|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 «Прикладная информатика»**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 10 |

**Название:**

Сбор данных со сторонних веб-сайтов. Тесты.

**Дисциплина:** Языки Интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-34Б |  | 05.12.2022 | С.А. Рахманов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д. В. Малахов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы:** Получить практические навыки формирования данных в формате XML и их визуализации с  
помощью клиентских и серверных средств с использованием XSLT-преобразований.

**Задание:**

Модифицировать код ЛР 8 таким образом, чтобы по запросу с указанными параметрами выдавался результат в формате XML (средствами стандартной сериализации ActiveSupport).

* Проверить формирование XML и сохранить в файл для отладки XSLT и второго приложения.
* Написать функциональный тест, проверяющий формат выдаваемых данных при запросе RSS.

Разработать XSLT-программу преобразования полученной XML в HTML.

Добавить в проверяемый XML-файл строку привязки к преобразованию <?xml-stylesheet type="text/xsl" href="some\_transformer.xslt"?>. Проверить корректность отображения браузером результата преобразования.

Проверить на автономной Ruby-программе корректность преобразования, используя следующий фрагмент кода:

require 'nokogiri'

doc = Nokogiri::XML(File.read('some\_file.xml'))

xslt = Nokogiri::XSLT(File.read('some\_transformer.xslt'))

puts xslt.transform(doc)

Разработать второе приложение, являющееся посредником между клиентом и первым приложением, задачей которого является преобразование XML в HTML или передача в неизменном виде браузеру для отображения браузером. Приложение должно запускаться с указанием номера порта TCP, отличным от номера порта первого приложения (например rails server -p 3001)!

* Подготовить каркас приложения, а также форму формирования запроса, форму отображения результата и соответствующие действия контролера.
* Добавить в контроллер преобразование XML в HTML с помощью ранее разработанного XSLT-файла.
* Подключить запрос XML с первого приложения и проверить работу приложений в связке.
* Написать функциональный тест, проверяющий что при различных входных данных результат генерируемой страницы различен.
* Доработать код контроллера и представлений данного приложения для выдачи браузеру XML-потока в неизменном виде (организовать возможность выбора формата выдачи для пользователя).
* Проверить, что браузер получает XML первого приложения в неизменном виде.
* Доработать код контроллера приложения таким образом, чтобы XML-поток первого приложения получал дополнительную строку, указывающую xsl. Модифицировать форму запроса параметров таким образом, чтобы браузер получал в ответ XML. При этом разместить XSLT-файл в директории public.
* Проверить, что браузер производит преобразование XML->HTML в соответствии с xlt.
* Реализовать функциональные тесты второго приложения. Проверить результаты, формируемые приложением, на соответствие выбранному формату выдачи.

Итоговая форма ввода параметра должна содержать кнопки или селектор, позволяющие проверить два варианта преобразования:

* Серверное xml+xslt->html
* Клиентское xml+xslt->html

**Тексты измененных файлов**

logic

***seq\_controller.rb***

# frozen\_string\_literal: true

require 'net/http'

# Top level class documentation

class SeqController < ApplicationController

def input; end

def view

return unless params[:str]

arr = params[:str]&.split&.map(&:to\_i)

if arr && !arr&.empty?

tmp = helpers.find\_seqs(arr)

@found = tmp.map { |x| x.join(' ') }.join(' || ')

@max = helpers.find\_max(tmp)

else

@max = @str

@found = @str

end

# генерация ссылки на второй сервер

url = URI.parse("http://localhost:3001/trans/render\_xml.#{params[:where] == 'client' ? 'xml' : 'html'}?&arr[]=#{arr&.join('&arr[]=')}&max[]=#{@max&.join('&max[]=')}&found=#{@found}")

# отправка запроса на сервер с возвратом ответа (сама переменная)

temp = []

temp << arr

temp << @max

temp << @found

if params[:where] == 'server'

res = Net::HTTP.get\_response(url)

if params[:what] == 'source'

render plain: res.body

else

render html: res.body.html\_safe, layout: false

end

else

@xml = Nokogiri::XML(temp.to\_xml)

@xml.root.add\_previous\_sibling Nokogiri::XML::ProcessingInstruction.new(@xml, 'xml-stylesheet',

'type="text/xsl" href="/xml\_styler.xslt"')

if params[:what] == 'source'

render plain: @xml, layout: false

else

respond\_to do |format|

format.xml { render xml: @xml.to\_s }

end

end

end

end

end

***seq\_helper.rb***

# frozen\_string\_literal: true

# top-level documentation

module SeqHelper

def find\_seqs(arr)

found = []

tmp\_found = []

arr.take(arr.size - 1).each\_index do |i|

tmp\_found << arr[i]

unless arr[i] < arr [i + 1]

found << tmp\_found

tmp\_found = []

end

end

tmp\_found << arr[-1]

found << tmp\_found

found

end

def find\_max(alr\_found)

sizes = alr\_found.map(&:size)

alr\_found[sizes.index(sizes.max)]

end

end

***application.html.erb***

<!DOCTYPE html>

<html>

<head>

<title>ЛР №10</title>

<meta name="viewport" content="width=device-width,initial-scale=1">

<%= csrf\_meta\_tags %>

<%= csp\_meta\_tag %>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>

<%= stylesheet\_link\_tag "application", "data-turbo-track": "reload" %>

<%= javascript\_importmap\_tags %>

</head>

<body class="bg-light">

<div class="col-md-6 mx-auto my-4 p-5 shadow">

<h1 class="h1">Поиск возрастающих последовательностей</h1>

<p>Вывести введенную цепочку, все найденные последовательности и наиболее длинную из них.</p>

