|  |  |
| --- | --- |
| 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 №** | **9** |

**Название:**

Создание простейших веб-приложений

RUBY ON RAILS. AJAX

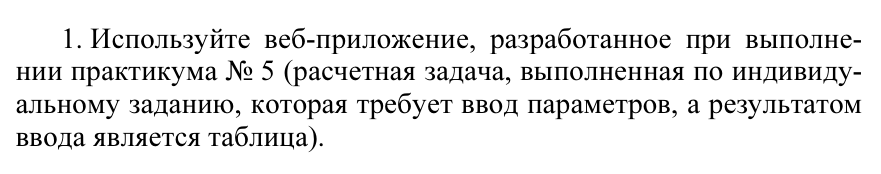
**Дисциплина:** Языки Интернет-программирования

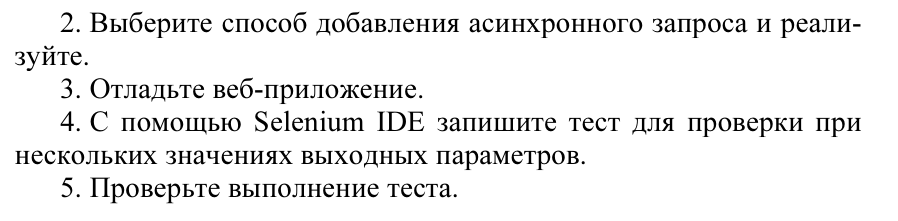


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-35Б |  | 10.11.2023 | В. И. Мамыкин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Е.Ю. Гаврилова |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Задание:**

****

****

**Цель:** углубление теоретических сведений о принципах работы асинхронного веб-интерфейса и получение практических навыков создания веб-приложения с использованием средств Ruby on Rails и технологии AJAX.

**sequences\_controller.rb**

# frozen\_string\_literal: true

class SequencesController < ApplicationController

def calculate

longest\_subsequence = [] # Самая длинная подпоследовательность

current\_subsequence = [] # Текущая подпоследовательность

all\_subsequences = [] # Все подпоследовательности

# unless params[:v2].nil?

sequence = params[:v2]&.split(' ')&.map(&:to\_i)

sequence&.each do |number|

if (Math.sqrt(number) % 1).zero?

# Если число является полным квадратом

current\_subsequence << number

else

# Если число не является полным квадратом

longest\_subsequence = current\_subsequence.clone if current\_subsequence.length > longest\_subsequence.length

cur = current\_subsequence.clone

all\_subsequences << cur.join(' ')

all\_subsequences.pop if all\_subsequences[all\_subsequences.size - 1] == ''

current\_subsequence = []

end

end

cur = current\_subsequence.clone

all\_subsequences << cur.join(' ') if cur.length.positive?

longest\_subsequence = current\_subsequence.clone if current\_subsequence.length > longest\_subsequence.length

all\_subsequences.pop if all\_subsequences[all\_subsequences.size - 1] == ''

subsequence\_count = longest\_subsequence.length

@number = [params[:v1], sequence&.size.to\_s]

@result = [all\_subsequences, longest\_subsequence.join(' '), subsequence\_count.to\_s, sequence&.join(' ')]

# end

# @number = [params[:v1], 0]

# @result = [all\_subsequences, '', '0', '']

logger.debug "from view controller: #{@result}"

respond\_to do |format|

format.html

format.json do

render json:

{ type: @result.class.to\_s, value1: @result, value2: @number }

end

end

end

end

**sequences\_controller\_test.rb**

require 'test\_helper'

class SequencesControllerTest < ActionDispatch::IntegrationTest

test 'should get calculate' do

get sequences\_calculate\_url

assert\_response :success

end

test 'should get sequence' do

get sequences\_calculate\_url, params: { v1: '10', v2: '1 2 3 4 9 16 3 2 4 10' }

assert\_equal assigns[:result], [['1', '4 9 16', '4'], '4 9 16', '3', '1 2 3 4 9 16 3 2 4 10']

end

test 'should get sequence - 2' do

get sequences\_calculate\_url, params: { v1: '3', v2: '1 4 4' }

assert\_equal assigns[:result], [['1 4 4'], '1 4 4', '3', '1 4 4']

end

end

**calculate.html.erb**

<h1>Sequences#input</h1>

<p>Find me in app/views/sequences/input.html.erb</p>

<form action="/sequences/calculate" method="get" data-remote="true" id="calc\_form" accept-charset="UTF-8" data-type="json">

