|  |  |
| --- | --- |
| **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 №** | 11 |

**Название:**

Добавление модели. ORM. Разработка БД, подключение, хранение и поиск данных.

**Дисциплина:** Языки Интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-34Б |  | 11.12.2022 | С.А. Рахманов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д. В. Малахов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы:** Освоить принципы применения средств объектно-реляционного преобразования в составе  
Ruby on Rails.

**Задание:**

Модифицировать код ЛР 8 таким образом, чтобы запросы, которые были ранее выполнены, сохранялись в БД и при следующем запросе не требовали повтора вычислений.

* Сформировать модель в соответствии с потребностями хранения данных. Входные параметры являются ключами, по которым извлекается результат.
* Выполнить создание БД и миграцию соответствующими запросами rake.
* Написать тест на добавление и поиск данных с помощью модели. Проверить выполнение теста.
* Модифицировать код приложения таким образом, чтобы результат вычислений преобразовывался в строковый или бинарный формат (на выбор: json, xml, и пр.). Проверить через отладочную печать в консоль, что преобразование выполняется корректно.
* Вставить код для сохранения данных в БД и запрос на поиск предыдущего результата вычислений.
* Добавить действие в контроллер, позволяющее определить, что хранится в БД через сериализацию в XML.
* Проверить, что при выполнении запроса, данные добавляются в БД.
* При помощи консоли сообщений Puma/Webrick определить, производится ли поиск результата предыдущего запроса в БД и не повторяются ли одни и те же вычисления.
* Модифицировать модель таким образом, чтобы добавление записей с одинаковыми параметрами было невозможно.
* Реализовать тест модели, проверяющий невозможность повторного добавления одних и тех же результатов вычислений.
* Реализовать функциональный тест, проверяющий, что результаты вычислений различны при различных входных параметрах.
* Проверить маршруты приложения с помощью rake routes и убрать лишние. Обеспечить доступ при обращении по адресу /.

**Тексты измененных файлов**

***seq\_controller.rb***

# frozen\_string\_literal: true

# Top-level class

class SeqController < ApplicationController

def input; end

def last

@str = Sequence.last

end

def base

@str = Sequence.all

end

def xml\_str

str = Sequence.all

res = str.map { |s| [JSON.parse(s.given).to\_s, JSON.parse(s.max\_seq).to\_s, JSON.parse(s.all\_seq).to\_s] }

render xml: res.to\_xml

end

def show

params[:str] ? @str = params[:str] : nil

res = Sequence.find\_by\_given(ActiveSupport::JSON.encode(@str))

if res

@source = 'DataBase'

@max\_found = ActiveSupport::JSON.decode(res.max\_seq)

@found = ActiveSupport::JSON.decode(res.all\_seq)

else

@source = 'Calculation'

arr = @str&.split&.map(&:to\_i)

if arr && !arr&.empty?

tmp = helpers.find\_seqs(arr)

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

@max\_found = helpers.find\_max(tmp)

else

@max\_found = @str

@found = @str

end

res = Sequence.create given: ActiveSupport::JSON.encode(@str), max\_seq: ActiveSupport::JSON.encode(@max\_found),

all\_seq: ActiveSupport::JSON.encode(@found)

res.save

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

p found

found

end

def find\_max(alr\_found)

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

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

end

end

***sequence.rb***

# frozen\_string\_literal: true

class Sequence < ApplicationRecord

validates\_uniqueness\_of :given

end

***application.html.erb***

<!DOCTYPE html>

<html>

<head>

<title>Lab11</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-10 mx-auto my-4 p-5 shadow">

