*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)*** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Компьютерные Системы и сети (ИУ6)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 11**

**Название лабораторной работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дисциплина:** Языки интернет программирования

Студент гр. ИУ6-31б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И С Марчук**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

## *Задание* Вариант 1.16.

## ЛР 8. Ruby on Rails

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.

**Текст задания:**

Дано натуральное число m. Написать программу, определяющую такое

натуральное число n, что двоичная запись числа n получается из двоичной записи числа m изменением порядка цифр на обратный порядок их

следования. Например: 6 = 110 , а 3 = 011. Вывести на печать числа и

их двоичное представление.

**ЛР 11. Добавление модели. ORM. Разработка БД, подключение, хранение и поиск данных.**

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 /.

Текст Программы

***routes.rb:***

Rails.application.routes.draw do

root 'start\_page#index'

get 'result' => 'start\_page#result', :as => 'result'

#resources :my\_result\_page

end

***start\_page\_controller.rb:***

class StartPageController < ApplicationController

require 'json'

def index ()

end

before\_action :parse\_params, only: :result

def result

# ищем запись в бд MyResult::ActiveRecord\_Relation

record = (MyResult.where(:pm => @m))[0]

#ActiveRecord::Base.connection.execute("delete from #{MyResult.table\_name} where pm = 144")

if(record) # если запись есть

@where = 'из базы данных'

# выводим ее на страницу

@temp\_arr = JSON.parse(record.result)["result"]

else # если записи нет

@where = 'из расчетов'

# расчитываем результат и выводим ответ на страницу

@temp\_arr = calculate\_result(@m)

# получаем строку результата и создаем запись в бд

result\_in\_db = MyResult.new({pm: @m, result: get\_gson\_string\_from\_result(@temp\_arr)})

result\_in\_db.save

end

end

private

def parse\_params

@m = params[:pm].to\_i

end

def calculate\_result (m)

bin\_m = ''

temp\_arr = Array.new()

ans = 0

bin\_ans = ''

i = 0

temp\_m = m

while temp\_m > 0 do

ans = ans \* 2 + (temp\_m % 2)

bin\_ans = bin\_ans + (temp\_m % 2).to\_s

bin\_m = (temp\_m % 2).to\_s + bin\_m

# новую строку в матрицу

temp\_arr.push Array.new(3)

# ячейки строки

temp\_arr[-1][0] = ans

temp\_arr[-1][1] = bin\_ans

temp\_arr[-1][2] = bin\_m

temp\_m = temp\_m / 2

i = i + 1

end

return temp\_arr

end

def get\_gson\_string\_from\_result(temp\_arr)

return "{\"result\": #{temp\_arr.to\_s}}"

end

end

***index.html.erb:***

<H3>Начальная страница</H3>

<H2>Марчук Иван ИУ6-31б Вариант 16</H2>

<!-- форма ввода отправляющая метод get на страничку result -->

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

<%= label\_tag("Введите число m:") %>

<%= number\_field\_tag(:pm) %>

<br/>

<%= submit\_tag("Посчитать") %>

<% end %>

***result.html.erb:***

<H3>Введенное значение m: <%= @m %></H3>

<H3>Решение:</H3>

<H4>Ответ получен <%= @where %></H4>

<table border="1" bordercolor=black>

<tr><td> № </td><td> Ответ </td><td> дв. ответ </td><td> дв. m </td></tr>

<% @temp\_arr.size.times do |i| %>

<tr>

<td><%= i %></td>

<td><%= @temp\_arr[i][0] %></td>

<td><%= @temp\_arr[i][1] %></td>

<td><%= @temp\_arr[i][2] %></td>

</tr>

<% end %>

</table>

<p>m в двоичной форме: <%= @temp\_arr[-1][2] %></p>

<p>Ответ: <%= @temp\_arr[-1][0] %> В двоичной форме: <%= @temp\_arr[-1][1] %></p>

**Для m = 144:json**

#### {"result": [[0, "0", "0"], [0, "00", "00"], [0, "000", "000"], [0, "0000", "0000"], [1, "00001", "10000"], [2, "000010", "010000"], [4, "0000100", "0010000"], [9, "00001001", "10010000"]]}

