

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
науки и высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №5
по дисциплине
"Интеграция кроссплатформенных программных
систем"
Тема: "Определение структуры JSON-документа
средствами JSON Schema. Программная обработка
JSON"

Выполнил:
ст. гр. ИСТм-121
В.Г. Хлызова

Принял:
Спирин И.В.

Владимир, 2022 г.

ЦЕЛЬ РАБОТЫ

Данная работа предусматривает разработку программы, использующей один из рассмотренных инструментов для обработки XML-документа. Обработка должна заключаться в формировании JSON-представления для XML-документа, поданного на вход программы. Структура JSON-документа должна быть предварительно определена с помощью JSON Schema. В процессе обработки должно изменяться содержимое и структура полученного при анализе графа объектов, JSON-представление должно быть сформировано с учётом изменений.

ХОД РАБОТЫ

Предметная область: проектирование информационных систем.

[Ключевые элементы BPMN моделей.](#)

Технические требования

- 1) предметная область должна предусматривать наличие данных трех типов: строкового, числового и типа дата
- 2) структура и наполнение XML-документа должны быть такими, чтобы часть данных можно было представить в табличной форме, и при этом в каждой строке таблицы было как минимум два числовых поля и как минимум одно поле типа дата
- 3) результат формирования HTML-представления должен содержать в табличной части вычисляемое поле, формируемое на основе данных строки; вычисление может быть любым, но осмысленным в терминах предметной области
- 4) результат формирования HTML-представления должен содержать в табличной части подвал с как минимум одним итогом; итоги могут вычисляться любым способом, но осмысленно в терминах предметной области
- 5) ввод каждая из программ должна брать со стандартного ввода, результат преобразования передавать на стандартный вывод

6) каждая из программ должна обеспечить обработку ошибок анализа документа соответствующими средствами, при этом в случае ошибки уровня error и fatal error анализ должен быть прекращен без выдачи результата преобразования на стандартный вывод, а при возникновении ошибки уровня warning анализ должен быть продолжен, а преобразование - выполнено; в любом случае при возникновении ошибки любого уровня необходимо на стандартный вывод об ошибках выдать информацию об уровне ошибки, системное сообщение об ошибке и место возникновения ошибки с указанием строки и столбца документа

7) при анализе документа необходимо выполнить проверку соответствия документа схеме, при этом предполагается, что файл со схемой поставляется вместе с программой и должен быть расположен в каталоге программы, из которого она и запускается, а анализируемый файл может и вовсе не содержать ссылки на схему

Ссылка на репозиторий:

https://github.com/arranay/Integration_of_cross_platform_software_systems

JSON schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "author": {
      "type": "string"
    },
    "date": {
      "type": "string"
    },
    "tracks": {
      "type": "array",
      "items": [
        {
          "type": "object",
          "properties": {
```

```
"id": {
  "type": "string"
},
"name": {
  "type": "string"
},
"events": {
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "id": {
          "type": "string"
        },
        "name": {
          "type": "string"
        },
        "type": {
          "type": "string"
        }
      }
    }
  ],
  "required": [
    "id",
    "name",
    "type"
  ]
}
],
},
"actions": {
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "id": {
          "type": "string"
        },
        "operation": {
          "type": "string"
        },
        "time": {
          "type": "integer"
        }
      }
    }
  ],
  "required": [
    "id",
    "operation",
    "time"
  ]
}
```

```

    },
    {
      "type": "object",
      "properties": {
        "id": {
          "type": "string"
        },
        "operation": {
          "type": "string"
        },
        "time": {
          "type": "integer"
        }
      },
      "required": [
        "id",
        "operation",
        "time"
      ]
    },
    {
      "type": "object",
      "properties": {
        "id": {
          "type": "string"
        },
        "operation": {
          "type": "string"
        },
        "time": {
          "type": "integer"
        }
      },
      "required": [
        "id",
        "operation",
        "time"
      ]
    }
  ],
  "forks": {
    "type": "array",
    "items": [
      {
        "type": "object",
        "properties": {
          "id": {
            "type": "string"
          },
          "condition": {

```

```

        "type": "string"
    }
},
    "required": [
        "id",
        "condition"
    ]
}
]
}
},
    "required": [
        "id",
        "name",
        "events",
        "actions",
        "forks"
    ]
}
]
},
    "streams": {
        "type": "array",
        "items": [
            {
                "type": "object",
                "properties": {
                    "id": {
                        "type": "string"
                    },
                    "from": {
                        "type": "object",
                        "properties": {
                            "id": {
                                "type": "string"
                            },
                            "type": {
                                "type": "string"
                            },
                            "itemId": {
                                "type": "string"
                            }
                        }
                    },
                    "required": [
                        "id",
                        "type",
                        "itemId"
                    ]
                },
                "to": {
                    "type": "object",

```

