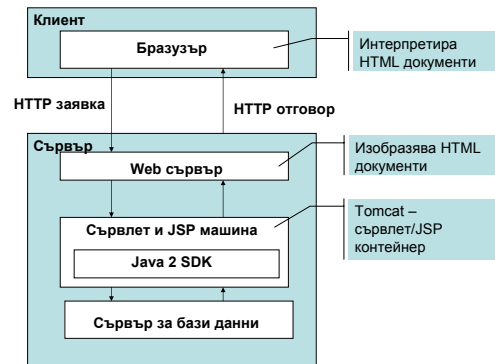


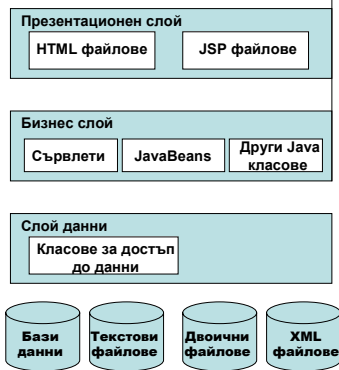
Servlets и JSP (JavaServer Pages) технологии

1. Web приложение – множество от Web страници, които се генерират в резултат на заявки на потребители.

2. Компоненти на Java web приложение



3. Архитектура на Java web приложение



Сървлет

1. **Сървлет** – Java приложение, което се изпълнява от сървъра и разширява възможностите за достъп до приложенията, разположени на сървъра, чрез програмния модел заявка-отговор.

Аплет – използва се от клиента за получаване на динамично съдържание.

2. Предимства пред CGI технологията

- платформено независим код;
- не създава нови процеси при всяко изпълнение на сървлета;
- управлява се от отделна нишка в процеса на Web сървъра;
- може да пише в log файловете на сървъра.

3. Архитектура на сървлет

- пакети `javax.servlet` и `javax.servlet.http`;
- клас, който:
 - реализира интерфейса `Servlet`;
 - или наследява класа `GenericServlet` (реализация на обвивката на `Servlet`) – рядко;
 - или наследява класа `HttpServlet`.

```
public interface Servlet {
    public void init(ServletConfig config) throws ServletException;
    public ServletConfig getServletConfig();
    public void service(ServletRequest req, ServletResponse res)
        throws ServletException, IOException;
    public String getServletInfo();
    public void destroy();
}
```

4. Жизнен цикъл на сървлет – управлява се от контейнера, в който сървлетът е разположен

- ако не съществува инстанция на сървлета, контейнерът:
 - зарежда сървлетния клас;
 - създава инстанция на сървлетния клас;
 - инициализира инстанцията на сървлета като извиква метода `init`;
- извиква метода `service`, като му предава обекта-заявка и обекта-отговор;
- ако контейнерът трябва да премахне сървлета, извиква метода `destroy`.

Инициализиране на сървлет – предефинира метода `init` на интерфейса `Servlet`.

Обслужващи методи – реализират:

- методи на клас, реализиращ интерфейса `Servlet`;
- метода `service` на класа `GenericServlet`;
- методи `doGet`, `doDelete`, `doOptions`, `doPost`, `doPut`, `doTrace` на класа `HttpServlet`.

Обслужващият метод:

- извлича информация от заявката;
- осигурява достъп до външни ресурси;
- конструира отговора:
 - получава изходен поток;
 - попълва заглавията (`<head>`);
 - записва съдържанието в тялото (`<body>`) на изходния поток.

Клас `HttpServlet` (`javax.servlet.http`)

Абстрактен клас за създаване на HTTP сървлет, подходящ за Web сайт.

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, java.io.IOException
```

Извиква се от сървъра (чрез метода `service`), за да позволи на сървлета да обработва HTTP `GET` заявка.

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, java.io.IOException
```

Извиква се от сървъра (чрез метода `service`), за да позволи на сървлета да обработва HTTP `POST` заявка (клиентът изпраща данни с неограничена дължина към сървлета единствен път – напр. номер на кредитна карта).

```
protected void doPut(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, java.io.IOException
```

Извиква се от сървъра (чрез метода `service`), за да позволи на сървлета да обработва HTTP `PUT` заявка (клиентът поставя файл на сървъра – аналогично на изпращане на файл чрез `FTP`).

```
protected void doDelete(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, java.io.IOException
```

Извиква се от сървъра (чрез метода `service`), за да позволи на сървлета да обработва HTTP `DELETE` заявка (клиентът премахва документ или `Web` страница от сървъра).

```
protected void doTrace(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, java.io.IOException
```

Извиква се от сървъра (чрез метода `service`), за да позволи на сървлета да обработва HTTP `TRACE` заявка.