**Для m = 144:xml**

<?xml version="1.0" encoding="UTF-8"?>

<hash>

<result type="array">

<result type="array">

<result type="integer">0</result>

<result>0</result>

<result>0</result>

</result>

<result type="array">

<result type="integer">0</result>

<result>00</result>

<result>00</result>

</result>

<result type="array">

<result type="integer">0</result>

<result>000</result>

<result>000</result>

</result>

<result type="array">

<result type="integer">0</result>

<result>0000</result>

<result>0000</result>

</result>

<result type="array">

<result type="integer">1</result>

<result>00001</result>

<result>10000</result>

</result>

<result type="array">

<result type="integer">2</result>

<result>000010</result>

<result>010000</result>

</result>

<result type="array">

<result type="integer">4</result>

<result>0000100</result>

<result>0010000</result>

</result>

<result type="array">

<result type="integer">9</result>

<result>00001001</result>

<result>10010000</result>

</result>

</result>

</hash>

**Файл миграции 20200106210143\_create\_my\_results.rb:**

class CreateMyResults < ActiveRecord::Migration[6.0]

def change

create\_table :my\_results do |t|

t.integer :pm

t.text :result

t.timestamps

end

end

end

**my\_result\_spec.rb**

require 'rails\_helper'

RSpec.describe MyResult, type: :model do

describe 'add record to database' do

it 'should add record to database and check it' do

field = MyResult.new(pm: 1500, result: '{"result":[[0], ["0"], ["0"]]}')

MyResult.find\_by(pm: 1500).destroy unless field.valid?

field.save

expect(MyResult.where(pm: 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 = MyResult.find\_by(pm: 144)

MyResult.new(pm: 144, result: '{"result":[[0], ["0"], ["0"]]}').save if exs.nil?

expect(MyResult.new(pm: 144).valid?).to be\_falsy

MyResult.find\_by(pm: 144).destroy

unless exs.nil?

MyResult.new(pm: 144, result: exs.result).save

end

end

end

end

**start\_page\_controller\_test.rb**

require 'test\_helper'

class StartPageControllerTest < ActionDispatch::IntegrationTest

# test "the truth" do

# assert true

# end

test 'test get root page' do

get root\_path

assert\_response :success

end

test 'test get view page' do

get '/result'

assert\_response :success

end

# удалить если показывать

#test "empty get" do

# get '/result'

# assert\_equal assigns[:bin\_m], ' '

# assert\_equal assigns[:ans], 0

# assert\_equal assigns[:bin\_ans], ' '

#end

#test "get with param m = 144" do

# get '/result', params: {mp: 144}

# assert\_equal " 10010000", assigns[:bin\_m]

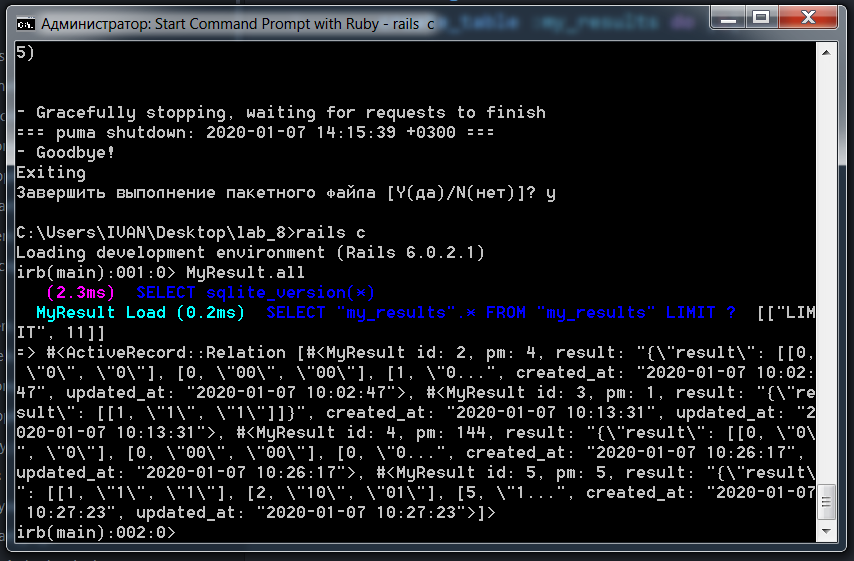
# assert\_equal 9, assigns[:ans]

