

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



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

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

О т ч е т

Лабораторная работа 10

Студент гр.ИУ6-32 _____ **Липко Д.Н.**
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____
(Подпись, дата) (И.О. Фамилия)

Задание 1

Модифицировать код ЛР 8 таким образом, чтобы по запросу с указанными параметрами выдавался результат в формате XML (средствами стандартной сериализации ActiveSupport).

- Проверить формирование XML и сохранить в файл для отладки XSLT и второго приложения.
- Написать функциональный тест, проверяющий формат выдаваемых данных при запросе RSS.

Разработать XSLT-программу преобразования полученной XML в HTML.

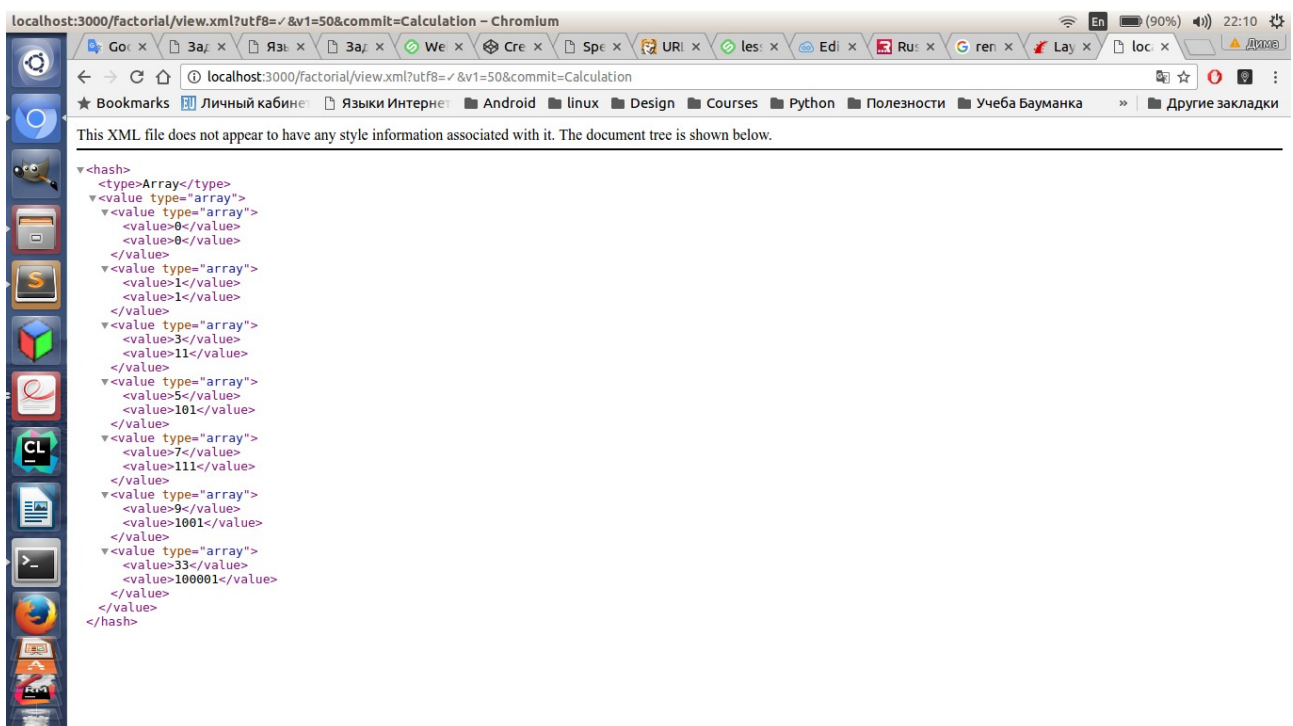
Добавить в проверяемый XML-файл строку привязки к преобразованию `<?xml-stylesheet type="text/xsl" href="some_transformer.xslt"?>`. Проверить корректность отображения браузером результата преобразования.