<label for="v1" > Число: </label>

<input type="text" id="v1" name="v1" pattern="^([0-9])+" value="10" required/>

<label for="v2" style="margin-left: 20px"> Последовательность: </label>

<input type="text" id="v2" name="v2" pattern="^([0-9]+[\s]{0,1})+" value="1 2 3 4 9 16 3 2 4 10" required/>

<button>Найти!</button>

</form>

<div id="result"> </div>

**calc.js**

function formation\_table (data) {

let tableHTML = "<table>"; // Начало таблицы

if (data.value2[0] == data.value2[1]) {

if (data.value1[2] != '0') {

tableHTML +=

'<tr><td style="border: 2px solid darkblue;"><strong>Введенная последовательность:</strong></td></tr>' +

'<tr><td style="border: 2px solid darkblue;">' + data.value1[3] + '</td></tr>' +

'<tr><td style="border: 2px solid darkblue;"><strong>Подпоследовательности:</strong></td></tr>';

data.value1[0].forEach(function (res) {

tableHTML += '<tr><td style="border: 2px solid darkblue;">' + res + '</td></tr>';

});

tableHTML +=

'<tr><td style="border: 2px solid darkblue;"><strong>Самая длинная подпоследовательность:</strong></td></tr>' +

'<tr><td style="border: 2px solid darkblue;">' + data.value1[1] + '</td></tr>' +

'<tr><td style="border: 2px solid darkblue;"><strong>Ее длина:</strong></td></tr>' +

'<tr><td style="border: 2px solid darkblue;">' + data.value1[2] + '</td></tr>';

} else {

tableHTML +=

'<tr><td style="border: 2px solid darkblue;"><strong>Введенная последовательность:</strong></td></tr>' +

'<tr><td style="border: 2px solid darkblue;">' + data.value1[3] + '</td></tr>' +

'<tr><td style="border: 2px solid darkblue;"><strong>Подполседовательностей квадратов натуральных чисел нет</strong></td></tr>';

}

} else {

tableHTML +=

'<tr><td style="border: 2px solid darkblue;"><strong>Введенная последовательность:</strong></td></tr>' +

'<tr><td style="border: 2px solid darkblue;">' + data.value1[3] + '</td></tr>' +

'<tr><td style="border: 2px solid darkblue;"><strong>Введенное число n:</strong></td></tr>' +

'<tr><td style="border: 2px solid darkblue;">' + data.value2[0] + '</td></tr>' +

'<tr><td style="border: 2px solid darkblue;"><strong>Число n не совпадает с длиной введенной последовательности!</strong></td></tr>';

}

tableHTML += "</table>"; // Конец таблицы

return "<hr/>" + tableHTML + "<hr/><p id='date'>" + Date() + "</p>"

}

function show\_result(data) {

console.log(data);

const result = document.getElementById("result");

result.innerHTML = formation\_table(data); // Вставляем сформированную таблицу в элемент с id "result"

}

$(document).ready(function () {

$("#calc\_form").bind("ajax:success",

function (xhr, data, status) {

// data is already an object

show\_result(data);

})

});

