Rinktiniai Java skyriai

Pratybos #4

2007-10-04

JSP pagrindai

Tobulybė paprastume

- It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience.

  - Albertas Einšteinas

- Everything should be made as simple as possible, but not simpler.

- Keep it simple, stupid!

- KISS

Kuo blogi servletai

Su servletu išspausdiname paprastą HTML puslapį

```
import java.io.*;
import java.servlet.*;
import java.servlet.http.*;
import java.util.*;

public class TestPage extends HttpServlet {
    public void doGet(
        HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
    {
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("  <title>Testas</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("  <h1>Testas</h1>");
        out.println("  <p>Šios dienos data: " + new Date() + "</p>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Kuo JSP geriau už servletus

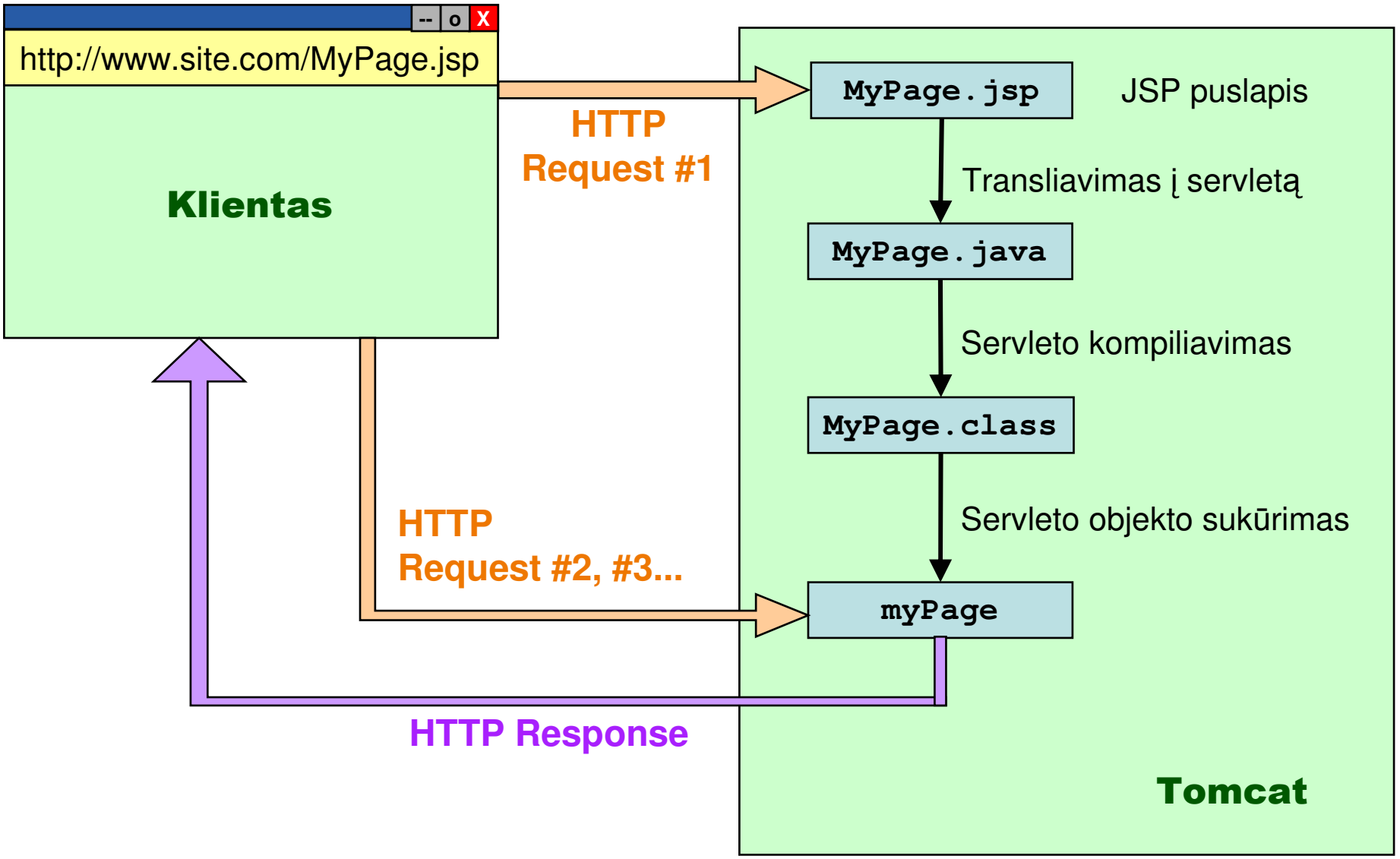
Tą patį puslapį pagaminame su JSP

```
<%@ page import="java.util.*" %>
<html>
<head>
  <title>Testas</title>
</head>
<body>
  <h1>Testas</h1>
  <p>Šios dienos data: <%= new Date() %></p>
</body>
</html>
```

JSP ir servletų ryšys

- JSP gali viską, ką gali servletai
- ... nes kiekvienas JSP puslapis **yra** servletas

JSP kompiliavimas



Dinaminė puslapių dalis

JSP reiškiniai