Проверить на автономной Ruby-программе корректность преобразования, используя следующий фрагмент кода:

```
require 'nokogiri'

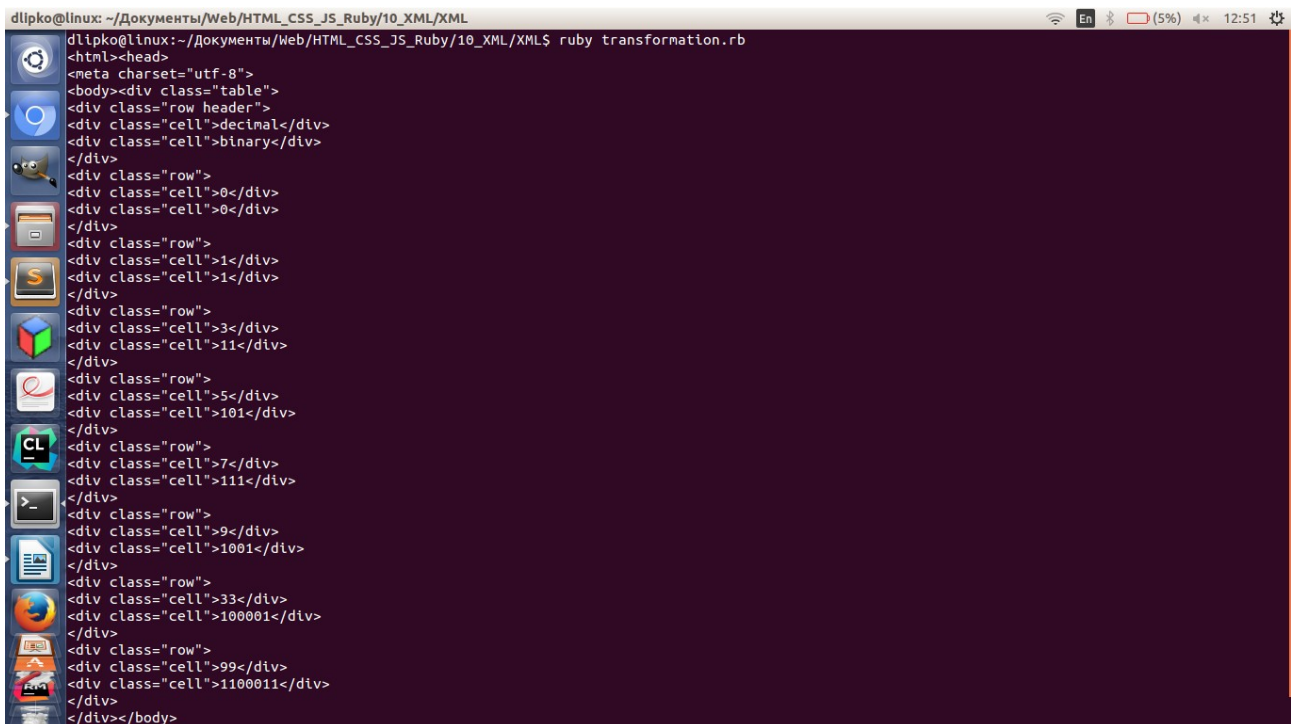
doc = Nokogiri::XML(File.read('some_file.xml'))
xslt = Nokogiri::XSLT(File.read('some_transformer.xslt'))

puts xslt.transform(doc)
```



Transformer.xslt

```
<?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>
      <head>
        <meta charset="utf-8">
      </head>
      <body>
        <div class='table'>
          <div class='row header'>
            <div class='cell'><xsl:value-of
select="//text/decimal/value"/></div>
            <div class='cell'><xsl:value-of
select="//text/binary/value"/></div>
            <xsl:for-each select='//result/value/value'>
              <div class='row'>
                <div class='cell'><xsl:value-of select="value"/></div>
                <div class='cell'><xsl:value-of select="value/following-
sibling::*"/></div>
              </xsl:for-each>
            </div>
          </div>
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```