<%= yield %>

</div>

</body>

</html>

***input.html.erb***

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

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

<%= form\_tag("/seq/show", :method => "get") do %>

<%= label\_tag("Введите последовательность:") %>

<%= text\_field\_tag(:str) %> <br/>

<br/>

<%= submit\_tag("Найти", class: 'btn btn-primary') %>

<% end %>

</br>

<%= form\_tag("seq/base", :method => "get") do%>

<%= submit\_tag("Посмотреть базу данных", class: 'btn btn-primary')%>

<% end%>

</br>

<%= form\_tag("seq/xml\_str", :method => "get", data:{turbo: false}) do%>

<%= submit\_tag("xml базы данных", class: 'btn btn-primary')%>

<% end%>

</br>

<%= form\_tag("seq/last", :method => "get") do%>

<%= submit\_tag("Последнее добавление", class: 'btn btn-primary')%>

<% end%>

***show.html.erb***

<h1 class="h1">Результат</h1>

<p>Найдены последовательности</p>

<p id="str">Введенная строка: <%= @str %></p>

<p id="source">Источник: <%= @source %></p>

<p id="max found: ">Максимальная последовательность: <%= @max\_found %></p>

<p id="found: ">Все последовательности: <%= @found %></p>

<%= link\_to "Repeat input", '/' %>

***last.html.erb***

<h1 class="h1">Последнее вычисление:</h1>

<h1 class="h1">Последнее вычисление:</h1>

<div>

<p>Промежуток <%= @str.given %></p>

<p>Макс. последовательность: <%= @str.max\_seq %> </p>

<p>Все последовательности: <%= @str.all\_seq %> </p>

<p>Создано: <%= @str.created\_at %></p>

<%= link\_to "Назад", "/" %>

</div>

***base.html.erb***

<h1>Содержимое базы данных</h3>

<table >

<tr>

<th> Данная строка </th>

<th> Макс. последовательность </th>

<th> Все последовательности </th>

<th> Дата создания </th>

<th> Время создания </th>

</tr>

<% @str.each do |i| %>

<tr>

<td> <%= i.given %> </td>

<td> <%= i.max\_seq %> </td>

<td> <%= i.all\_seq %> </td>

<td> <%= i.created\_at %> </td>

<td> <%= i.updated\_at %> </td>

</tr>

<tr></tr>

<% end %>

</table>

<%= link\_to "Назад", "/" %>

***routes.rb***

# frozen\_string\_literal: true

Rails.application.routes.draw do

root 'seq#input', as: :home

get 'seq/show'

get 'seq/base'

get 'seq/last'

get 'seq/xml\_str'

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

# Defines the root path route ("/")

# root "articles#index"

end

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

# frozen\_string\_literal: true

require 'test\_helper'

TEST\_DATA = [

{

input: { str: '1 2 3 4 3 4 5 1 1 1 5 3 4 5 6 10 0 -1 -3' },

output: ['1 2 3 4 3 4 5 1 1 1 5 3 4 5 6 10 0 -1 -3', '3 4 5 6 10',

'1 2 3 4 | 3 4 5 | 1 | 1 | 1 5 | 3 4 5 6 10 | 0 | -1 | -3']

},

{

input: { str: '1 2 3 4 1 2 3 1 2' },

output: ['1 2 3 4 1 2 3 1 2', '1 2 3 4', '1 2 3 4 | 1 2 3 | 1 2']

},

{

input: { str: '38 23 20 28 49 22 39 9 3 56 4' },

output: ['38 23 20 28 49 22 39 9 3 56 4', '20 28 49', '38 | 23 | 20 28 49 | 22 39 | 9 | 3 56 | 4']

}

].freeze

# Top-level class

class SeqControllerTest < ActionDispatch::IntegrationTest

test 'should get show' do

get '/seq/show'

assert\_response :success

end

test 'should get input' do

get '/'

assert\_response :success

end

test 'should return empty result' do

input = { str: '' }

get "/seq/show?#{input.to\_query}"

assert\_equal('', assigns(:str))

assert\_equal('', assigns(:max\_found))

assert\_equal('', assigns(:found))

end

test 'should return correct data' do

TEST\_DATA.each do |data|

input = data[:input]

output = data[:output]

get "/seq/show?#{input.to\_query}"

assert\_equal(output[0], assigns(:str))

assert\_equal(output[1], assigns(:max\_found))

assert\_equal(output[2], assigns(:found))

end

end

test 'different data' do

input1 = TEST\_DATA[0][:input]

input2 = TEST\_DATA[1][:input]

get "/seq/show?#{input1.to\_query}"

a = [assigns(:str), assigns(:max\_found), assigns(:found)]

# aa = assigns(:max\_found)

get "/seq/show?#{input2.to\_query}"

b = [assigns(:str), assigns(:max\_found), assigns(:found)]

# bb = assigns(:max\_found)

# assert\_not\_equal(aa, bb)

a.each\_index do |i|

assert\_not\_equal(a[i], b[i])

end

end

# test "should get 11 for view with with 1+10" do

# get '/seq/show', params: {v1: 1, v2: 10, op: '+'}

# assert\_equal(assigns[:result], 11)

# end

# test "should get Unknown! for incorrect params" do

# get '/seq/show'

# assert\_equal(assigns[:result], 'Unknown!')

# end

end

***sequence\_test.rb***

# frozen\_string\_literal: true

require 'test\_helper'