```
Šios dienos data: <%= new Date() %>

Studento vardas: <%= resultSet.getString("fname") %>

Puslapio parametras: <%= request.getParameter("id") %>
```

Dinaminė puslapių dalis

JSP skriptletai

```
Šios dienos data: <% out.println(new Date()); %>

<% if (searchEngine.equals("Google")) { %>
   <a href="http://www.google.com">Google</a>
<% } else { %>
   <a href="http://www.yahoo.com">Yahoo!</a>
<% } %>

<table>
   <% for (Student stud: studentList) { %>
      <tr><td><%= stud.getName() %></td></tr>
   <% } %>
</table>
```

Dinaminė puslapių dalis

Specialūs kintamieji (jų nereikia aprašyti)

```
request     - HttpServletRequest
response    - HttpServletResponse
session     - HttpSession
out         = response.getWriter()
application - ServletContext
exception   - tik puslapyje, kur isErrorPage="true"
```

Gali būti naudojami JSP reiškiniuose ir skriptletuose

Dinaminė puslapių dalis

JSP deklaracijos (metodų ir laukų aprašai)

```
<%!
private int count = 0;
private synchronised int nextCount() {
    count++;
    return count;
}
%>
<p>
    Jūs esate <%= nextCount() %> šio puslapio lankytojas.
</p>
```

Dinaminė puslapių dalis

JSP direktyva “page”

```
<%@ page import=“java.util.*, myPackage.*” %>
```

Sekančių paketų importuoti nebūtina, nes jie importuojami pagal nutylėjimą:

```
java.lang.*
javax.servlet.*
javax.servlet.jsp.*
javax.servlet.http.*
```

Dinaminė puslapių dalis

JSP direktyva “page”

```
<%@ page
    import=“.........”
    contentType=“text/html; charset=utf-8”
    errorPage=“error.jsp”
%>
```

Dinaminė puslapių dalis

JSP direktyva “page”

```
<%@ page isErrorPage=“true” %>
<html>
<body>
<h1>Įvyko klaida!</h1>
<p>
  <%= exception.getMessage() %>
</p>
<pre>
  <% exception.printStackTrace(new PrintWriter(out)); %>
</pre>
</body>
</html>
```

Dinaminė puslapių dalis

JSP direktyva “include”