# assert\_equal " 00001001", assigns[:bin\_ans]

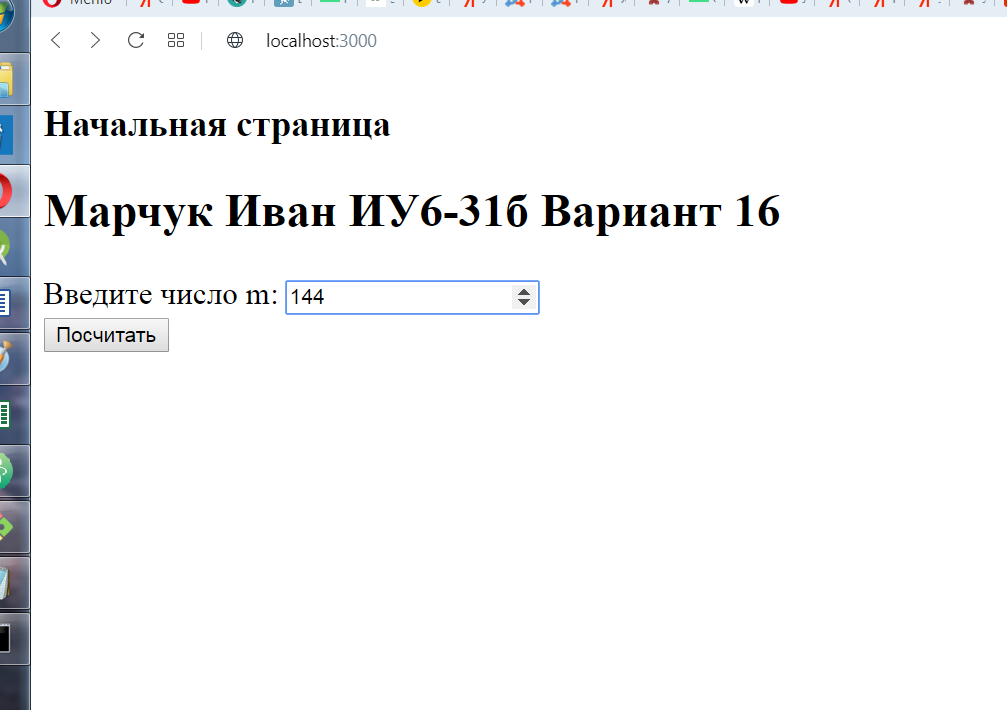
#end

end

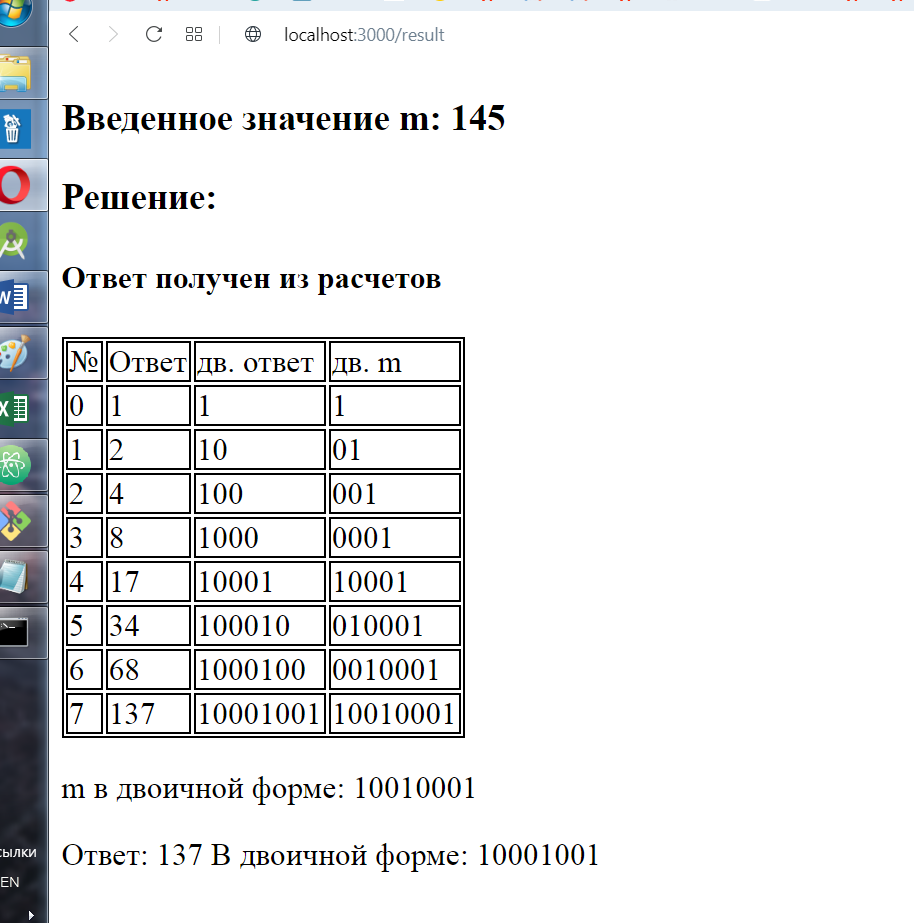
Скриншоты



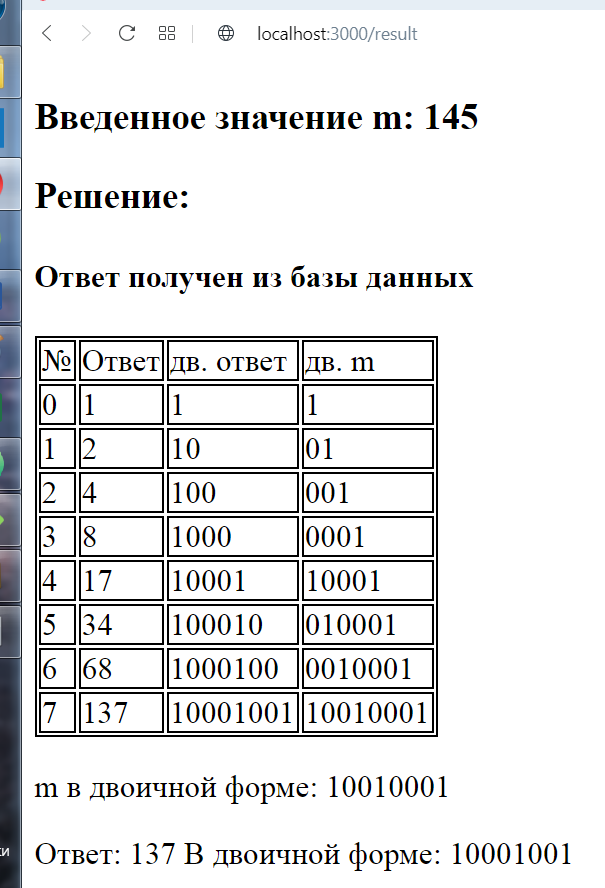
БД приложения в XML формате:



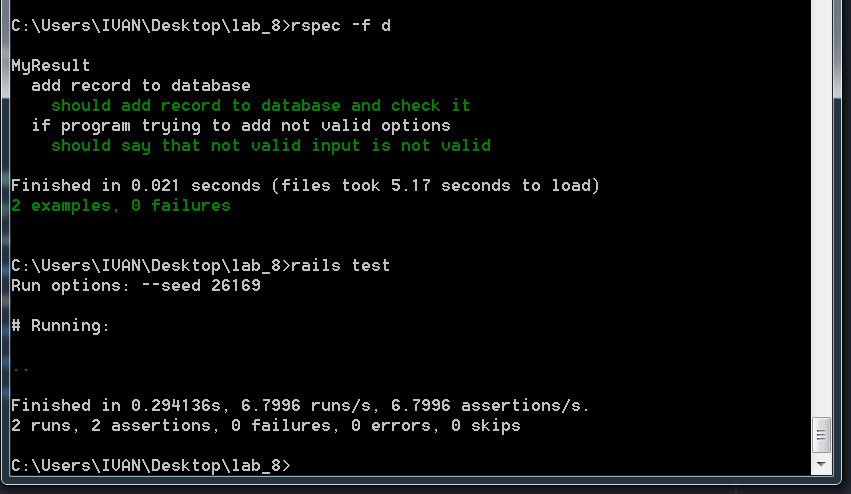
Страница ввода данных



Страница вывода результата (если результат не вычислялся)



Страница вывода результата (если результат вычислялся раньше)



Результаты теста (rspec и rake)

**Вывод:**

Для решения поставленной задачи созданное ранее web-приложение было модифицировано таким образом, чтобы все результаты вычислений сохранялись в БД.