*Федеральное государственное бюджетное образовательное учреждение* *высшего профессионального образования*

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

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

**Отчет**

**по Лабораторной работе №10**

**Дисциплина: Языки интернет программирования**

**Название лабораторной работы: Формирование XML. Отображение XML в HTML средствами сервера и клиента. XSLT. Тесты.**

**Вариант 24**

Студент гр. ИУ6-32 **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Скударнов И. В.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Самарёв Р. С.**

(Подпись, дата) (И.О. Фамилия)

Москва, 2016

**Задание:**

Модифицировать код лабораторной работы 8 таким образом, чтбы по запросу с указанными параметрами выдавался результат в формате XML. Разработать второе приложение, являющееся посредником между клиентом и первым приложением, задачей которого является преобразование XML в HTML или передача в неизменном виде браузеру для отображения браузером.

**Код:**

**palind\_controller.rb – контроллер первого приложения**

class PalindController < ApplicationController

def input

end

def palind(n)

mas = Array.new(n+1)

mas[0] = 0

for i in 11..n

ik = i\*\*2

if ik.to\_s == ik.to\_s.reverse

mas[i] = ik

mas[0] += 1

puts "Число #{i}, его квадрат #{mas[i]}"

else

mas[i]=nil

end

end

return mas

end

def view

@n1 = params[:n1].to\_i

@result = palind(@n1)

format\_answer(@n1, @result)

end

def format\_answer(n1, result)

respond\_to do |format|

format.html

format.xml do

render xml: {

result: {

description: 'Result Array',

type: 'Array', value: result

}

}

end

end

end

end

**palind\_controller\_test.rb – функциональные тесты контроллера**

require 'test\_helper'

class PalindControllerTest < ActionDispatch::IntegrationTest

test "should get input" do

get palind\_input\_url

assert\_response :success

end

test "should get view" do

get palind\_view\_url

assert\_response :success

end

test 'should get 3' do

get palind\_view\_url, params: {n1: 30}

answer = 3

result = controller.instance\_variable\_get(:@result)[0]

assert\_equal answer, result

end

test 'should get 5' do

get palind\_view\_url, params: {n1: 112}

answer = 5

result = controller.instance\_variable\_get(:@result)[0]

assert\_equal answer, result

end

test 'should correct format xml to hash' do

get '/palind/view.xml', params: {n1: 50}

answer = {

'hash' => {

'result' => {

'description' => "Result Array",

'type' => 'Array', 'value' => assigns[:result]

}

}

}

assert\_equal answer, Hash.from\_xml(response.body)

end

test 'should have correct result tag' do

get '/palind/view.xml', params: {n1: 50}

assert\_select 'result' do

assert\_select 'type', 'Array'

result\_array\_value\_answer = { 'value' => assigns[:result]}

assert\_select 'value>value', result\_array\_value\_answer

end

end

end

**transformer.xslt – код XSLT-программы**

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

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<p>palindroms were found<p>

<div class='table'>

<div class='row header'>

<div class='cell'>Number</div>

<div class='cell'>Squared Number</div>

</div>

<xsl:for-each select='//result/value/value'>

<xsl:choose>

<xsl:when test="not(. = '') and not(position() = '1')">

<div class='row'>

<div class='cell'><xsl:value-of select="position()"/></div>

<div class='cell'><xsl:value-of select="."/></div>

</div>

</xsl:when>

</xsl:choose>

</xsl:for-each>

</div>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

**xml\_transformer\_controller.rb – контроллер второго приложения**

require 'nokogiri'

class XmlTransformerController < ApplicationController

def input

end

def answer

page = Nokogiri::XML(open("http://localhost:3000/palind/view.xml?utf8=%E2%9C%93&n1=#{params[:n1]}&commit=Show+palindroms"))

xslt = Nokogiri::XSLT(File.read('public/transformer.xslt'))

xslt\_href = Nokogiri::XML::ProcessingInstruction.new(page, 'xml-stylesheet', 'type="text/xsl" href="transformer.xslt"')

request.format = params[:format].to\_sym

respond\_to do |format|

format.html { render html: xslt.transform(page).to\_s.html\_safe }

format.xml do

page.at('hash').add\_previous\_sibling xslt\_href

render xml: page

end

end

end

end

**input.html.erb**

<h1>XmlTransformer#input</h1>

<p>Find me in app/views/xml\_transformer/input.html.erb</p>

<div>

<%= form\_tag('/palind/view', :method => 'get') do %>

<div>

<%= label\_tag('Enter n', nil, for: 'input\_n1') %>

<%= text\_field\_tag(:n1, nil, id: 'input\_n1') %>

</div>

<br/>

<div>

<%= radio\_button\_tag(:format, 'html', true, id: 'choose\_html') %>

<%= label\_tag('html\_answer', ('HTML answer').html\_safe, for: 'choose\_html') %>