Извличане на информация от заявката

– отговорът съдържа данни, предадени от сървъра към клиента;

– реализира интерфейса `ServletResponse`, дефиниращ методи за:

– получаване на изходен поток:

– за символни данни използва `PrintWriter`, върнат от метода `getWriter` на отговора;

```
public PrintWriter getWriter() throws IOException
```

– за двоични данни използва `ServletOutputStream`, върнат от метода `getOutputStream` на отговора;

– тип на съдържанието (`text/html`) – чрез метода `public void setContentType(String type)`

- буфериране на изхода – чрез метода `setBufferSize(int)`;
- информация за локализация;
- статус кодове за причината за неудовлетворяване или пренасочване на заявката;
- специфична информация за клиента (`cookies`).

Конструироване на отговора

– заявката съдържа данни, предадени от клиента към сървлета;

– реализира интерфейса `ServletRequest`, дефиниращ методи за достъп до:

– параметри `getParameter`, които се предават от клиента към сървлета;

```
public String getParameter(String name)
```

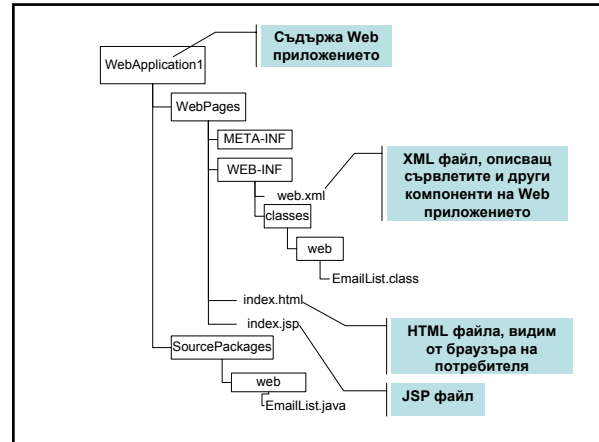
– атрибути с информация от контейнера към сървлета или между сервлети;

– тип на протокола;

– локализираща информация.

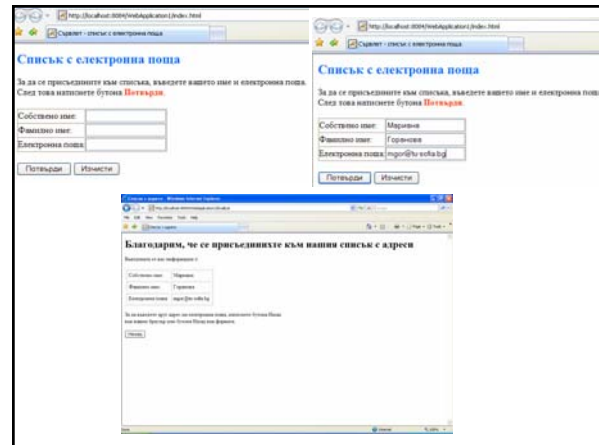
Apache Tomcat – Web сървър и сървлет и JSP машина

- **инсталиране на Tomcat**
- **копиране на файла servlet.jar от директорията <tomcat-install>\common\lib в директорията jre\lib\ext на Java SDK.**



Пример: Списък с електронни адреси

Включва HTML страница, намираща се на сървър, която съдържа текстови полета за въвеждане на имена и електронен адрес, и два бутона Submit и Reset. Потребител интерпретира тази страница на своя брауър. При натискане на бутона Submit HTML страницата извиква сървлет, който предава въведените данни от потребителя и връща HTML страница, която изобразява въведената информация от потребителя и бутон Return за повторно въвеждане на информация. При натискане на бутона Reset се изчистват текстовите полета.