# top-level documentation

class SequenceTest < ActiveSupport::TestCase

test 'cannot save twice' do

res = Sequence.create given: '1 2 3', max\_seq: '1 2 3', all\_seq: '1 2 3'

res.save

res = Sequence.create given: '1 2 3', max\_seq: '0 1 2 3', all\_seq: '1 2 3'

res.save

res = Sequence.find\_by\_given('1 2 3')

assert\_equal(res.max\_seq, '1 2 3')

end

end

***20221212175731\_create\_sequences.rb***

# frozen\_string\_literal: true

# top-level documentation

class CreateSequences < ActiveRecord::Migration[7.0]

def change

create\_table :sequences do |t|

t.string :given

t.string :max\_seq

t.string :all\_seq

t.timestamps

end

end

end

***schema.rb***

ActiveRecord::Schema[7.0].define(version: 20\_221\_212\_175\_731) do

create\_table 'sequences', force: :cascade do |t|

t.string 'given'

t.string 'max\_seq'

t.string 'all\_seq'

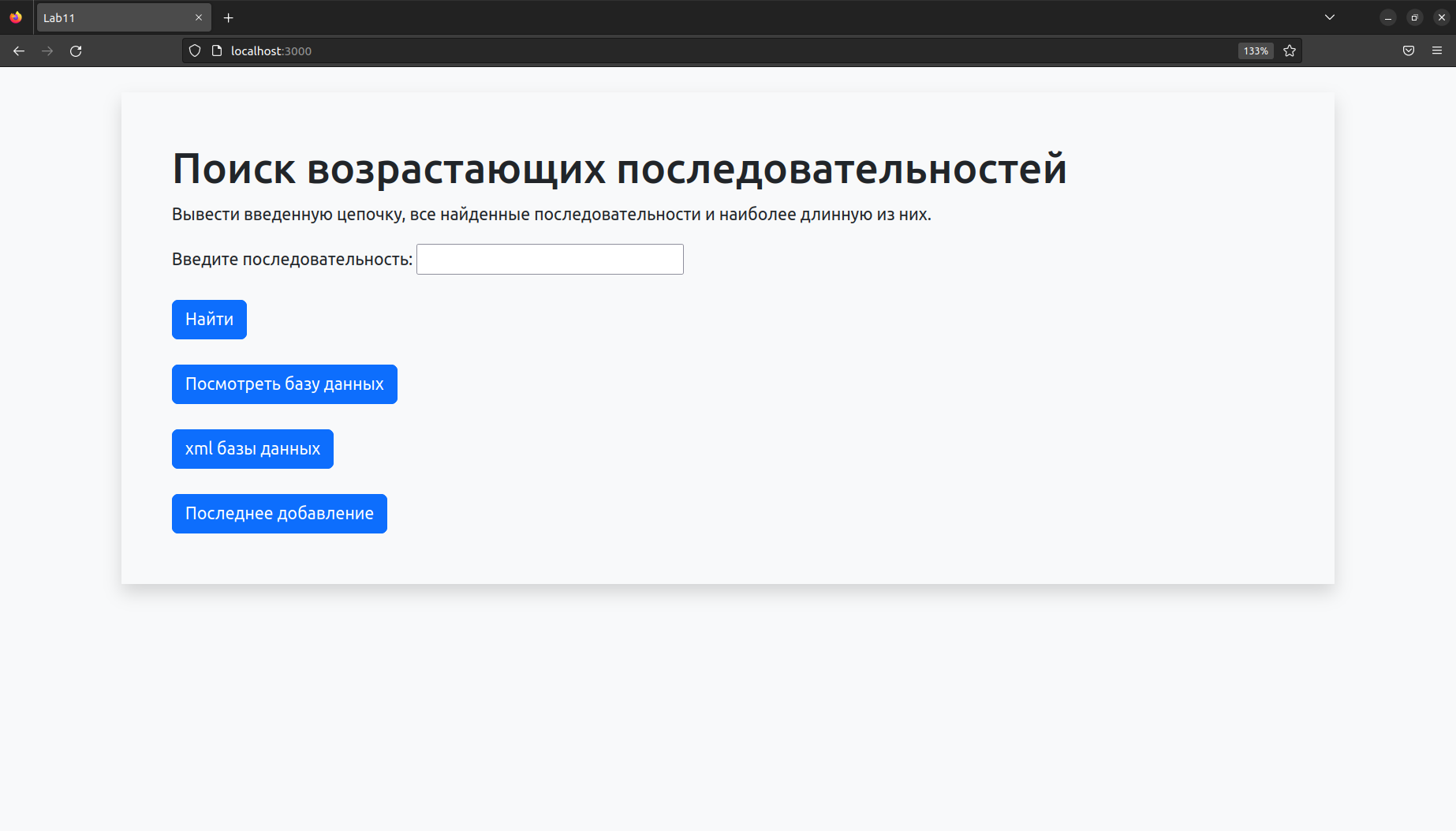
