



**«Московский государственный технический университет  
имени Н.Э. Баумана»**

**(национальный исследовательский университет)**

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

## **О т ч е т**

**по лабораторной работе № 11**

**Дисциплина: Языки Интернет-программирования.**

Студент гр. ИУ6-33Б

\_\_\_\_\_  
(Подпись, дата)

**Дасов Т.Д.**

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия)

Москва, 2019

## ***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 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 /.

## ***Маршруты routes.rb:***

```
Rails.application.routes.draw do
  root 'main#index', :as => 'root'

  get 'admin/handling', :as => 'handle'
  get 'admin/res', :as => 'res'

  get 'main/result' => 'main#result', :as => 'result'
  get 'main/index' => 'main#index', :as => 'index'
end
```

## ***Контроллер (main\_controller.rb):***

```
# frozen_string_literal: true

# MainController class
class MainController < ApplicationController
  def index; end

  def result
    @res = FriendNumbers.call(input: params[:n].to_i).res
    respond_to do |format|
      format.js
      format.html
      format.json { render json: @res.to_json }
      format.xml { render xml: @res.to_xml }
    end
  end
end
```

### ***Контроллер для доступа к БД приложения (admin\_controller.rb):***

```
# frozen_string_literal: true

# AdminController class
class AdminController < ApplicationController
  def auth; end

  def handling
    @rec = Calculation.all
  end

  def res
    @rec = ConvertObject.call(obj: Calculation.all).res.to_xml
  end
end
```

### ***Представление (handling.html.erb):***

```
<div class="container admin-body">
  <div class="row admin-page-row">
    <div class="col">
      <h1>Администрирование</h1>
    </div>
  </div>
  <div class="row admin-page-row">
    <div class="col-8">
      <table class="result-table">
        <th colspan="2">База данных запросов</th>
        <% unless @rec.nil? %>
          <% @rec.each do |record| %>
            <tr>
              <td><%= record.input %></td>
              <td><%= record.json_responce %></td>
            </tr>
          <% end %>
        <% else %>
          <p>База данных пуста</p>
        <% end %>
      </table>
    </div>
    <div class="col-4">
      <%= link_to 'Взять данные из БД', res_path, :class => 'button-def', :remote => true
    %><br/>
    <div id="xml-data-output"></div>
  </div>
</div>
```

### ***JS код асинхронной загрузки данных на страницу (res.js.erb):***

```
var insert = `
  <p>
    <%= @rec %>
  </p>
`;
$('document').ready(function() {
  prepare();
  main();
});
function prepare() {
  $('#xml-data-output').empty();
}
function main() {
  $('#xml-data-output').append(` ${insert} `);
}
```

### ***Сервисный класс для вычисления дружественных чисел (friend\_numbers.rb):***

```
# frozen_string_literal: true

# Friend Number interactor class
class FriendNumbers
  include Interactor

  before :find_in_db

  private

  def find_in_db
    if Calculation.find_res(context.input).nil?
      context.res = calc(context.input)
      Calculation.insert_res(context.input, context.res)
    else
      context.res = Calculation.find_res(context.input)
    end
  end

  def calc(_range)
    arr = []
    par = params[:n].to_i
    par.times do |i|
      par.times do |j|
        arr << [i, j] if compare_nums(i, j) && !arr.include?([j, i]) && i != j
      end
    end
    arr
  end

  def find_dev(x_val)
    res = []
    divider = x_val / 2
    while divider.positive?
      res.push(divider) if (x_val % divider).zero?
      divider -= 1
    end
    res
  end
end
```

### ***Сервисный класс для преобразования объектов БД в удобный для программы формат (convert\_object.rb):***

```
# frozen_string_literal: true

# Converts object from database to array
class ConvertObject
  include Interactor

  def call
    context.res = context.obj.map { |el| [el.input, el.json_response] }
  end
end
```