index.html

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title> Сървлет - списък с електронна поща</title>
</head>
<body>
<h2><font color=#0066FF>Списък с електронна поща</font></h2>
За да се присъедините към списъка, въведете вашето име и електронна поща.<br>
След това натиснете бутона <b><font color=#FF3300>Потвърди.</b></font>
<br>
<form method="post" action="EmailList">
<table cellspacing="5" cellpadding="5" border="1">
<tr><td align="left"> Собствено име:</td>
<td><input type="text" name="firstName"></td></tr>
<tr><td align="left"> Фамилно име:</td>
<td><input type="text" name="lastName"></td></tr>
<tr><td align="left"> Електронна поща:</td>
<td><input type="text" name="emailAddress"></td></tr>
</table>
<p><input type="submit" value="Потвърди">
<input type="reset" value="Изчисти"></p>
</form>
</body>
</html>

```

HTML код за форма

(за разполагане на контроли)

```

<form method="post" action="EmailList">
...
</form>

```

action – URL на сървлета или JSP, който се извиква от потребителя при натискане на бутона Submit;
method – HTTP метод (get/post), който ще използва брауъра за HTTP заявката.

Контроли

Текстово поле

```
<input type="text" name="firstName">
```

type – тип на текстовото поле (text/password/hidden);

name – име на контрола, което се използва от сървлета/JSP за достъп до данните в контрола.

Бутон

```
<input type="submit" value="Потвърди">
```

type – тип на бутона:

- submit – активира атрибута action на формата;
- reset – установява всички контроли във формата в начално състояние;
- button – създава JavaScript бутон;

value – име на бутона.

EmailList.java

```
package web;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class EmailList extends HttpServlet {
    // Обработка HTTP POST заявка
    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        // Установява типа на съдържанието на отговора, който ще се изпрати
        // на клиента – html текст
        response.setContentType("text/html;charset=windows-1251");
        request.setCharacterEncoding(response.getCharacterEncoding());

        // Получава изходен поток за символни данни
        PrintWriter out = response.getWriter();

        // Получава параметрите, предадени от клиента към сървлета
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
        String emailAddress = request.getParameter("emailAddress");
```

```
// Запис на текст в html страницата
out.println("<html>\n" // HTML документ
+"<head>\n" // заглавна секция
+"<title>Списък с адреси</title>\n" // заглавие
+"</head>\n"
+"<body>\n" // тяло
+"<h1>Благодарим, че се присъединихте към нашия списък с "
+"адреси</h1>\n" // параграф heading 1
+"Въведената от вас информация е:</p>\n" // стандартен параграф
+"<table cellspacing="5" cellpadding="5" border="1">\n" // таблица
+"<tr>\n" // ред
+"<td align="left">Собствено име:</td>\n" // елемент
+"<td>" + firstName + "</td>\n" // елемент
+"</tr>\n" // ред
+"<tr>\n" // ред
+"<td align="left">Фамилно име:</td>\n"
+"<td>" + lastName + "</td>\n"
+"</tr>\n" // ред
+"<tr>\n" // ред
+"<td align="left">Електронна поща:</td>\n"
+"<td>" + emailAddress + "</td>\n"
+"</tr>\n"
+"</table>\n"
```

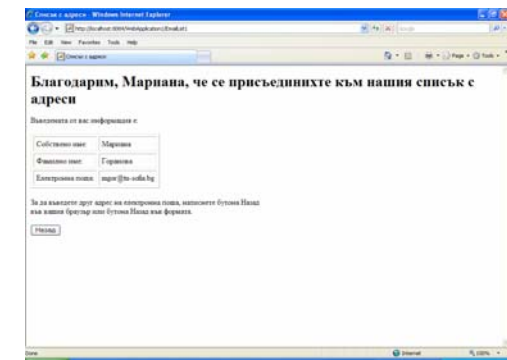
```
+"<p>За да въведете друг адрес на електронна поща, натиснете бутона "
+"Назад<br>\n"
+"във вашия браузър или бутона Назад във формата.</p>\n"
+"<form action="index.html" " // форма
+" method="post">\n"
+"<input type="submit" value="Назад">\n" // submit бутон
+"</form>\n"
+"</body>\n"
+"</html>\n");
out.close();
}
```

web.xml