t.datetime 'created\_at', null: false

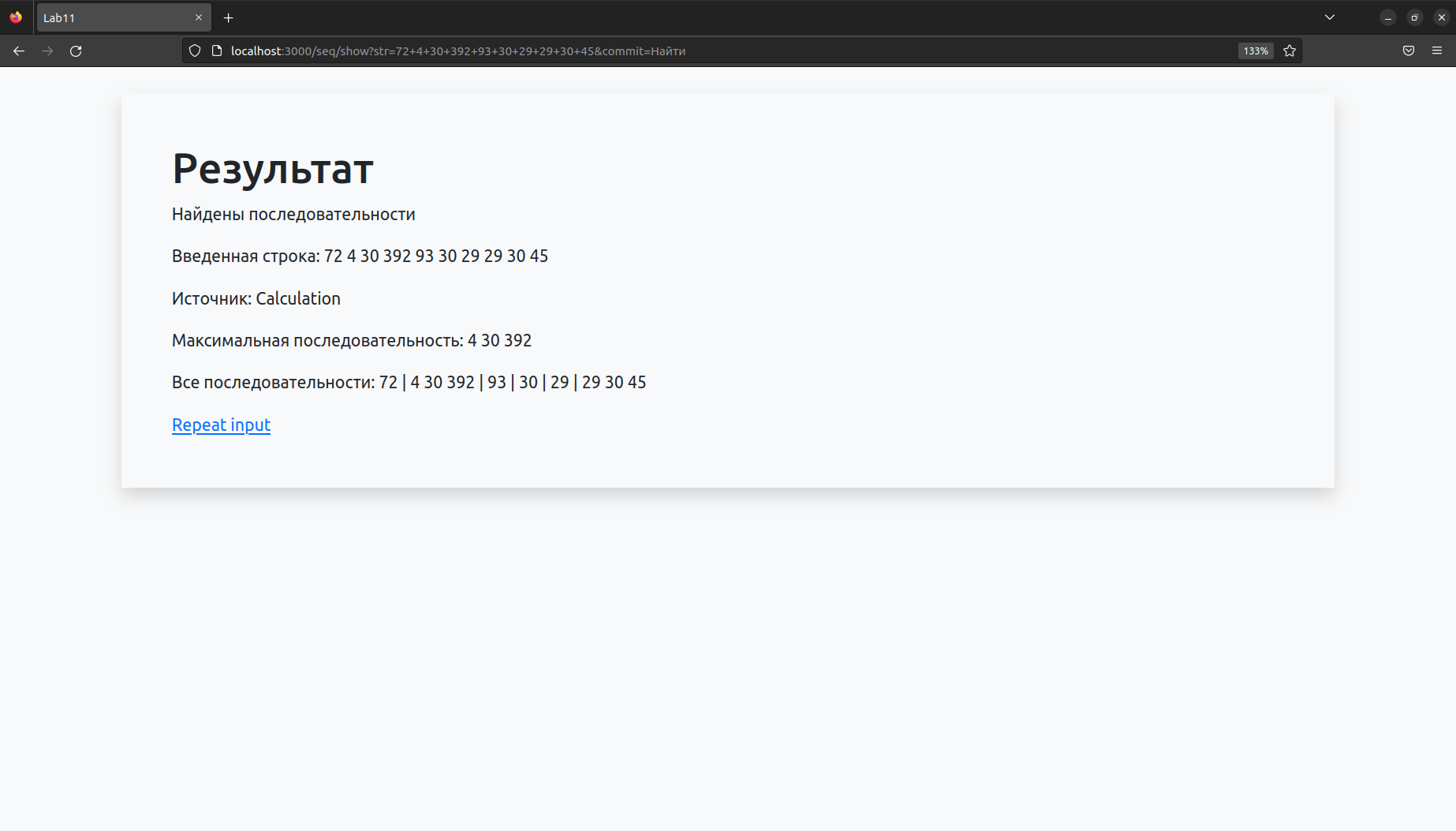
t.datetime 'updated\_at', null: false

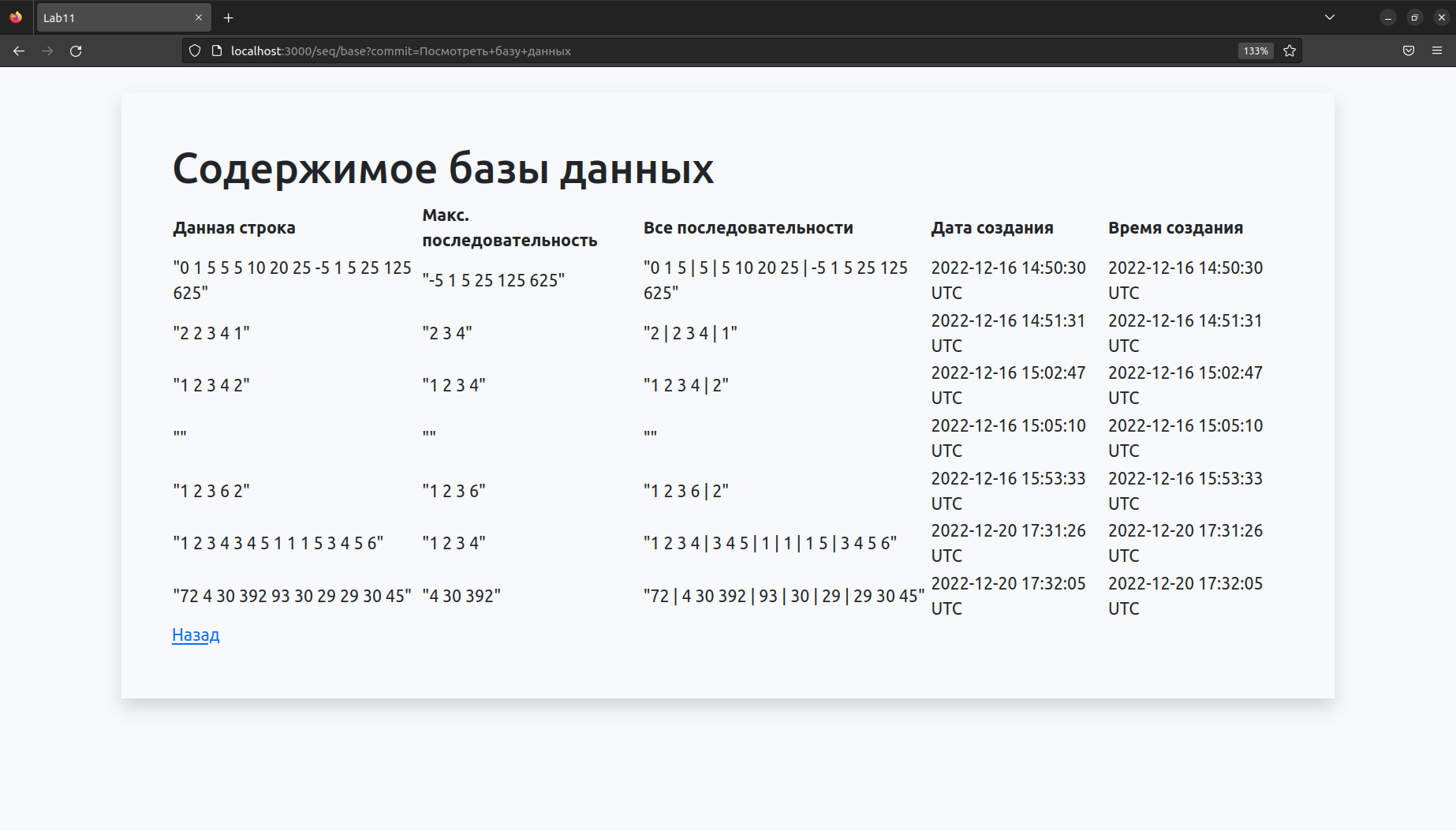
end

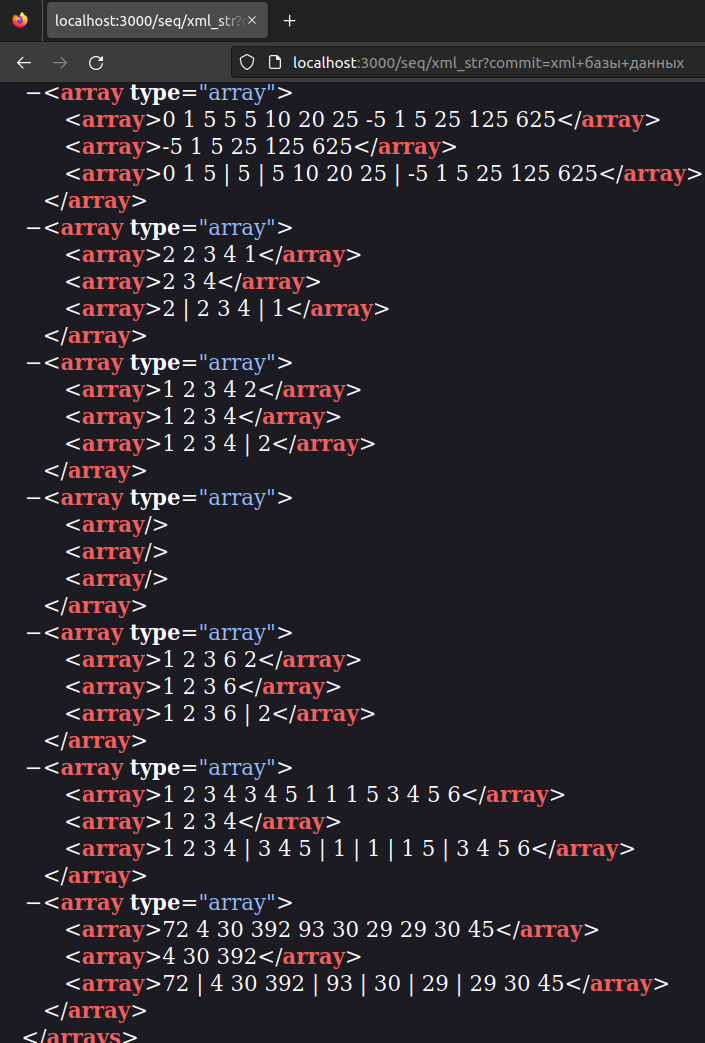
end

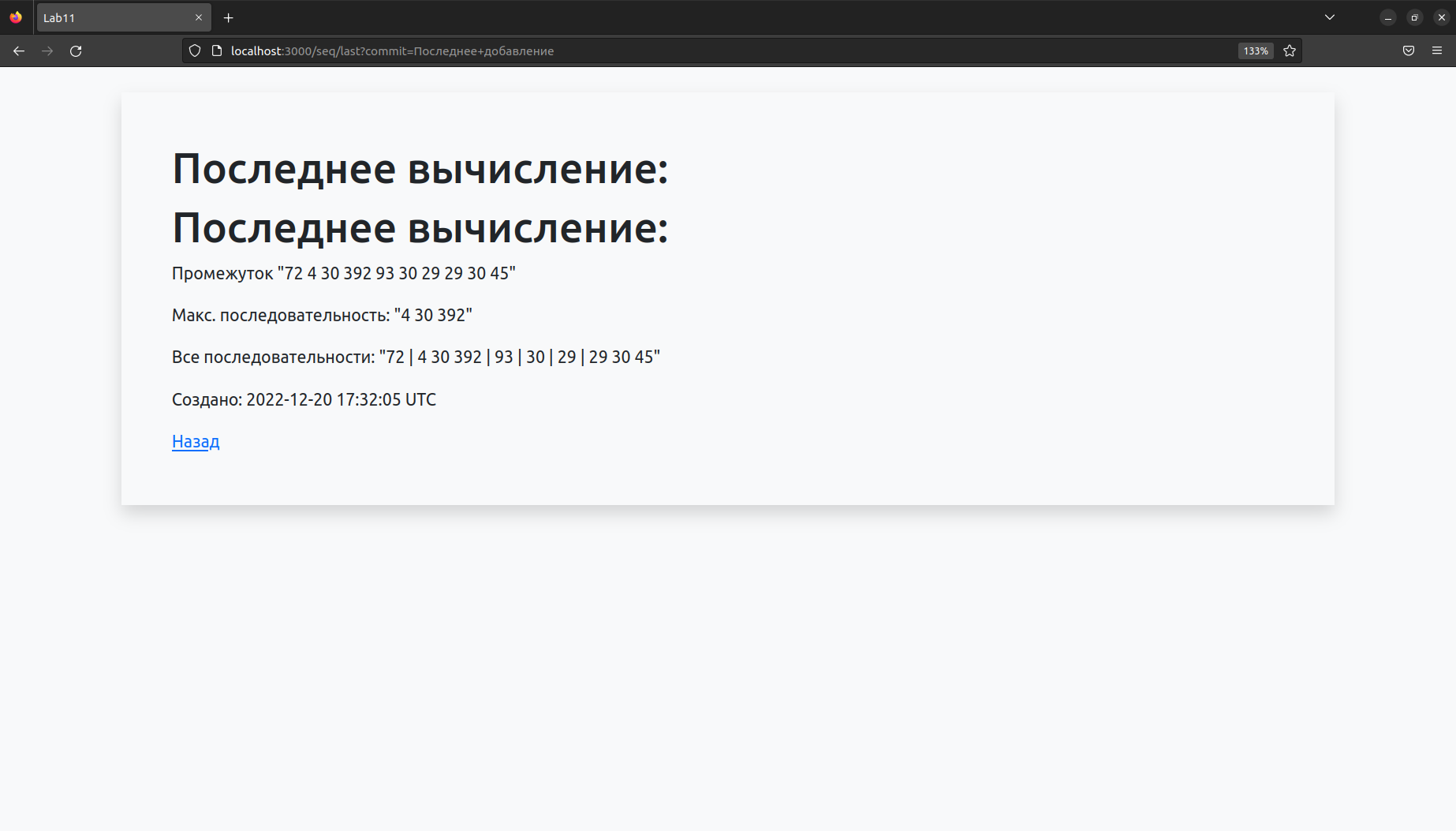
**Результаты выполнения:**



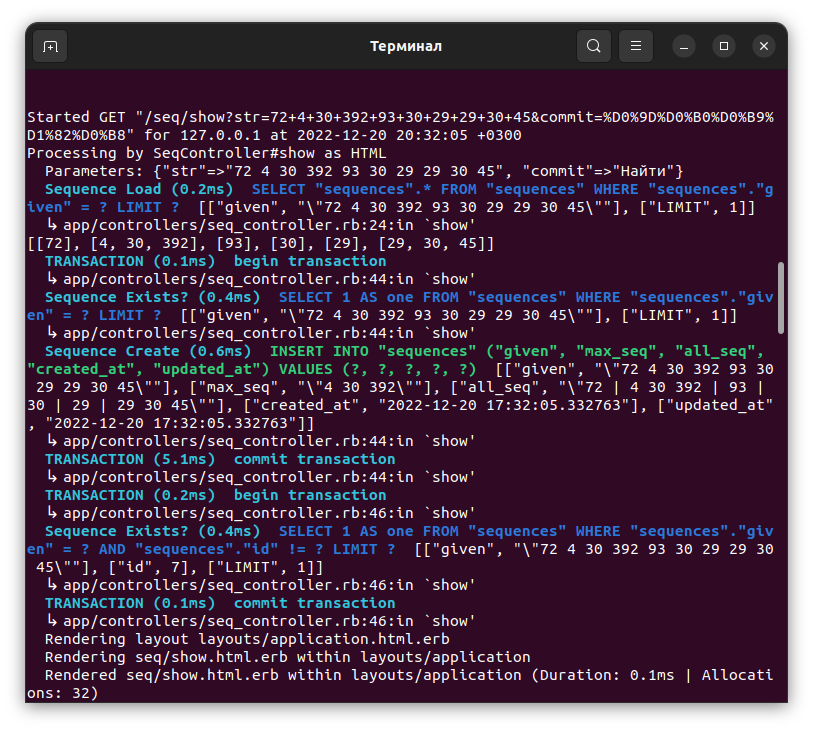