```
dlipko@linux: ~/Документы/Web/HTML_CSS_JS_Ruby/10_XML/XML$ ruby transformation.rb
<html><head>
<meta charset="utf-8">
<body><div class="table">
<div class="row header">
<div class="cell">decimal</div>
<div class="cell">binary</div>
</div>
<div class="row">
<div class="cell">0</div>
<div class="cell">0</div>
</div>
<div class="row">
<div class="cell">1</div>
<div class="cell">1</div>
</div>
<div class="row">
<div class="cell">3</div>
<div class="cell">11</div>
</div>
<div class="row">
<div class="cell">5</div>
<div class="cell">101</div>
</div>
<div class="row">
<div class="cell">7</div>
<div class="cell">111</div>
</div>
<div class="row">
<div class="cell">9</div>
<div class="cell">1001</div>
</div>
<div class="row">
<div class="cell">33</div>
<div class="cell">100001</div>
</div>
<div class="row">
<div class="cell">99</div>
<div class="cell">1100011</div>
</div>
</div></body>
```

Задание 1

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.

Приложение должно запускаться с указанием номера порта TCP, отличным от номера порта первого приложения (например rails server -p 3001)!

- Подготовить каркас приложения, а также форму формирования запроса, форму отображения результата и соответствующие действия контролера.
- Добавить в контроллер преобразование XML в HTML с помощью ранее разработанного XSLT-файла.
- Подключить запрос XML с первого приложения и проверить работу приложений в связке.
- Написать функциональный тест, проверяющий что при различных входных данных результат генерируемой страницы различен.
- Доработать код контроллера и представлений данного приложения для выдачи браузеру XML-потока в неизменном виде (организовать возможность выбора формата выдачи для пользователя).
- Проверить, что браузер получает XML первого приложения в неизменном виде.
- Доработать код контроллера приложения таким образом, чтобы XML-поток первого приложения получал дополнительную строку, указывающую xsl. Модифицировать форму запроса параметров таким образом, чтобы браузер получал в ответ XML. При этом разместить XSLT-файл в директории public.
- Проверить, что браузер производит преобразование XML->HTML в соответствии с xlt.
- Реализовать функциональные тесты второго приложения. Проверить результаты, формируемые приложением, на соответствие выбранному формату выдачи.

Итоговая форма ввода параметра должна содержать кнопки или селектор, позволяющие проверить два варианта преобразования:

- Серверное xml+xslt->html
- Клиентское xml+xslt->html

Factorial_controller.rb

```
class FactorialController < ApplicationController
  def input
    end

  def binary (number)
    number_2 = "0" if number == 0
    while number != 0
      mod = number % 2
      number /= 2
      number_2 = mod.to_s + number_2.to_s
    end
    number_2
  end

  def view
    @result = []
    params[:v1].to_i.times { |number|
      number_10 = number.to_s
      number_2 = binary number
      if number_10.reverse.eql?( number_10 ) and number_2.reverse.eql?
( number_2 )
        @result += [[number_10, number_2]]
      end
    }