```
<?xml version="1.0" encoding="windows-1251"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
<servlet>
<servlet-name>EmailList</servlet-name>
<servlet-class>web.EmailList</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EmailList</servlet-name>
<url-pattern>/EmailList</url-pattern>
</servlet-mapping>
</web-app>
```

Пример: Използване на други Java класове

Запис във файл на Списък с електронни адреси



User.java – дефинира потребител на приложението

```
package web;
public class User {
    private String firstName; // Потребител
    private String lastName; // собствено име
    private String emailAddress; // фамилно име
    // електронен адрес
    public User() {}
    public User(String first, String last, String email) {
        firstName = first;
        lastName = last;
        emailAddress = email;
    }
    // Методи за достъп
    public void setFirstName(String first) { firstName = first; }
    public String getFirstName() { return firstName; }
    public void setLastName(String last) { lastName = last; }
    public String getLastName() { return lastName; }
    public void setEmailAddress(String email) { emailAddress = email; }
    public String getEmailAddress() { return emailAddress; }
}
```

UserIO.java – записва данните на потребител в текстов файл

```
package web;
import java.io.*;
public class UserIO {
    public synchronized static void addRecord (User user, String filename)
        throws IOException {
        PrintWriter out = new PrintWriter(new FileWriter(filename, true));
        out.println(user.getEmailAddress() + "|"
            + user.getFirstName() + "|"
            + user.getLastName());
        out.close();
    }
}
```

Не позволява двама потребителя да пишат едновременно във файла

Отваря файла за добавяне на записи

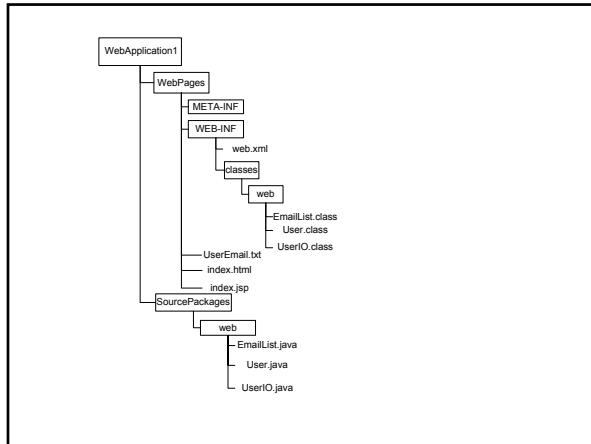
EmailList1.java

```
package web;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class EmailList1 extends HttpServlet {
    // Обработка HTTP POST заявка
    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        // Установява типа на съдържанието на отговора, който ще се изпрати
        // на клиента – html текст
        response.setContentType("text/html;charset=windows-1251");
        request.setCharacterEncoding(response.getCharacterEncoding());
        // Получава изходен поток за символни данни
        PrintWriter out = new PrintWriter(response.getOutputStream());
        // Получава параметрите, предадени от клиента към сървлета
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
        String emailAddress = request.getParameter("emailAddress");
    }
}
```

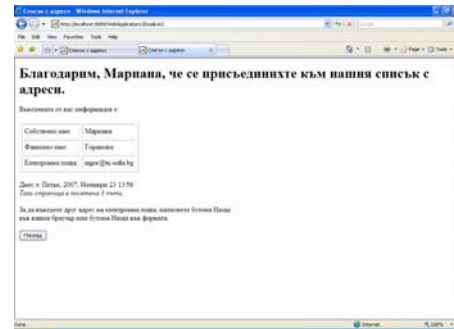
```
// Създава потребител
User user = new User(firstName, lastName, emailAddress);
// Получава пътя до файла
String fileName = getServletContext().getRealPath("UserEmail.txt");
// Добавя въведената информация от потребителя във файл
UserIO.addRecord(user, fileName);
out.println("<html>\n"
    + "<head>\n"
    + "<title>Списък с адреси</title>\n"
    + "</head>\n"
    + "<body>\n"
    + "<h1>Благодарим, "+user.getFirstName()
    + ", че се присъединихте към нашия списък с адреси</h1>\n"
    + "<p>Въведената от вас информация е:</p>\n"
    + "<table cellpadding='5' cellspacing='5' border='1'>\n"
    + "<tr><td align='right'>Собствено име:</td>\n"
    + "<td>" + user.getFirstName() + "</td>\n"
    + "</tr>\n"
    + "<tr><td align='right'>Фамилно име:</td>\n"
    + "<td>" + user.getLastName() + "</td>\n"
    + "</tr>\n"
```