<%= yield %>

</div>

</body>

</html>

***input.html.erb***

<%= form\_with url: '/seq/view.xml', method: :get do |form| %>

<%= form.label('text-input', 'Введите число: ', class: 'form-check-label') %>

<%= form.text\_field :str, value: '1 2 3 4 3 4 5 1 5 3 4 5 6', class: 'form-control',id: 'text-input' %>

<br/>

<p>Где провести рендер HTML:</p>

<%= form.radio\_button :where, :server, class: 'form-check-input', required: 'required' %>

<%= form.label :where\_server, 'На стороне сервера', class: 'form-check-label' %>

<br/>

<%= form.radio\_button :where, :client, class: 'form-check-input', required: 'required' %>

<%= form.label :where\_client, 'На стороне клиента', class: 'form-check-label' %>

<br/>

<br/>

<p>Формат вывода:</p>

<%= form.radio\_button :what, :source, class: 'form-check-input', required: 'required' %>

<%= form.label :what\_source, 'Исходный код', class: 'form-check-label' %>

<br/>

<%= form.radio\_button :what, :transformed, class: 'form-check-input', required: 'required' %>

<%= form.label :what\_transformed, 'HTML страница', class: 'form-check-label' %>

<br/>

<br/>

<%= form.submit('Вычислить', class: 'btn btn-primary') %>

<% end %>

***view.html.erb***

<%= "text" %>

***routes.rb***

# frozen\_string\_literal: true

Rails.application.routes.draw do

get 'seq/input'

get 'seq/view'

# Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html

# root 'seq#input', as: :home

root 'seq#input', as: :home

# Defines the root path route ("/")

# root "articles#index"

end

***xml\_styler.xslt***

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

<xsl:template match="/">

<html>

<head>

<title>Лаб №10 Клиент</title>

<meta name="viewport" content="width=device-width,initial-scale=1"/>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous"></link>

</head>

<body class="bg-light">

<div class="col-md-9 mx-auto bg-white my-4 p-5 shadow">

<h1 class="h1">Поиск возрастающих последовательностей</h1>

<p>Вывести введенную цепочку, все найденные последовательности и наиболее длинную из них.</p>

<table class="table table-bordered my-2">

<tr>

<td class="text-center" style="width: fit-content">Исходная</td>

<td class="text-center" style="width: fit-content">Максимальная</td>

<td class="text-center" style="width: fit-content">Все</td>

</tr>

<tr>

<xsl:for-each select="objects/object">

<td class="text-center" style="width: fit-content">

<xsl:value-of select="current()" />

</td>

</xsl:for-each>

</tr>

</table>

<a class="btn btn-primary" href="/">Назад</a>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

***seq\_controller\_test.rb***

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

<xsl:template match="/">

<html>

<head>

<title>Лаб №10 Клиент</title>

<meta name="viewport" content="width=device-width,initial-scale=1"/>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous"></link>

</head>

<body class="bg-light">

<div class="col-md-9 mx-auto bg-white my-4 p-5 shadow">

<h1 class="h1">Поиск возрастающих последовательностей</h1>

<p>Вывести введенную цепочку, все найденные последовательности и наиболее длинную из них.</p>

<table class="table table-bordered my-2">

<tr>

<td class="text-center" style="width: fit-content">Исходная</td>

<td class="text-center" style="width: fit-content">Максимальная</td>

<td class="text-center" style="width: fit-content">Все</td>

</tr>

<tr>

<xsl:for-each select="objects/object">

<td class="text-center" style="width: fit-content">

<xsl:value-of select="current()" />

</td>

</xsl:for-each>

</tr>

</table>

<a class="btn btn-primary" href="/">Назад</a>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

trans

***transformed\_controller.rb***

# frozen\_string\_literal: true

require 'net/http'

require 'nokogiri'

# top-level documentation

class TransController < ApplicationController

def input; end

def render\_xml

return unless params[:arr] && params[:max]

tmp = []

tmp << params[:arr].map(&:to\_i)

tmp << params[:max].map(&:to\_i)

tmp << params[:found]

@xml = Nokogiri::XML(tmp.to\_xml)

@xml.root.add\_previous\_sibling Nokogiri::XML::ProcessingInstruction.new(@xml, 'xml-stylesheet',