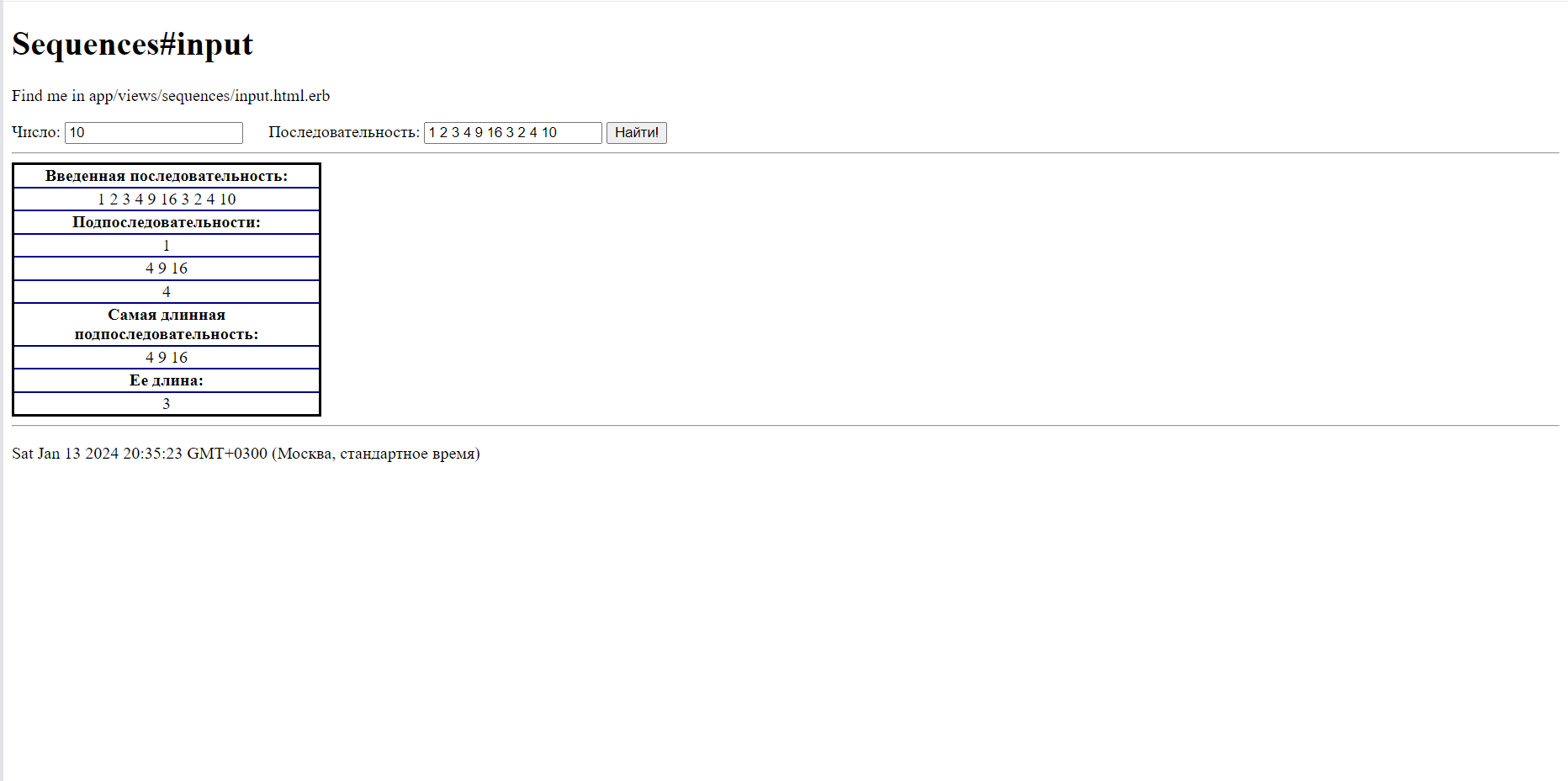


Рисунок 1 – страница ввода/вывод результата

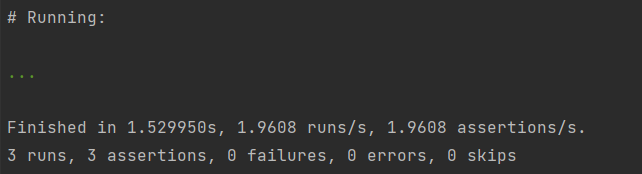


Рисунок 2 – результат функционального тестирования

**Вывод:** были изучены теоретические сведения о принципах работы асинхронного веб-интерфейса и получены практические навыки создания веб-приложения с использованием средств Ruby on Rails и технологии AJAX.