



**«Московский государственный технический университет  
имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

## **Отчет**

### **Лабораторная работа № 8**

**Дисциплина: Языки интернет программирования**

**Название лабораторной работы: Создание каркаса RoR-приложения.  
Настройка базовых параметров. Размещение файлов.  
Создание форм. Функциональные тесты.**

Студент гр.ИУ6-32

\_\_\_\_\_  
(Подпись, дата)

**Дубровин Е.Н.**  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

**Самарев Р.С.**  
(И.О. Фамилия)

## Часть 1

### Задание:

Метод Ньютона – Рафсона основывается на утверждении, что квадратный корень числа  $A$  можно найти с помощью рекуррентной формулы:

$$x_{i+1} = \frac{1}{2} \left( x_i + \frac{A}{x_i} \right), \text{ где } A — \text{положительное число, } x_i — \text{текущее приближение}$$

квадратного корня,  $x_{i+1}$  — очередное приближенное значение квадратного корня из числа  $A$ . Написать программу, определяющую квадратный корень заданных чисел, используя приведенную формулу.

1. Для решения задачи был создан контроллер `Sqrt` с представлениями `input` и `view`.

### 2. Контроллер `Sqrt`.

```
class SqrtController < ApplicationController
  def input
    end

  def view
    @a_value = (buf = params[:a].to_f).positive? ? buf : 0
    sqrt = sqrt_aprox_enum(@a_value)

    @result_array = sqrt.each_cons(2).take_while do |aproxes|
      (aproxes[0] - aproxes[-1]).abs > 1e-3
    end.flatten.uniq
    # original (aprox + 1)**(2 - @a_value) / @a_value > 1e-3
  end

  private def sqrt_aprox_enum(a)
    Enumerator.new do |yielder|
      aprox = 1e-3
      loop do
        yielder.yield aprox
        aprox = 0.5 * (aprox + a / aprox)
      end
    end
  end
end
```

### 3. Представление `input`.

```
<div class='headline'><%= t('calc_sqrt') %></div>
<div>
  <%= form_tag('/sqrt_aprox', :method => 'get') do %>
    <%= label_tag(t('enter_number')) %>
    <%= text_field_tag(:a) %>
    <br/><br/>
    <%= submit_tag(t('calculate')) %>
  <% end %>
</div>
```

## 4. Представление view.

```
<div class='headline'><%= t('sqrt_of') %><%= @a_value.to_s.gsub(/\.0(?:\d+)/,
'') %></div>
<div class='table'>
  <div class='row header'>
    <div class='cell'><%= t('iteration') %></div>
    <div class='cell'><%= t('aproximation') %></div>
  </div>
  <% @result_array.each_with_index do |x, i| %>
    <div class='row'>
      <div class='cell'><%= i+1 %></div>
      <div class='cell'><%= x %></div>
    </div>
  <% end %>
</div>
<div><%= link_to t('repeat'), '/' %></div>
```

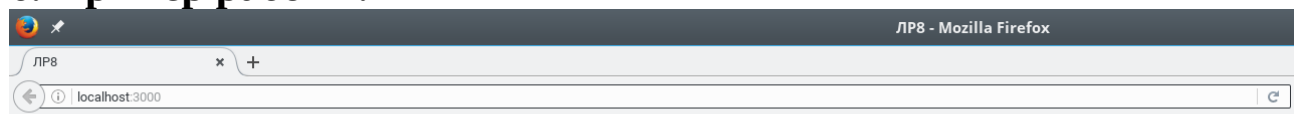
## 5. Файл конфигурации маршрутов.

```
Rails.application.routes.draw do
  root 'sqrt#input'
```

```
  get '/sqrt_aprox' => 'sqrt#view'
```

```
  # For details on the DSL available within this file, see
  http://guides.rubyonrails.org/routing.html
end
```

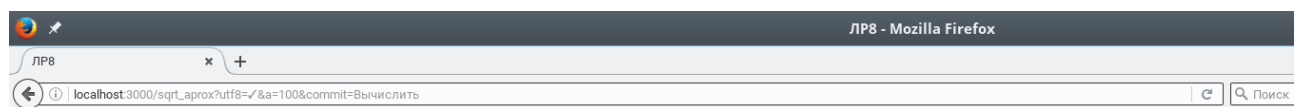
## 6. Пример работы.