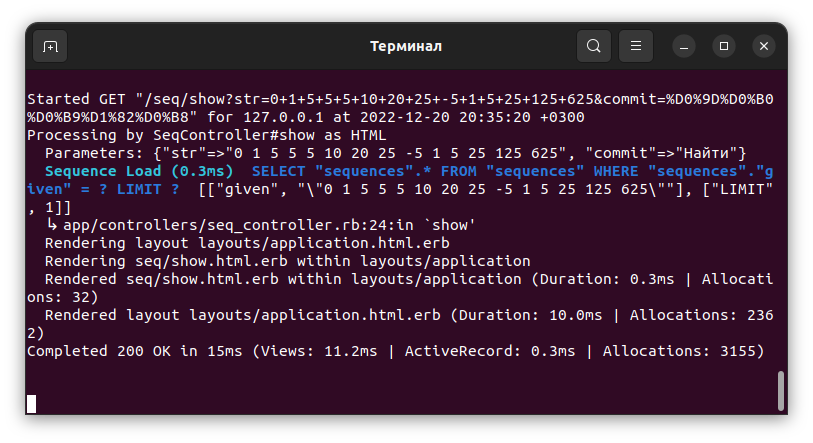


****

**Консоль при добавлении результатов:**

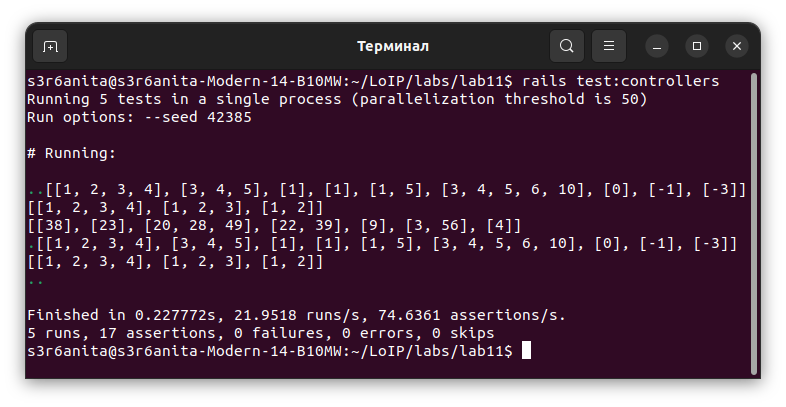
****

**Консоль при получении результата из базы данных:**

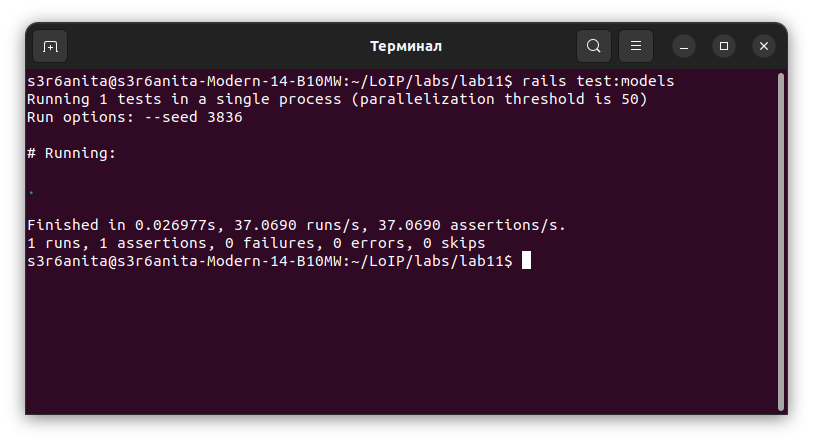


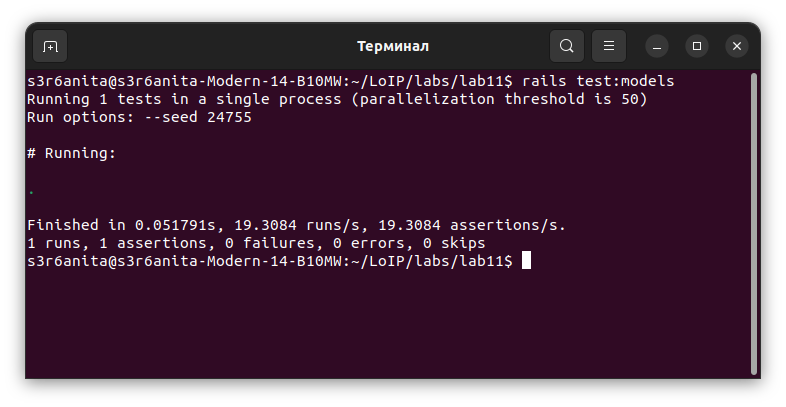