'type="text/xsl" href="/xml\_styler.xslt"')

respond\_to do |format|

format.xml { render xml: @xml, layout: false }

format.html do

xslt = Nokogiri::XSLT(IO.read("#{Rails.root}/public/xml\_styler.xslt"))

result = Nokogiri::HTML5(xslt.transform(@xml).to\_s)

render xml: result.to\_s

end

end

end

end

***render.html.erb***

<h1>trans#render</h1>

<p>Find me in app/views/trans/render.html.erb</p>

***puma.rb***

# frozen\_string\_literal: true

# Puma can serve each request in a thread from an internal thread pool.

# The `threads` method setting takes two numbers: a minimum and maximum.

# Any libraries that use thread pools should be configured to match

# the maximum value specified for Puma. Default is set to 5 threads for minimum

# and maximum; this matches the default thread size of Active Record.

#

max\_threads\_count = ENV.fetch('RAILS\_MAX\_THREADS', 5)

min\_threads\_count = ENV.fetch('RAILS\_MIN\_THREADS') { max\_threads\_count }

threads min\_threads\_count, max\_threads\_count

# Specifies the `worker\_timeout` threshold that Puma will use to wait before

# terminating a worker in development environments.

#

worker\_timeout 3600 if ENV.fetch('RAILS\_ENV', 'development') == 'development'

# Specifies the `port` that Puma will listen on to receive requests; default is 3000.

#

port ENV.fetch('PORT', 3001)

# Specifies the `environment` that Puma will run in.

#

environment ENV.fetch('RAILS\_ENV', 'development')

# Specifies the `pidfile` that Puma will use.

pidfile ENV.fetch('PIDFILE', 'tmp/pids/server.pid')

# Specifies the number of `workers` to boot in clustered mode.

# Workers are forked web server processes. If using threads and workers together

# the concurrency of the application would be max `threads` \* `workers`.

# Workers do not work on JRuby or Windows (both of which do not support

# processes).

#

# workers ENV.fetch("WEB\_CONCURRENCY") { 2 }

# Use the `preload\_app!` method when specifying a `workers` number.

# This directive tells Puma to first boot the application and load code

# before forking the application. This takes advantage of Copy On Write

# process behavior so workers use less memory.

#

# preload\_app!

# Allow puma to be restarted by `bin/rails restart` command.

plugin :tmp\_restart

***routes.rb***

# frozen\_string\_literal: true

Rails.application.routes.draw do

# Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html

get 'trans/render\_xml'

# get 'trans/input'

# Defines the root path route ("/")

# root "articles#index"

end

***xml\_styler.xslt***

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

<xsl:template match="/">

<html>

<head>

<title>Лаб №10 Сервер</title>

<meta name="viewport" content="width=device-width,initial-scale=1"/>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous"></link>

</head>

<body class="bg-light">

<div class="col-md-9 mx-auto bg-white my-4 p-5 shadow">

<h1 class="h1">Поиск возрастающих последовательностей</h1>

<p>Вывести введенную цепочку, все найденные последовательности и наиболее длинную из них.</p>

<table class="table table-bordered my-2">

<tr>

<td class="text-center" style="width: fit-content">Исходная</td>

<td class="text-center" style="width: fit-content">Максимальная</td>

<td class="text-center" style="width: fit-content">Все</td>

</tr>

<tr>

<xsl:for-each select="objects/object">

<td class="text-center" style="width: fit-content">

<xsl:value-of select="current()" />

</td>

</xsl:for-each>

</tr>

</table>

<a class="btn btn-primary" href="/">Назад</a>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

***trans\_controller\_test.rb***

# frozen\_string\_literal: true

require 'test\_helper'

class TransControllerTest < ActionDispatch::IntegrationTest

test 'should get render' do

get 'http://localhost:3001/trans/render\_xml', params: { arr: (1..9).to\_a, max: (1..9).to\_a, found: (1..9).to\_s, format: 'html' }

assert\_response :success

end

test 'should render html' do

get 'http://localhost:3001/trans/render\_xml', params: { arr: (1..9).to\_a, max: (1..9).to\_a, found: (1..9).to\_s, format: 'html' }

assert\_empty(Nokogiri::HTML(@response.body).errors)