    respond_to do |format|
      format.html
      format.xml do
        render xml: {
          result: {
            description: 'Result',
            type: @result.class.to_s,
            value: @result
          },
          text: {
            decimal: { type: 'String', value: 'decimal' },
            binary: { type: 'String', value: 'binary' },
            repeatcalculation: { type: 'String', value: 'repeat_calculation' }
          }
        }
      end
    end
  end
end
```

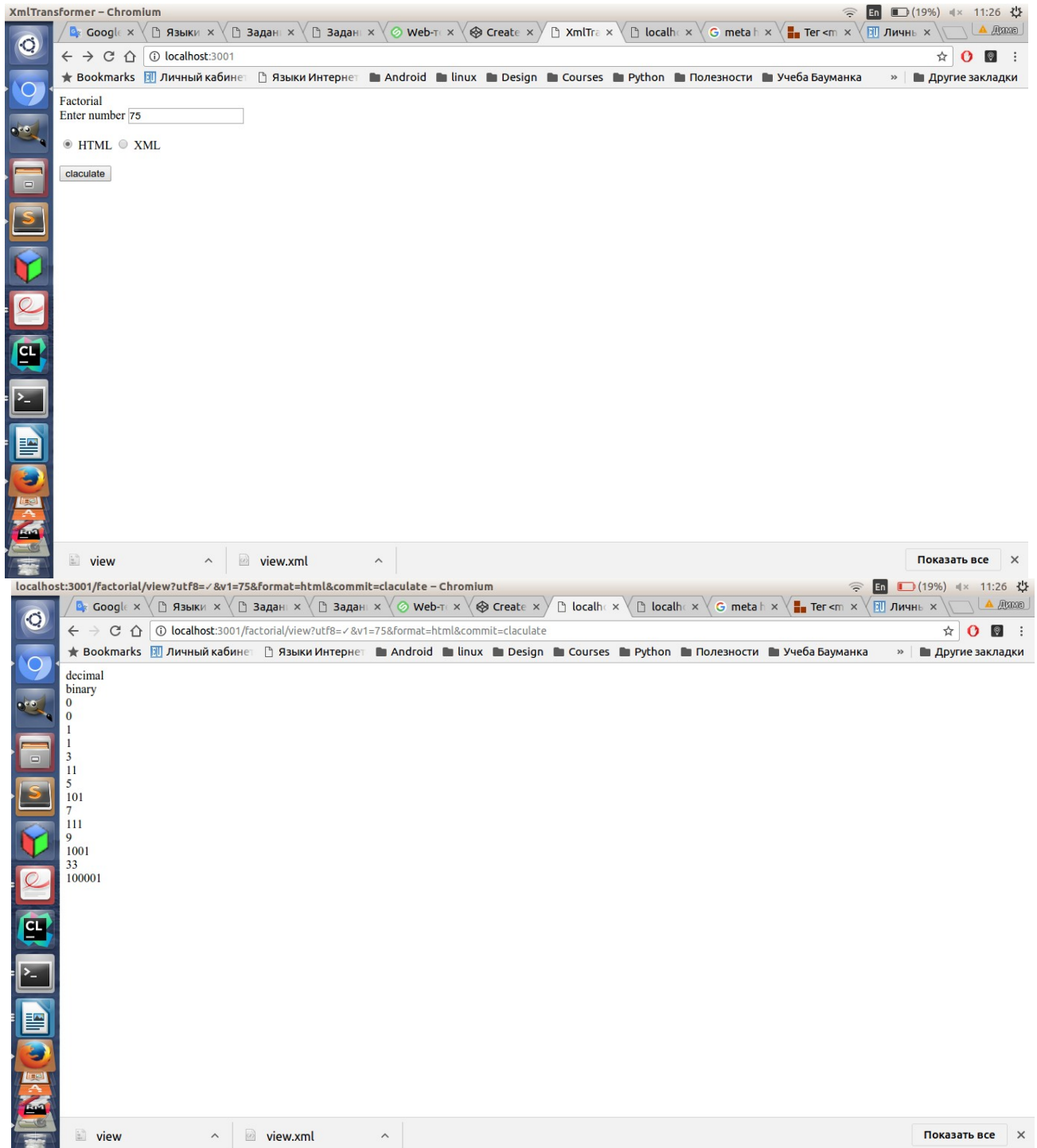
Input.html.erb

```
<div class="valbut">
  <%= form_tag("/factorial/view.xml", :method => "get") do %>
    <%= label_tag("Value:") %>
    <%= text_field_tag(:v1) %> <br/>
  <br/>
  <%= submit_tag("Calculation") %>
<% end %>
</div>
```

View.html.erb

```
<div class='valbut'>
  <div class='row header'>
    <div class='cell'><%= t('Decimal') %></div>
    <div class='cell'><%= t('Binary') %></div>
  </div>
  <% if @result != " %>
    <% @result.each do |x| %>
      <div class='row'>
        <div class='cell'><%= x[0] %></div>
        <div class='cell'><%= x[1] %></div>
      </div>
    <% end %>
  <% else %>
    <div class='row'>
      <div class='cell'><%= 'Такие значения' %></div>
      <div class='cell'><%= ' не найдены' %></div>
    </div>
  <% end %>
  <div class="space"></div>
  <%= link_to "Repeat calculation", :factorial_input %>
</div>
```

Интерфейс



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