



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

О Т Ч Е Т

по лабораторной работе № 8


Название: Ruby on Rails

Дисциплина: Языки интернет-программирования

Студент

ИУ6-35 Б

(Группа)


24.11.2023

(Подпись, дата)

Дулина И.А.

(И.О. Фамилия)

Преподаватель

Е.Ю. Гаврилова

(Подпись, дата)

(И.О. Фамилия)

Вариант 8

Цель работы: углубление теоретических сведений о принципах проектирования Model-View-Controller и получение практических навыков создания веб-приложения с использованием средств Ruby on Rails, построения простейших форм и выполнения вычислений на стороне серверной части приложения.

Задание:

Разработать веб-приложение, имеющее HTML-страницу с формой ввода данных и HTML-страницу для представления результатов. Результат расчёта должен быть представлен в форме таблицы, оформленной с помощью элемента table или отдельными ячейками div и имеющей не менее двух колонок. Если по условию задания результат может быть представлен только в виде одной строки таблицы, необходимо реализовать вывод промежуточных результатов расчёта в качестве дополнительных строк. В этом случае первой колонкой таблицы будет порядковый номер итерации.

Под вводом с клавиатуры в тексте заданий следует понимать ввод в поле ввода данных формы на HTML-странице.

Написать программу, которая вводит цепочку целых чисел (количество чисел не менее 10) и определяет наиболее длинную монотонно возрастающую их последовательность. Вывести на печать введенную цепочку, все найденные последовательности и наиболее длинную из них. При программировании использовать функцию.

Создаём приложение

```
$ rails new chisla
```

Создаём контроллер

```
$ cd chisla
```

```
$ rails generate controller Chisla input view
```

В gem file добавляем строку

```
gem 'rails-controller-testing'
```

Код файла app/views/chisla/input.html.erb

```
<h1>Chisla#input</h1>
<p>Find me in app/views/chisla/input.html.erb</p>
<%=form_tag("/chisla/view", :method => "get") do %>
  <%=label_tag("Введите не менее 10 чисел")%>
  <%=text_field_tag(:str)%><br/><br/>
  <%=submit_tag("Найти наиболее длинную монотонно возрастающую
последовательность")%>
<% end%>
```

Код файла app/views/chisla/view.html.erb

```
<h1>Chisla#view</h1>
<p>Find me in app/views/chisla/view.html.erb</p>
<p>Результат: <%=@final%></p>
<p><%= @err%></p>
<p>Таблица результатов</p>
<table border="1">
<tr>
  <th>Изначальный ряд</th>
  <th>все возможные последовательности</th>
  <th>Самая длинная</th>
</tr>
<% @solution.each do |row|%>
  <tr>
    <%row.each do |value|%>
      <td>
        <%=value%>
```

```

        </td>
    <%end%>
</tr>
<%end%>
<table>
<%= link_to "Найти для других чисел", :chisla_input %>

```

Код файла app/controllers/chisla_controller.rb

```

class ChislaController < ApplicationController
  def input
    end

  def view
    begin
      res = params[:str].scan(/-?\d+(?:\.\d+)?/).map(&:to_i)
      raise StandardError if res.length < 10
      max = 0
      all = []
      solution = ""
      i = 0
      loop do
        posl, len, i = create_posl(i, res)
        all << posl.join(' ')
        if len > max
          max = len
          solution = posl.join(' ')
        end
        break if i >= res.length
      end
    end
  end
end

```

```

result = []
all.length.times do |j|
  if solution == all[j]
    str = '+'
  else
    str = ''
  end
  if j == 0
    result << [res.join(' '), all[j], str]
  else
    result << [' ', all[j], str]
  end
end
@final = solution
@solution = result
rescue StandardError
  @err = 'Что-то пошло не так'
  @solution = []
  # Ignored
end
end

```

```

def create_posl(i, res)
  len = 0
  posl = []
  loop do
    len += 1
    posl << res[i]
    break if i + 1 == res.length
    break if (res[i+1] <= res[i])
  end
end

```

```

        i += 1
      end
      i+=1
      [posl, len, i]
    end
  end
end

```

Код файла test/controllers/ chisla_controller_test.rb

```

require "test_helper"

class ChislaControllerTest < ActionDispatch::IntegrationTest
  test "should get input" do
    get chisla_input_url
    assert_response :success
  end

  test "should get view" do
    get chisla_view_url
    assert_response :success
  end

  test "for error" do
    get chisla_view_url, params: { str: 'fgdf' }
    assert_equal assigns[:err], 'Что-то пошло не так'
  end

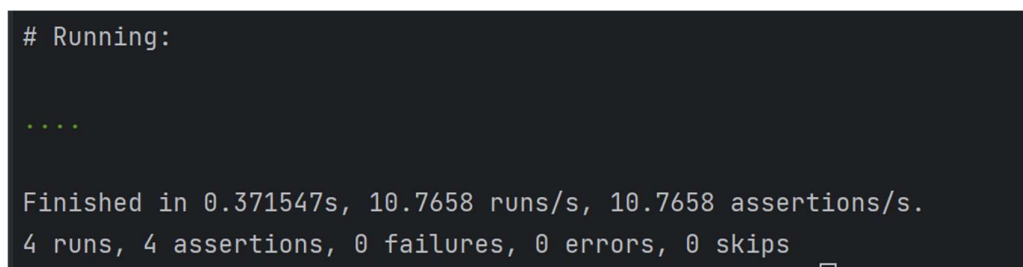
  test "for right" do
    get chisla_view_url, params: { str:'5 6 7 1 2 0 -1 2 5 6' }
    assert_equal assigns[:final], '-1 2 5 6'
  end
end

```

Код файла config/routes.rb (замена обращения по корневому адресу на обращение к действиям созданного контроллера)

```
Rails.application.routes.draw do
  get 'chisla/input'
  get 'chisla/view'
  # Define your application routes per the DSL in
https://guides.rubyonrails.org/routing.html
```

```
  # Defines the root path route ("/")
  # root "articles#index"
  root 'chisla#input'
end
```



```
# Running:
...
Finished in 0.371547s, 10.7658 runs/s, 10.7658 assertions/s.
4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
```

Рисунок 1 – результат автоматического тестирования

Chisla#input

Find me in app/views/chisla/input.html.erb

Введите не менее 10 чисел

Рисунок 2 – пример ввода данных на странице

Chisla#view

Find me in app/views/chisla/view.html.erb

Результат: -5 -4 -3 -2 -1 0 1 2 3

Таблица результатов

Изначальный ряд	все возможные последовательности	Самая длинная
-5 -4 -3 -2 -1 0 1 2 3 2 1 5 6 6 7 8	-5 -4 -3 -2 -1 0 1 2 3	+
	2	
	1 5 6	
	6 7 8	

[Найти для других чисел](#)

Рисунок 3 – пример выводов результатов вычислений

Вывод: для решения поставленной задачи было создано web-приложение на основе фреймворка Ruby on Rails, проведено автоматическое тестирование (см. Рис. 1), которое показало корректность работы приложения и отображения веб-страниц в браузерах (см. Рис. 2 и 3).