end

test 'should render xml' do

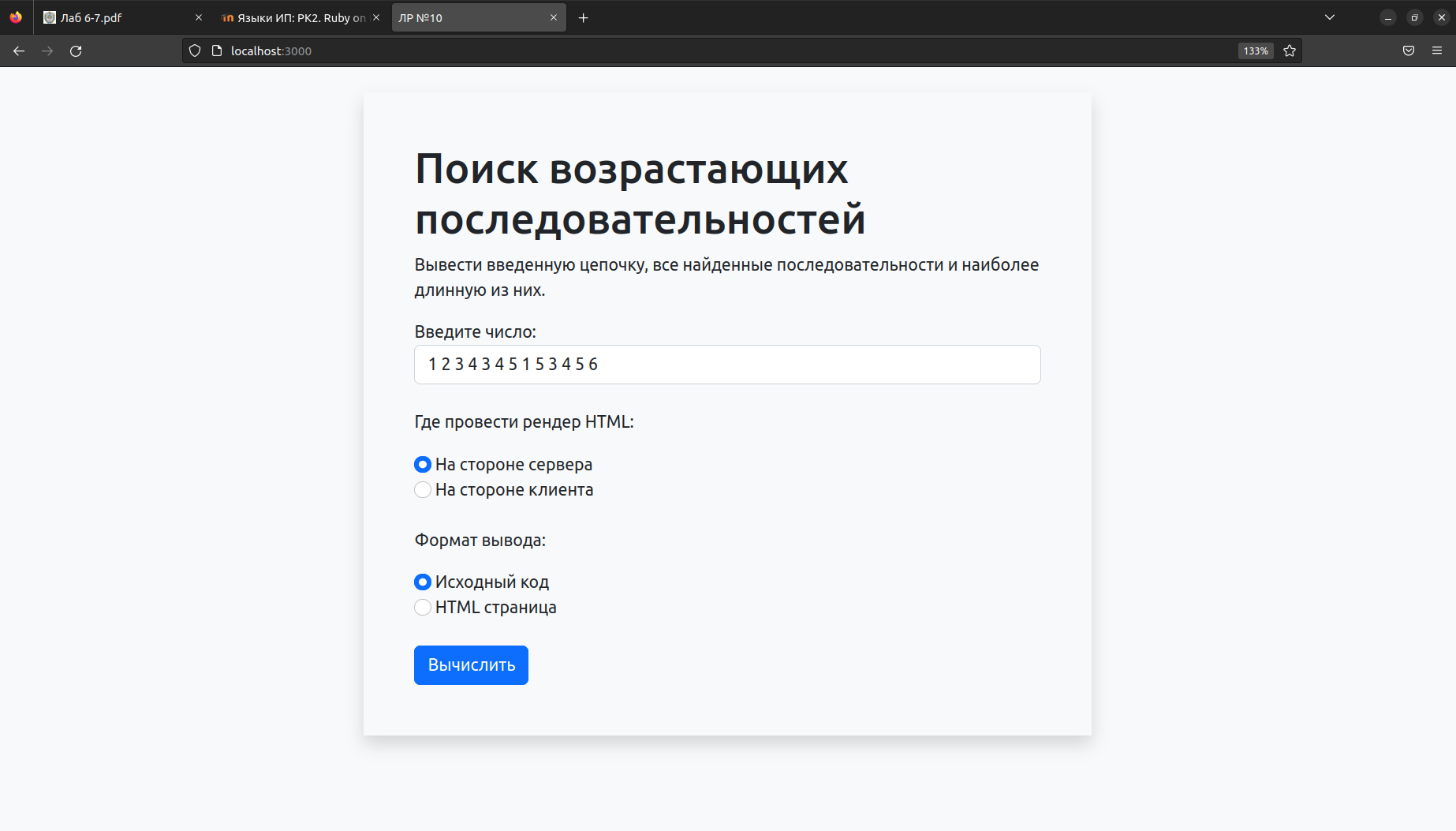
get 'http://localhost:3001/trans/render\_xml', params: { arr: (1..9).to\_a, max: (1..9).to\_a, found: (1..9).to\_s, format: 'xml' }