**Тестирование:**

для контроллера:

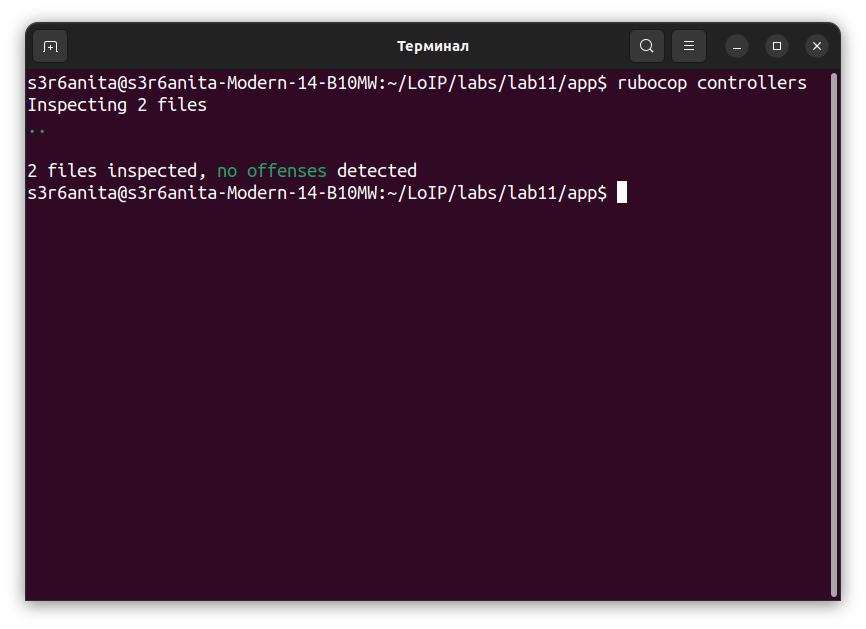


для модели





**Результаты проверки анализатором rubocop:**

****

**Вывод:** в результате выполнения были освоены принципы применения средств объектно-реляционного преобразования в составе  
Ruby on Rails.