### ***Модель для вычислений (calculation.rb):***

```
# frozen_string_literal: true

require 'json'

# Model contains each calculation sended by user
class Calculation < ApplicationRecord
  validates :input, uniqueness: true

  class << self
    def find_res(n_val)
      return if Calculation.find_by(input: n_val).nil?

      JSON.parse(Calculation.find_by(input: n_val).json_response)
    end

    def insert_res(n_val, res)
      record = Calculation.new(input: n_val, json_response: res.to_json)
      record.save if record.valid?
    end
  end
end
```

### ***Rspec тест модели (calculation\_spec.rb):***

```
# frozen_string_literal: true

require 'rails_helper'

RSpec.describe 'Calculation', type: :model do
  describe 'add record to database' do
    it 'should add record to database and check it' do
      field = Calculation.new(input: 1500, json_response: '{}')
      Calculation.find_by(input: 1500).destroy unless field.valid?
      field.save
      expect(Calculation.where(input: 1500).exists?).to be_truthy
    end
  end

  context 'if program trying to add not valid options' do
    it 'should say that not valid input is not valid' do
      exs = Calculation.find_by(input: 300)
      Calculation.new(input: 300, json_response: '{}').save if exs.nil?
    end
  end
end
```

```

    expect(Calculation.new(input: 300).valid?).to be_falsy
    Calculation.find_by(input: 300).destroy
    unless exs.nil?
      Calculation.new(input: 300, json_response: exs.json_response).save
    end
  end
end
end
end

```

### ***БД приложения в XML формате:***

```

<?xml version="1.0" encoding="UTF-8"?> <arrays type="array"> <array type="array">
<array type="integer">300</array> <array>[[220,284]]</array> </array> <array
type="array"> <array type="integer">400</array> <array>[[220,284]]</array>
</array> <array type="array"> <array type="integer">500</array>
<array>[[220,284]]</array> </array> <array type="array"> <array
type="integer">1300</array> <array>[[220,284],[1184,1210]]</array> </array> <array
type="array"> <array type="integer">230</array> <array>[]</array> </array> <array
type="array"> <array type="integer">503</array> <array>[[220,284]]</array>
</array> <array type="array"> <array type="integer">504</array>
<array>[[220,284]]</array> </array> <array type="array"> <array
type="integer">302</array> <array>[[220,284]]</array> </array> <array
type="array"> <array type="integer">502</array> <array>[[220,284]]</array>
</array> <array type="array"> <array type="integer">550</array>
<array>[[220,284]]</array> </array> <array type="array"> <array
type="integer">505</array> <array>[[220,284]]</array> </array> <array
type="array"> <array type="integer">30</array> <array>[]</array> </array> <array
type="array"> <array type="integer">0</array> <array>[]</array> </array> </arrays>

```

### ***Пример работы приложения:***

Дружественные числа!

---

Введите диапазон

Найти

Ваши последние результаты:

220 284

*Рис. 1 (Главная страница приложения)*

## Администрирование

База данных запросов		Взять данные из БД
300	[[220,284]]	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;arrays type="array"&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;300&lt;/array&gt; &lt;array&gt;[[220,284]]&lt;/array&gt; &lt;/array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;400&lt;/array&gt; &lt;array&gt;[[220,284]]&lt;/array&gt; &lt;/array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;500&lt;/array&gt; &lt;array&gt;[[220,284]]&lt;/array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;1300&lt;/array&gt; &lt;array&gt;[[220,284],[1184,1210]] &lt;/array&gt; &lt;/array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;230&lt;/array&gt; &lt;array&gt; []&lt;/array&gt; &lt;/array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;503&lt;/array&gt; &lt;array&gt; [[220,284]]&lt;/array&gt; &lt;array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;504&lt;/array&gt; &lt;array&gt; [[220,284]]&lt;/array&gt; &lt;array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;302&lt;/array&gt; &lt;array&gt; [[220,284]]&lt;/array&gt; &lt;array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;502&lt;/array&gt; &lt;array&gt; [[220,284]]&lt;/array&gt; &lt;array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;550&lt;/array&gt; &lt;array&gt; [[220,284]]&lt;/array&gt; &lt;array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;505&lt;/array&gt; &lt;array&gt; [[220,284]]&lt;/array&gt; &lt;array&gt; &lt;array type="array"&gt; &lt;array type="integer"&gt;30&lt;/array&gt; &lt;array&gt; []&lt;/array&gt; &lt;/array&gt;</pre>
400	[[220,284]]	
500	[[220,284]]	
1300	[[220,284],[1184,1210]]	
230	[]	
503	[[220,284]]	
504	[[220,284]]	
302	[[220,284]]	
502	[[220,284]]	
550	[[220,284]]	
505	[[220,284]]	
30	[]	

