map-marker</@class>
59         </i>
60         <span>Кронверкский пр., д.49, литА.</span>
61     </dt>

```

```

62 </td>
63 <td> <@class>room</@class>
64 <dl>
65 <dd>2306/2 бывш(. 305) ауд.</dd>
66 <dt>
67 <i>
68 <@class>fa fa-map-marker</@class>
69 </i>
70 <span>Кронверкский пр., д.49, литА.</span>
71 </dt>
72 </dl>
73 </td>
74 <td> <@class>lesson</@class>
75 <dl>
76 <dd>ИнформатикаЛаб()</dd>
77 <dt>
78 <b>Малышева ТатьянаАлексеевна</b>
79 </dt>
80 <#text>четная неделя</#text>
81 </dl>
82 </td>
83 <td> <@class>lesson-format</@class>
84 <#text>Очно - дистанционный</#text>
85 </td>
86 </tr>
87 </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)

```

---



```

1 <tbody>
2   <tr>
3     <td>
4       <__class>time</__class>
5       <dd>
6         <__text>2306/2 бывш(. 305) ауд.</__text>
7         <__class>rasp_aud_mobile</__class>
8       </dd>
9       <div>2, 4, 6, 8, 10, 12, 14, 16</div>
10      <dt>
11        <__class>rasp_corp_mobile</__class>
12        <i>
13          <__class>fa fa-map-marker</__class>
14        </i>
15        <span>Кронверкский пр., д.49, литА.</span>
16      </dt>
17      <span>11:40-13:10</span>
18    </td>
19    <td>
20      <__class>room</__class>
21      <dl>
22        <dd>2306/2 бывш(. 305) ауд.</dd>
23        <dt>
24          <i>
25            <__class>fa fa-map-marker</__class>
26          </i>
27          <span>Кронверкский пр., д.49, литА.</span>
28        </dt>
29      </dl>
30    </td>
31    <td>
32      <__class>lesson</__class>
33      <dl>
34        <__text>четная неделя</__text>
35        <dd>ИнформатикаЛаб()</dd>
36        <dt>
37          <b>Малышева ТатьянаАлексеевна</b>
38        </dt>
39      </dl>
40    </td>
41    <td>
42      <__text>Очно - дистанционный</__text>
43      <__class>lesson-format</__class>
44    </td>
45    <th>
46      <__class>day</__class>
47      <span>Чт</span>
48    </th>
49  </tr>
50  <tr>
51    <td>
52      <__class>time</__class>
53      <dd>
54        <__text>2306/2 бывш(. 305) ауд.</__text>
55        <__class>rasp_aud_mobile</__class>
56      </dd>
57      <div>2, 4, 6, 8, 10, 12, 14, 16</div>

```

```

58     <dt>
59         <__class>rasp_corp_mobile</__class>
60         <i>
61             <__class>fa fa-map-marker</__class>
62         </i>
63         <span>Кронверкский пр., д.49, литА.</span>
64     </dt>
65     <span>13:30-15:00</span>
66 </td>
67 <td>
68     <__class>room</__class>
69     <dl>
70         <dd>2306/2 бывш(. 305) ауд.</dd>
71         <dt>
72             <i>
73                 <__class>fa fa-map-marker</__class>
74             </i>
75             <span>Кронверкский пр., д.49, литА.</span>
76         </dt>
77     </dl>
78 </td>
79 <td>
80     <__class>lesson</__class>
81     <dl>
82         <__text>четная неделя</__text>
83         <dd>ИнформатикаЛаб(</dd>
84         <dt>
85             <b>Малышева ТатьянаАлексеевна</b>
86         </dt>
87     </dl>
88 </td>
89 <td>
90     <__text>Очно - дистанционный</__text>
91     <__class>lesson-format</__class>
92 </td>
93 <th>
94     <__class>day</__class>
95 </th>
96 </tr>
97 </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.
           ↳ lastindex + 1))
8     for match in re.finditer('|'.join([
9         r'(\{)',
10        r'(\})',
11        r'(\[)',
12        r'(\])',
13        r'''((?:\.|["'])*)''' ,
14        r'(:)',
15        r'(',')',
16    ]), s)
17 ]
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> <th>
3     <@class>day</@class>
4     <span>ЧТ</span>
5   </th>
6   <td> <@class>time</@class>
7     <span>11:40-13:10</span>
8     <div>2, 4, 6, 8, 10, 12, 14, 16</div>
9     <dd>
10       <@class>rasp_aud_mobile</@class>
11       <#text>2306/2 бывш(. 305) ауд.</#text>
12     </dd>
13     <dt>
14       <@class>rasp_corp_mobile</@class>
15       <i>
16         <@class>fa fa-map-marker</@class>
17       </i>
18       <span>Кронверкский пр., д.49, литА.</span>
19     </dt>
20   </td>
21   <td> <@class>room</@class>
22     <dl>
23       <dd>2306/2 бывш(. 305) ауд.</dd>
24       <dt>
25         <i>
26           <@class>fa fa-map-marker</@class>
27         </i>
28         <span>Кронверкский пр., д.49, литА.</span>
29       </dt>
30     </dl>
31   </td>
32   <td> <@class>lesson</@class>
33     <dl>

```