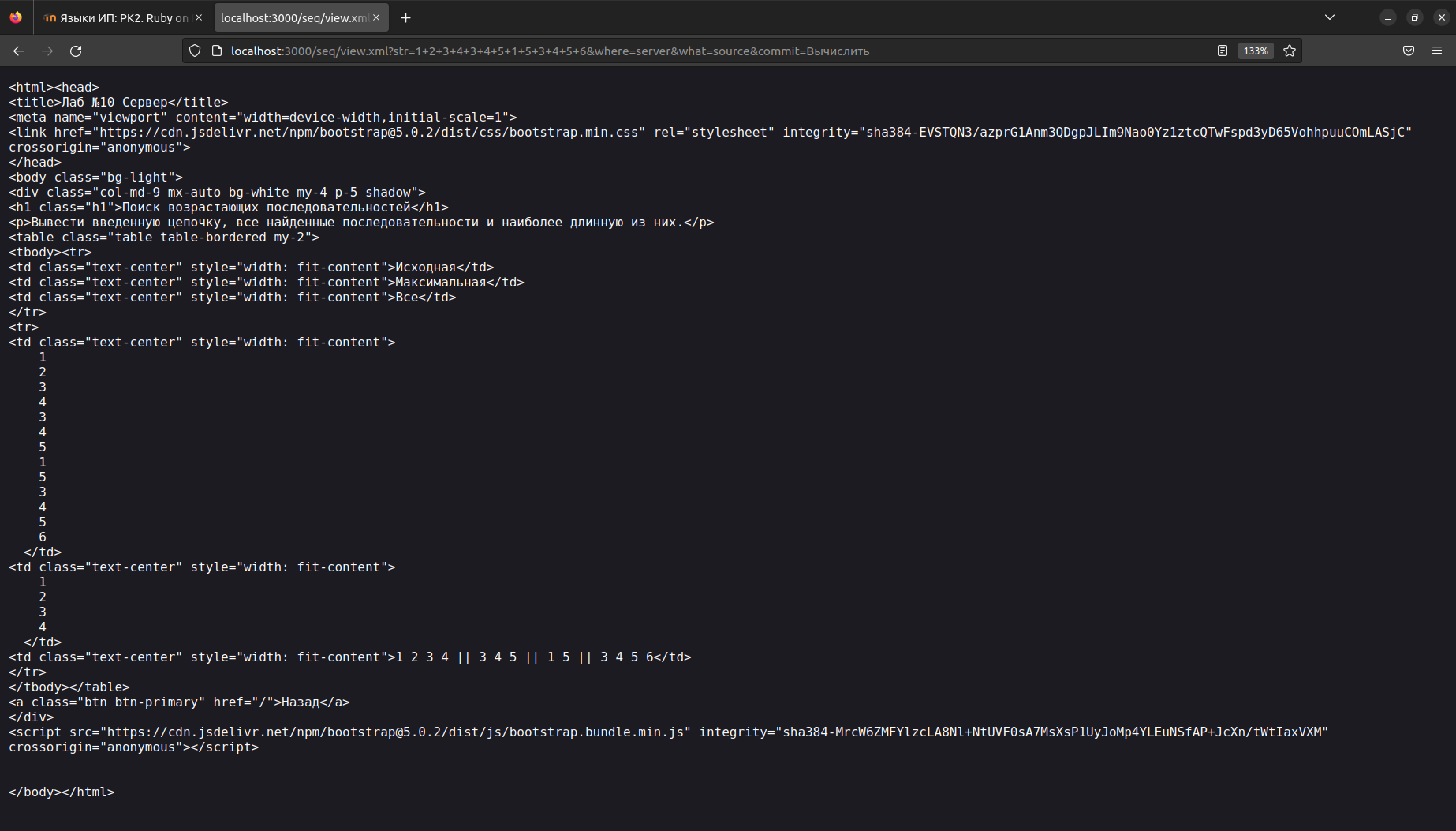
assert\_empty(Nokogiri::XML(@response.body).errors)