Рис. 2 (Страница «Администрирование»)

## Тесты приложения:

```

x tetovske@pop-os ~/YAIP/laby/lab11 study rspec -f d
Calculation
  add record to database
    should add record to database and check it
  if program trying to add not valid options
    should say that not valid input is not valid:))
Finished in 0.01499 seconds (files took 0.84806 seconds to load)
2 examples, 0 failures

tetovske@pop-os ~/YAIP/laby/lab11 study
```

Рис. 3 (Rspec тесты)

```

tetovske@pop-os ~/YAIP/laby/lab11 study rubocop app/interactors app/controllers/
Inspecting 5 files
.....
5 files inspected, no offenses detected
tetovske@pop-os ~/YAIP/laby/lab11 study
```

Рис. 4 (Отчет Rubocop)

```

tetovske@pop-os ~/YAIP/laby/lab11 7 study rails c
Running via Spring preloader in process 26024
Loading development environment (Rails 6.0.1)
[1] pry(main)> Calculation.all
(0.1ms) SELECT sqlite_version(*)
Calculation Load (0.1ms) SELECT "calculations".* FROM "calculations"
=> [#<Calculation:0x000055d85343b170
  id: 4,
  input: 300,
  json_response: "[[220,284]]",
  created_at: Mon, 25 Nov 2019 16:59:30 UTC +00:00,
  updated_at: Mon, 25 Nov 2019 16:59:30 UTC +00:00>,
#<Calculation:0x000055d85373ec78
  id: 5,
  input: 400,
  json_response: "[[220,284]]",
  created_at: Mon, 25 Nov 2019 16:59:55 UTC +00:00,
  updated_at: Mon, 25 Nov 2019 16:59:55 UTC +00:00>,
#<Calculation:0x000055d85373e3e0
  id: 6,
  input: 500,
  json_response: "[[220,284]]",
  created_at: Mon, 25 Nov 2019 17:08:53 UTC +00:00,
  updated_at: Mon, 25 Nov 2019 17:08:53 UTC +00:00>,
#<Calculation:0x000055d85373e0e8
  id: 7,
  input: 1300,
  json_response: "[[220,284],[1184,1210]]",
  created_at: Mon, 25 Nov 2019 17:10:06 UTC +00:00,
  updated_at: Mon, 25 Nov 2019 17:10:06 UTC +00:00>,
#<Calculation:0x000055d85373dd28
  id: 8,
  input: 230,
  json_response: "[]",
  created_at: Mon, 25 Nov 2019 17:36:37 UTC +00:00,
  updated_at: Mon, 25 Nov 2019 17:36:37 UTC +00:00>,
#<Calculation:0x000055d85373dbe8
  id: 9,
  input: 503,
  json_response: "[[220,284]]",
  created_at: Wed, 27 Nov 2019 01:07:32 UTC +00:00,
  updated_at: Wed, 27 Nov 2019 01:07:32 UTC +00:00>,
#<Calculation:0x000055d85373d9b8
  id: 10,
  input: 400,
  json_response: "[[220,284]]",
  created_at: Wed, 27 Nov 2019 01:07:32 UTC +00:00,
  updated_at: Wed, 27 Nov 2019 01:07:32 UTC +00:00>]

```

Рис. 5 (Просмотр содержимого БД через команду „rails c“)

**Вывод:** в ходе выполнения лабораторной работы был доработан код ЛР8 таким образом, чтобы все запросы сохранялись в БД. Для реализации бизнес-логики был использован гем “Interactor”. Приложение было протестировано и проверено на соответствие стилю программой Rubocop.