```

        "properties": {
            "id": {
                "type": "string"
            },
            "type": {
                "type": "string"
            },
            "itemId": {
                "type": "string"
            }
        },
        "required": [
            "id",
            "type",
            "itemId"
        ],
        "name": {
            "type": "null"
        }
    },
    "required": [
        "id",
        "from",
        "to",
        "name"
    ]
}
]
}
},
"required": [
    "name",
    "author",
    "date",
    "tracks",
    "streams"
]
}

```

JSON:

```

{
  "name" : "Example",
  "author" : "Valeria Khlyzova",
  "date" : "2022-06-06",
  "tracks" : [ {
    "id" : "track1",
    "name" : "Customer",
    "events" : [ {

```

```

        "id" : "event11",
        "name" : "Start",
        "type" : "start"
    } ],
    "actions" : [ {
        "id" : "action11",
        "operation" : "Order and pay for the product",
        "time" : 5000
    } ],
    "forks" : [ ]
}, {
    "id" : "track2",
    "name" : "Seller",
    "events" : [ {
        "id" : "event21",
        "name" : "End",
        "type" : "end"
    } ],
    "actions" : [ {
        "id" : "action21",
        "operation" : "Check the product in storehouse",
        "time" : 7000
    } ],
    {
        "id" : "action22",
        "operation" : "Send the ordered product",
        "time" : 5000
    },
    {
        "id" : "action23",
        "operation" : "Return money",
        "time" : 3000
    } ],
    "forks" : [ {
        "id" : "fork21",
        "condition" : "is the product in storehouse?"
    } ]
} ],
"streams" : [ {
    "id" : "stream1",
    "from" : {
        "id" : "from1",
        "type" : "event",
        "itemId" : "event11"
    },
    "to" : {
        "id" : "to1",
        "type" : "action",
        "itemId" : "action11"
    },
    "name" : null
}, {
    "id" : "stream2",

```



```

    "from" : {
      "id" : "from2",
      "type" : "action",
      "itemId" : "action11"
    },
    "to" : {
      "id" : "to2",
      "type" : "action",
      "itemId" : "action21"
    },
    "name" : null
  }, {
    "id" : "stream3",
    "from" : {
      "id" : "from3",
      "type" : "action",
      "itemId" : "action21"
    },
    "to" : {
      "id" : "to3",
      "type" : "fork",
      "itemId" : "fork21"
    },
    "name" : null
  }, {
    "id" : "stream4",
    "from" : {
      "id" : "from4",
      "type" : "fork",
      "itemId" : "fork21"
    },
    "to" : {
      "id" : "to4",
      "type" : "action",
      "itemId" : "action23"
    },
    "name" : "no"
  }, {
    "id" : "stream5",
    "from" : {
      "id" : "from5",
      "type" : "fork",
      "itemId" : "fork21"
    },
    "to" : {
      "id" : "to5",
      "type" : "action",
      "itemId" : "action22"
    },
    "name" : "yes"
  }, {

```

```

    "id" : "stream6",
    "from" : {
      "id" : "from6",
      "type" : "action",
      "itemId" : "action22"
    },
    "to" : {
      "id" : "to6",
      "type" : "event",
      "itemId" : "event21"
    },
    "name" : null
  }, {
    "id" : "stream7",
    "from" : {
      "id" : "from7",
      "type" : "action",
      "itemId" : "action23"
    },
    "to" : {
      "id" : "to7",
      "type" : "event",
      "itemId" : "event21"
    },
    "name" : null
  } ]
}

```

Результат работы программы:

Changed name

+

×

—

⌵

→

↶

↷

🏠

Файл

|

C:/Users/valer/Desktop/labs/Integration_of_cross_platform_software_systems/lab5/lab5/src/main/resources/result.html

🔍

📄

☆

⚙️

🖼️

👤

⋮

Common info about BPMN model

Model name:

Changed name

Model author:

Changed author

Date:

2022-06-13

Tracks

Customer

Actions

Operation

Order and pay for the product

Time

5000

Events

Name

Type

Start

start

Seller

Actions

Operation

Check the product in storehouse

Time

7000

Send the ordered product

5000

Return money

3000

Events

Name

Type

End

end

Forks

Condition

is the product in storehouse?

Result time

20000 ms

Streams

From

To

Name

Start

Order and pay for the product

Order and pay for the product

Check the product in storehouse

Check the product in storehouse

is the product in storehouse?

is the product in storehouse?

Return money

no

is the product in storehouse?

Send the ordered product

yes

Send the ordered product

End

Return money

End

Рисунок 1 - Результат работы программы.

ВЫВОД

В ходе выполнения лабораторной работы были изучены определение структуры JSON-документа средствами JSON Schema и программная обработка JSON.