```
<html>
<head>
  <%@ include file=“head.html” %>
</head>
<body>
  <%@ include file=“menu.jsp” %>
  ... puslapio turinys ...
  <%@ include file=“/WEB-INF/footer.jsp” %>
</body>
</html>
```

JSP vs servletai

- JSP labiausiai tinka:
  - prezentacijai (informacijos atvaizdavimui)
  - HTML, XML ir kitokių tekstinių formatų išvedimui
    - jeigu nemaža dalis jų turinio yra statiška
- Servletai labiausiai tinka:
  - logikai (informacijos apdorojimui)
  - visiškai dinaminio turinio išvedimui
  - binarinių duomenų išvedimui

JSP vs servletai

“Use the right tool for the right job”

Best practices

- Kiek kodo dėti į JSP puslapį?
  - Kuo mažiau
  - Stenkitės kuo daugiau kodo perkelti į klases ir servletus
  - JSP puslapiuose kreipkitės į savo klases – deleguokite joms kuo daugiau vykdymo logikos
  - JSP puslapyje palikite tik prezentacijos logiką

Best practices

- Kodėl reikia kodą kelti į klases?
  - Nes kodas, esantis klasėse, yra:
    - lengviau rašomas
    - lengviau kompiliuojamas
    - lengviau testuojamas
    - lengviau debug'inamas
    - pakartotinai panaudojamas
  - Nes kuo mažiau kodo JSP puslapiuose:
    - tuo jie suprantamesni
    - tuo paprasčiau juos keisti

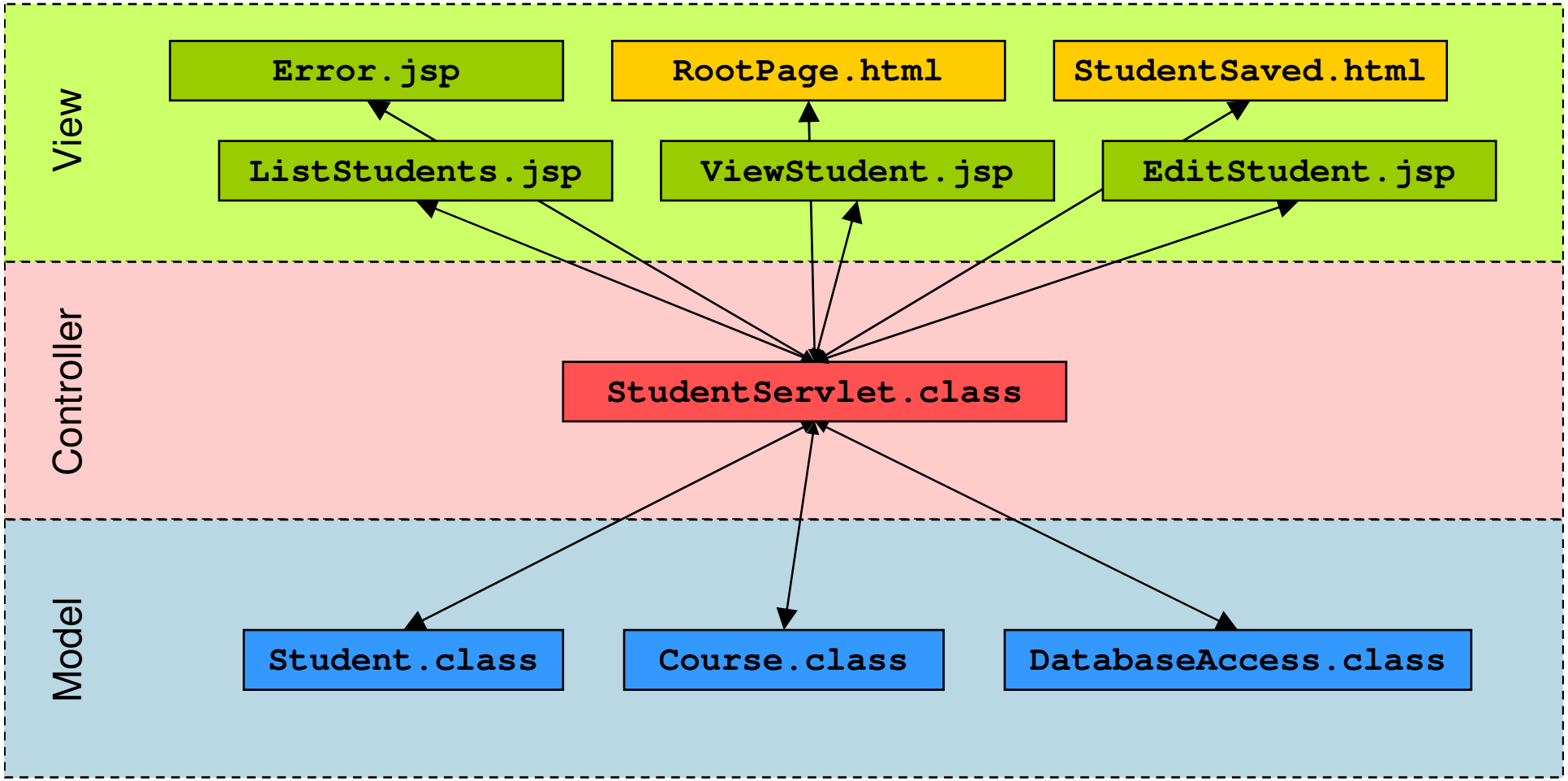
Best practices

- Kiek kodo dėti į servletą?
  - Daugiau nei į JSP :)
  - Bet mažiau nei į specializuotas klases

MVC

- Model-View-Controller (MVC) - architektūrinis pattern'as, kuris atskiria duomenų struktūrą nuo jos atvaizdavimo
  - Model – duomenų struktūra (Java klasės)
  - View – duomenų atvaizdavimas (JSP, HTML ...)
  - Controller – tarpininkas tarp Model ir View:
    - kiekvienam "modeliui" parenka "vaizdą"
    - reaguodamas į įvykius "vaizde", keičia "modelį"
- Servletai atlieka "kontrolieriaus" vaidmenį:
  - duomenų apdorojimą deleguoja klasėms
  - duomenų atvaizdavimą deleguoja JSP puslapiams

MVC


View
Error.jsp
RootPage.html
StudentSaved.html
ListStudents.jsp
ViewStudent.jsp
EditStudent.jsp
Controller
StudentServlet.class
Model
Student.class
Course.class
DatabaseAccess.class

Kaip iš servleto kviesti JSP

- `response.sendRedirect("/MyPage.jsp");`
  - vartotojas mato adreso (URL) pasikeitimą
  - ypač tinka apdorojus POST užklausą – tuomet galima permesti vartotoją į "rezultatų" puslapį
-
```
request.setAttribute("student", new Student("Jonas"));
ServletContext context = getServletContext();
RequestDispatcher dispatcher =
    context.getRequestDispatcher("/WEB-INF/Stud.jsp");
dispatcher.forward(request, response);
```
  - vartotojas nemato adreso pasikeitimo
  - taip galima JSP puslapiams perduoti duomenų struktūrą, t.y. "modelį", kurį reikia atvaizduoti
  - JSP atvaizduoja **request** objekte esančią struktūrą:

```
<%= ((Student) request.getAttribute("Student")).getName() %>
```