<%= radio\_button\_tag(:format, 'xml', false, id: 'choose\_xml') %>

<%= label\_tag('xml\_answer', ('XML answer').html\_safe, for: 'choose\_xml') %>

</div>

<br/>

<div>

<%= submit\_tag("Show palindroms") %>

</div>

<% end %>

</div>

**xml\_transformer\_controller\_test.rb – тесты контроллера**

require 'test\_helper'

class XmlTransformerControllerTest < ActionDispatch::IntegrationTest

test "should get input" do

get xml\_transformer\_input\_url

assert\_response :success

end

test 'should get answer html' do

get '/palind/view', params: { format: 'html' }

assert\_response :success

end

test 'should get answer xml' do

get '/palind/view', params: { format: 'xml' }

assert\_response :success

end

test 'result from first app shold be correct' do

n1\_value = rand(1000)

page = Nokogiri::XML(open("http://localhost:3000/palind/view.xml?n1=#{n1\_value}"))

assert\_equal palind(n1\_value), page.at('result>value').elements.map { |el| el.children.text.to\_i }

end

def palind(n)

mas = Array.new(n+1)

mas[0] = 0

for i in 1..10

mas[i]=0

end

for i in 11..n

ik = i\*\*2

if ik.to\_s == ik.to\_s.reverse

mas[i] = ik

mas[0] += 1

puts "Число #{i}, его квадрат #{mas[i]}"