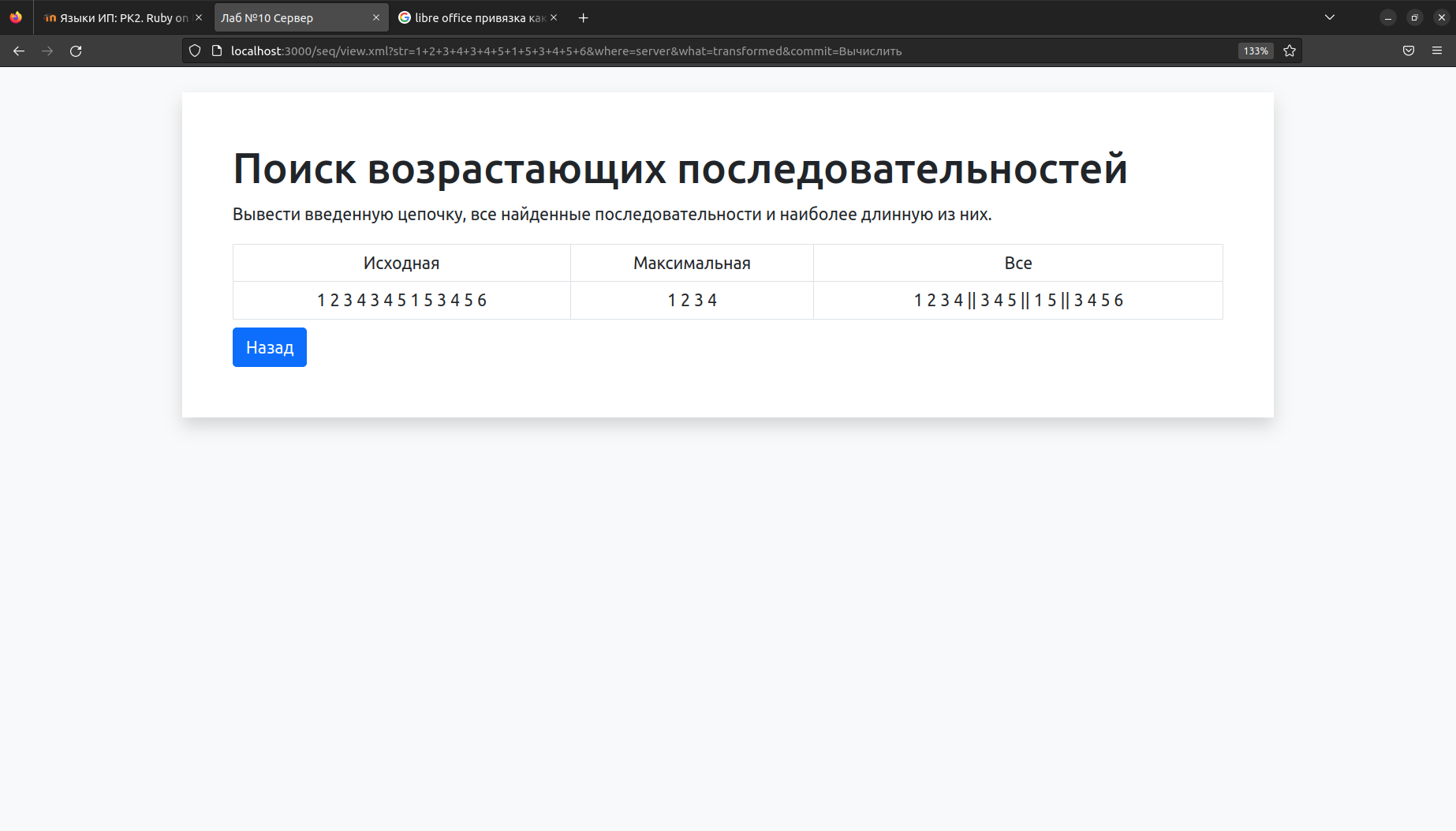
end

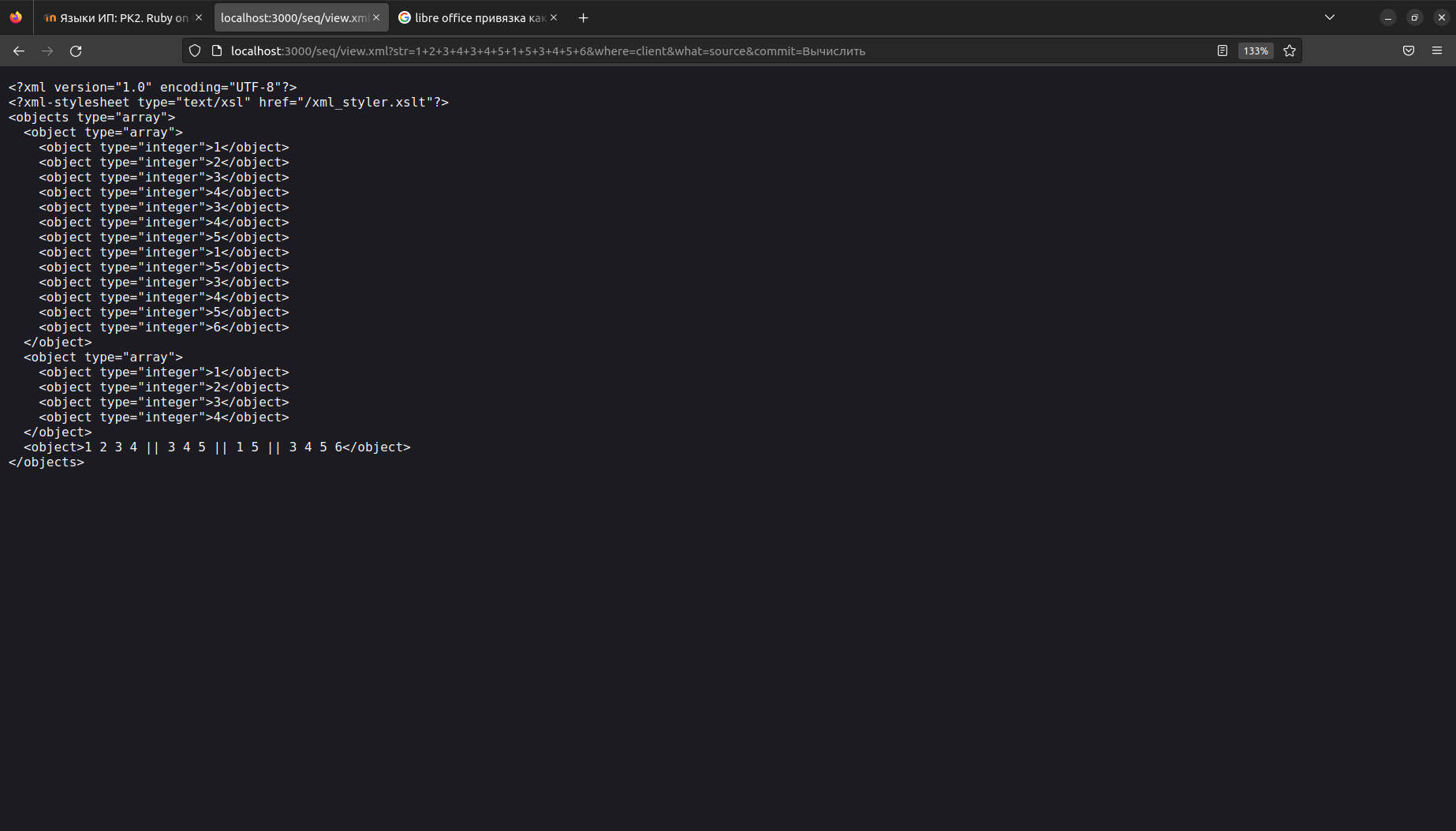
end

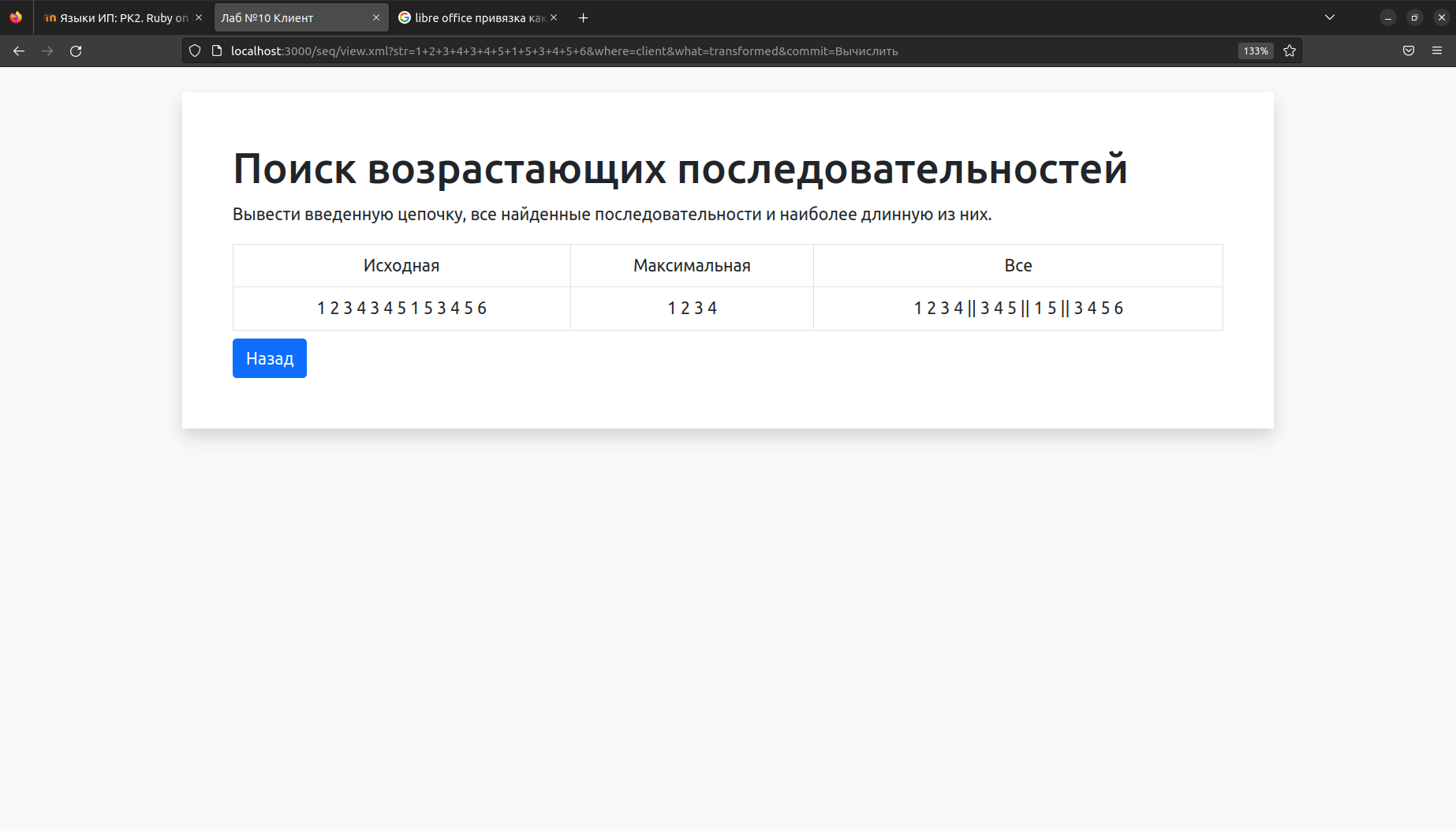
**Результаты выполнения:**



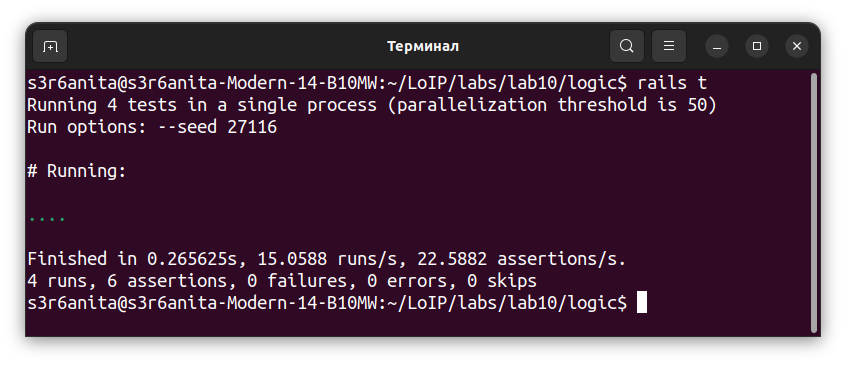