### Вычисление квадратного корня

Введите число:

Вычислить



### Квадратный корень 100

Итерация	Приближение
1	0.001
2	50000.0005
3	25000.00124999999
4	12500.002624999895
5	6250.005312499108
6	3125.010656242754
7	1562.5213280668172
8	781.2926635966157
9	390.7103283034969
10	195.4831361974763
11	97.99734463768978
12	49.508890225104054
13	25.76436474057566
14	14.822847335384223
15	10.784594750729779
16	10.028540197249
17	10.000040611237678

Повторить вычисление

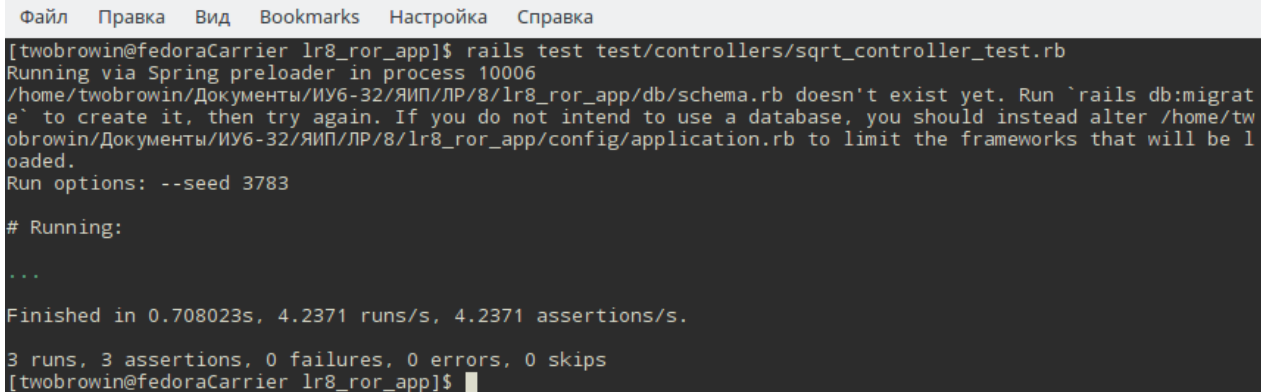
## 7. Класс тестирования контроллера Sqrt.

```
class SqrtControllerTest < ActionDispatch::IntegrationTest
  test 'should get input' do
    get '/'
    assert_response :success
  end

  test 'should get view' do
    get '/sqrt_aprox'
    assert_response :success
  end

  test 'sqrt_aprox' do
    get "/sqrt_aprox?a=#{rand(1e+6)/1e+3}"
    answer = controller.instance_variable_get(:@a_value)**0.5
    result = controller.instance_variable_get(:@result_array)[-1]
    assert_in_delta answer, result
  end
end
```

## 8. Вывод тестирования:



```
Файл  Правка  Вид  Bookmarks  Настройка  Справка
[twobrowin@fedoraCarrier lr8_ror_app]$ rails test test/controllers/sqrt_controller_test.rb
Running via Spring preloader in process 10006
/home/twobrowin/Документы/ИУ6-32/ЯИП/ЛР/8/lr8_ror_app/db/schema.rb doesn't exist yet. Run `rails db:migrate`
to create it, then try again. If you do not intend to use a database, you should instead alter /home/twobrowin/Документы/ИУ6-32/ЯИП/ЛР/8/lr8_ror_app/config/application.rb to limit the frameworks that will be loaded.
Run options: --seed 3783

# Running:

...

Finished in 0.708023s, 4.2371 runs/s, 4.2371 assertions/s.

3 runs, 3 assertions, 0 failures, 0 errors, 0 skips
[twobrowin@fedoraCarrier lr8_ror_app]$
```

## Выводы

- Научился создавать каркас RoR приложения, создавать контроллеры и представления, проводить тестирование контроллеров.
- Работа приложения осуществляется без использования базы данных.