```

34         <dd>ИнформатикаЛаб()</dd>
35         <dt>
36             <b>Малышева ТатьянаАлексеевна</b>
37         </dt>
38         <#text>четная неделя</#text>
39     </dl>
40 </td>
41     <td> <@class>lesson-format</@class>
42         <#text>Очно - дистанционный</#text>
43 </td>
44 </tr>
45     <tr> <th>
46         <@class>day</@class>
47     </th>
48     <td> <@class>time</@class>
49         <span>13:30-15:00</span>
50         <div>2, 4, 6, 8, 10, 12, 14, 16</div>
51     <dd>
52         <@class>rasp_aud_mobile</@class>
53         <#text>2306/2 бывш(. 305) ауд.</#text>
54     </dd>
55     <dt>
56         <@class>rasp_corp_mobile</@class>
57         <i>
58             <@class>fa fa-map-marker</@class>
59         </i>
60         <span>Кронверкский пр., д.49, литА.</span>
61     </dt>
62 </td>
63     <td> <@class>room</@class>
64     <dl>
65         <dd>2306/2 бывш(. 305) ауд.</dd>
66         <dt>
67             <i>
68                 <@class>fa fa-map-marker</@class>
69             </i>
70             <span>Кронверкский пр., д.49, литА.</span>
71         </dt>
72     </dl>
73 </td>
74     <td> <@class>lesson</@class>
75     <dl>
76         <dd>ИнформатикаЛаб()</dd>
77         <dt>
78             <b>Малышева ТатьянаАлексеевна</b>
79         </dt>
80         <#text>четная неделя</#text>
81     </dl>
82 </td>
83     <td> <@class>lesson-format</@class>
84         <#text>Очно - дистанционный</#text>
85 </td>
86 </tr>
87 </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

```
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)),
28             ('lib: ', timeit("task2(s)", globals=locals(), number
29             ↪ =100)),
30             ('regex: ', timeit("task3(s)", globals=locals(), number
31             ↪ =100))
32         ]
33     )
34     open(Path(__file__).with_name(f'out4.txt'), mode="w+").write(
35         ↪ res)
36     print(res)
```

### out4.txt

```
1 no lib + no regex: 0.1315980129984382
2 lib: 0.047403575998032466
3 regex: 0.05752121200202964
```

## 1.7 Дополнительное задание №4

12. Дополнительное задание №4 (позволяет набрать +5 процентов от максимального числа баллов BaPC 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     print(item)
7     match item:
8         case str():
9             return item.__repr__()
10        case list():
11            return f'[{", ".join(val2toml(i) for i in item)}]'
12        case _:
13            raise RuntimeError('forbidden type')
14
15
16 def obj2toml(item: dict, parent=''):
17     res = []
18
19     def tables_at_end(item):
20         _, value = item
21         return isinstance(value, dict)
22
23     for (key, val) in sorted(item.items(), key=tables_at_end):
24         match val:
25             case dict():
26                 path = f'{{parent}}.{{key}}' if parent else key
27                 res.append(f'\n[{{path}}]\n{obj2toml(val, path)}')
28             case _:
29                 res.append(f'{{key}} = {val2toml(val)}')
30     return '\n'.join(res)
31
32
33 def task5():
34     s = read()
35     d = parse_json(s)
36     res = obj2toml(d)
37     write(5, res, 'toml')
```

### out5.toml

```
1
2 [glossary]
3 title = 'example glossary'
4
5 [glossary.GlossDiv]
6 title = 'S'
```

```

7
8 [glossary.GlossDiv.GlossList]
9
10 [glossary.GlossDiv.GlossList.GlossEntry]
11 ID = 'SGML'
12 SortAs = 'SGML'
13 GlossTerm = 'Standard Generalized Markup Language'
14 Acronym = 'SGML'
15 Abbrev = 'ISO 8879:1986'
16 GlossSee = 'markup'
17
18 [glossary.GlossDiv.GlossList.GlossEntry.GlossDef]
19 para = 'A meta-markup language, used to create markup languages
        ↪ such as DocBook.'
20 GlossSeeAlso = ['GML', 'XML']

```

---

## 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

sdfsdf