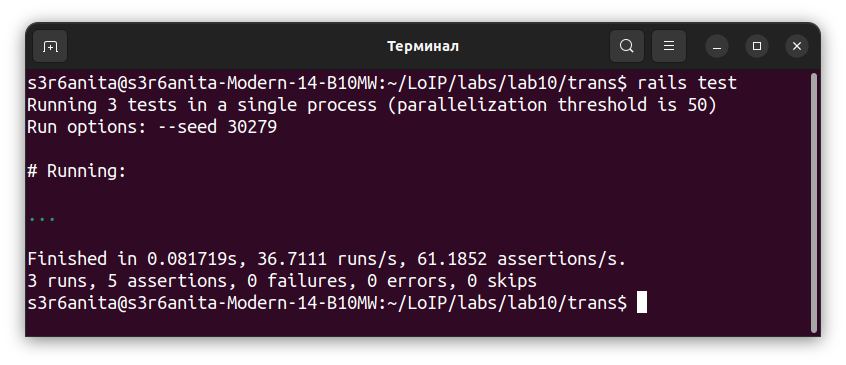




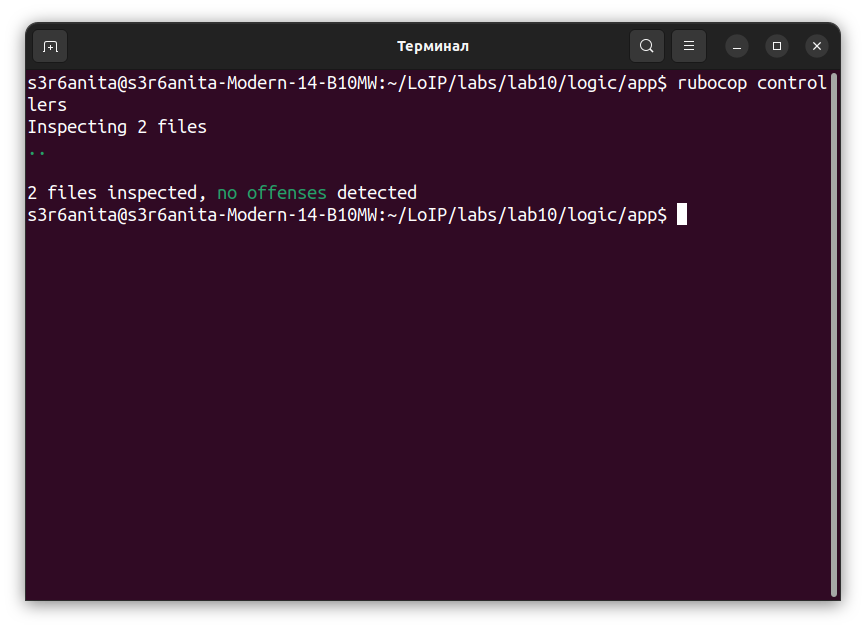


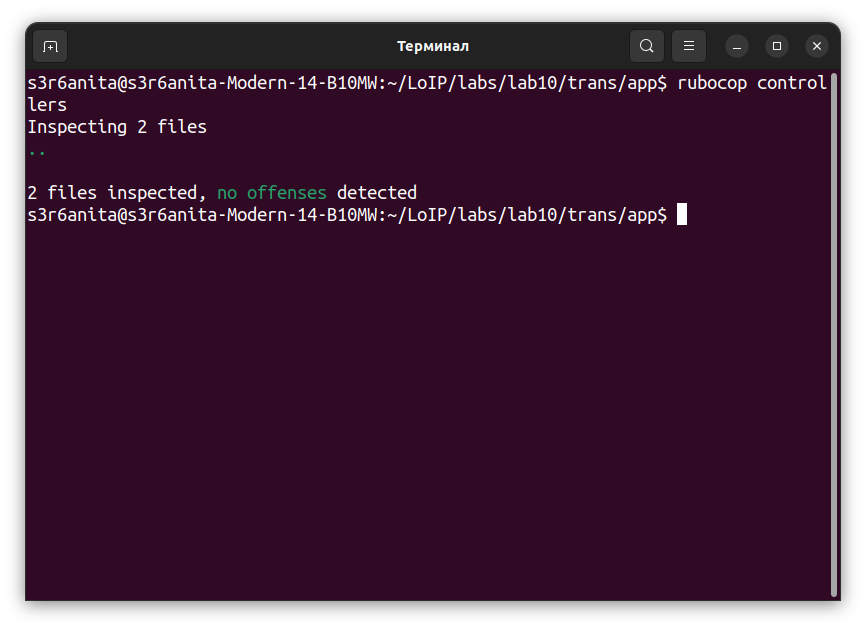
**Тестирование:**





**Результаты проверки анализатором rubocop:**





**Вывод:** в результате выполнения были получены практические навыки формирования данных в формате XML и их визуализации с  
помощью клиентских и серверных средств с использованием XSLT-преобразований.