else

mas[i]=0

end

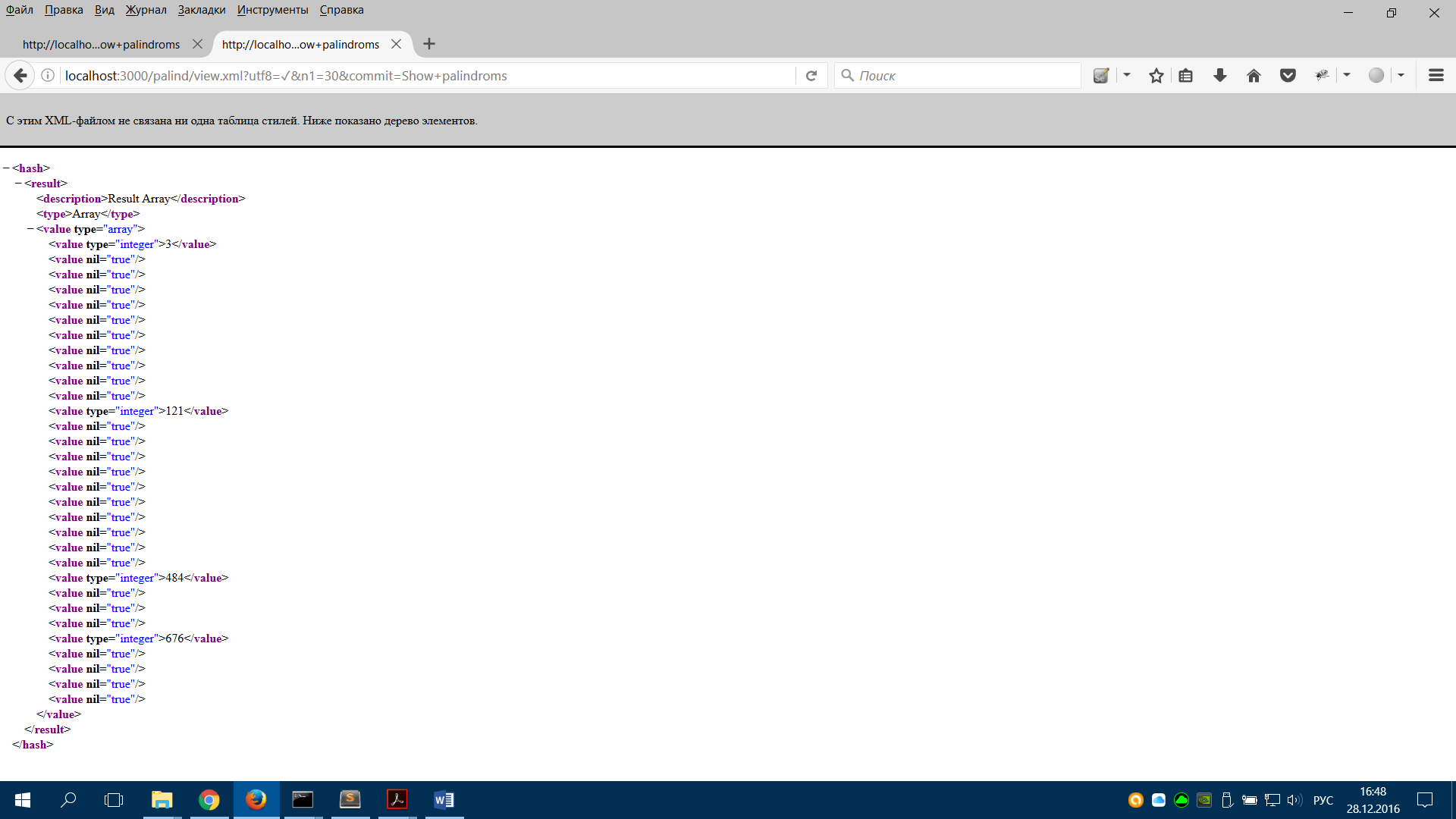
end

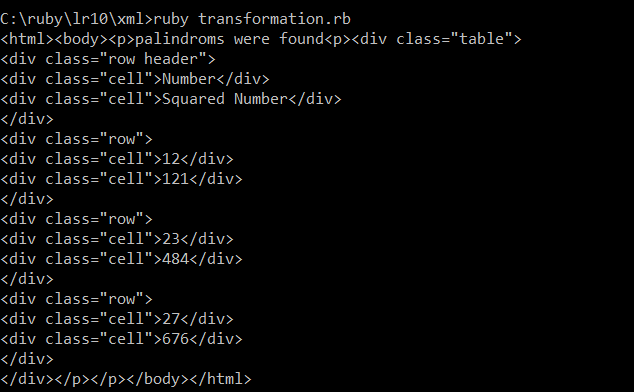
return mas

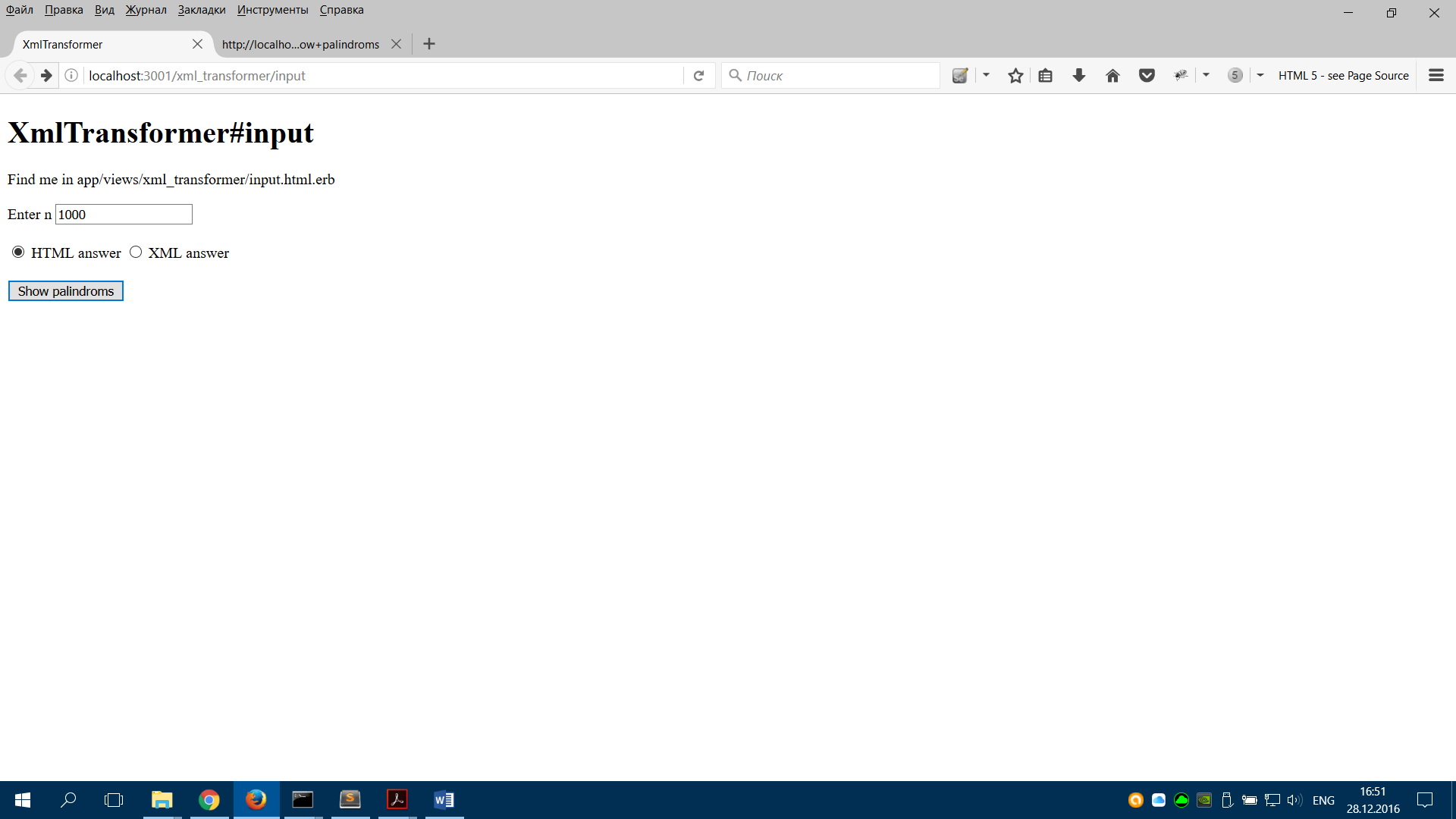
end

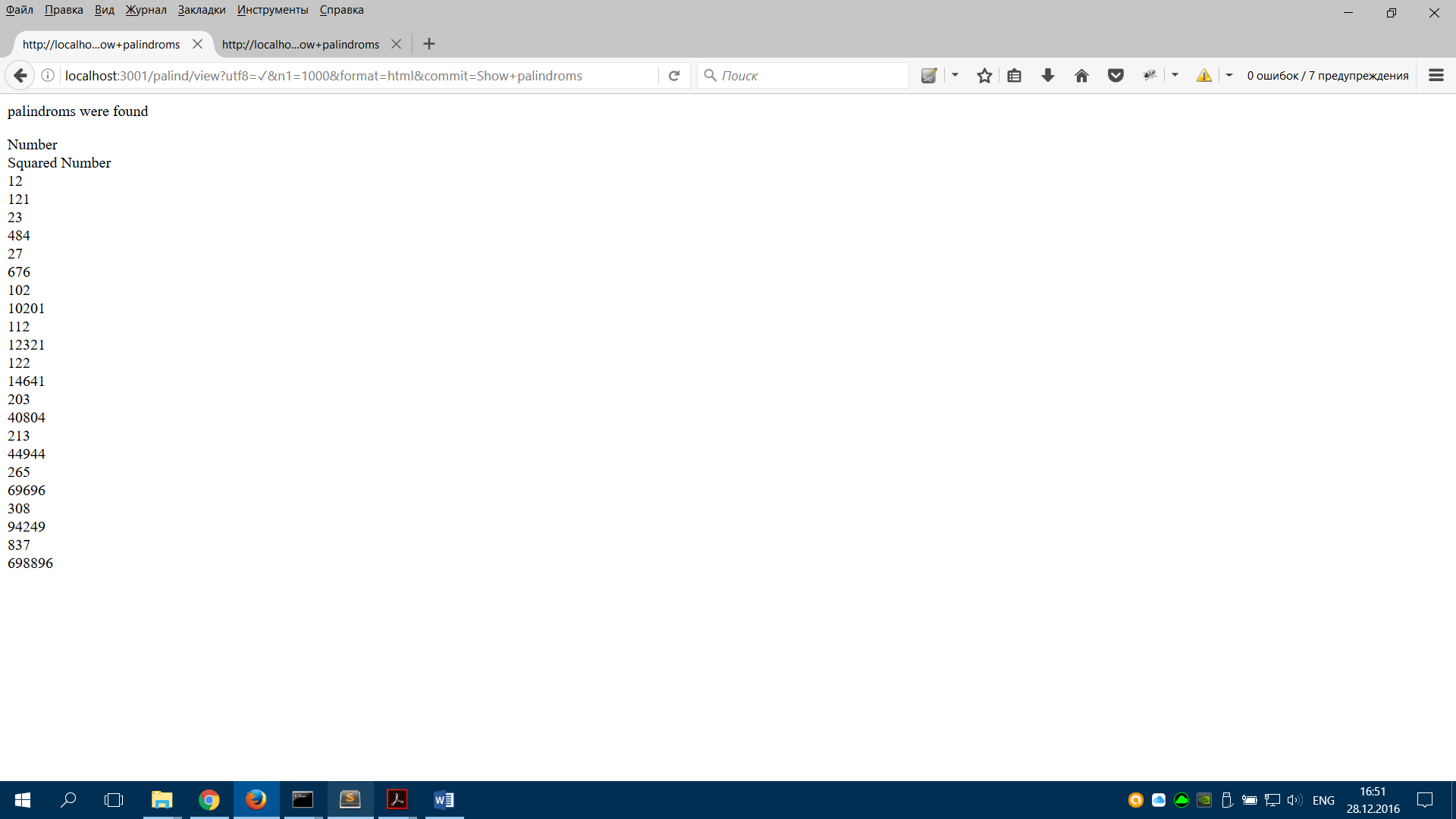
end

**Скриншоты:**

****

****

****

****

**Вывод:** научился организовывать вывод информации в формате XML. Проверять формирование XML. Разрабатывать XSLT-программу преобразования XML в HTML.