```
+ " <tr><td align='right'>Електронна поща:</td>\n"
+ " <td>" + user.getEmailAddress() + "</td>\n"
+ " </tr>\n"
+ "</table>\n"
+ "<p>За да въведете друг адрес на електронна поща, "
+ "натиснете бутона Назад <br>\n"
+ "във вашия браузър или бутона Назад във формата.</p>\n"
+ "<form action='..index.html' "
+ "method='post'>\n"
+ " <input type='submit' value='Назад'>\n"
+ "</form>\n"
+ "</body>\n"
+ "</html>\n");
out.close();
}
```

UserEmail.txt
 mgor@tu-sofia.bg|Мариана|Горанова



Пример: Декларация на променлива на екземплярите



```
package web;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import java.text.*;

public class EmailList2 extends HttpServlet {
    private int counter;
    public void init() throws ServletException {
        counter = 0;
    }
    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        // Установява типа на съдържанието на отговора, който ще се изпрати
        // на клиента – html текст
        response.setContentType("text/html;charset=windows-1251");
        request.setCharacterEncoding(response.getCharacterEncoding());
        // Получава изходен поток за символни данни
        PrintWriter out = response.getWriter();
        // Получава параметрите, предадени от клиента към сървета
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
        String emailAddress = request.getParameter("emailAddress");
    }
}
```

Променлива на екземплярите

```
// Създава потребител
User user = new User(firstName, lastName, emailAddress);
// Получава пътя до файла
String fileName = getServletContext().getRealPath("UserEmail.txt");
// Добавя въведената информация от потребителя във файл
UserIO.addRecord(user, fileName);

int i = 0;
synchronized (this) {
    counter++;
    i = counter;
}

Locale locale = request.getLocale();
DateFormat full = DateFormat.getDateInstance(DateFormat.FULL,
    DateFormat.SHORT,
    locale);
String date = full.format(new Date());
```

Не позволява две нишки да модифицират едновременно променливата на екземплярите

```
// Запис на текст в html страницата
out.println("<html>\n" // HTML документ
    +"<head>\n" // заглавна секция
    +"<title>Списък с адреси</title>\n" // заглавие
    +"</head>\n"
    +"<body>\n" // тяло
    +"<h1>Благодарим, " + user.getFirstName()
    +" , че се присъединихте към нашия списък с адреси.</h1>\n"
    +"<p>Въведената от вас информация е:</p>\n"
    +"<table cellspacing='5' cellpadding='5' border='1'>\n"
    +"<tr>\n" // ред
    +"<td align='left'>Собствено име: </td>\n" // елемент
    +"<td>" + user.getFirstName() + "</td>\n" // елемент
    +"</tr>\n"
    +"<tr>\n" // ред
    +"<td align='left'>Фамилно име: </td>\n"
    +"<td>" + user.getLastName() + "</td>\n"
    +"</tr>\n"
    +"<tr>\n" // ред
    +"<td align='left'>Електронна поща: </td>\n"
    +"<td>" + user.getEmailAddress() + "</td>\n"
    +"</tr>\n"
    +"</table>\n"
```

```
+"<p>Днес е: " + date + "<br>\n"
    +"</>Тази страница е посетена " + i + " пъти.</></p>\n"
    +"<p>За да въведете друг адрес на електронна поща, "
    +"<br>натиснете бутона Назад <br>\n"
    +"<p>във вашия браузър или бутона Назад във формата.</p>\n"
    +"<form action='index.html' " // форма
    +"<br>method='post'>\n"
    +"<input type='submit' value='Назад'>\n" // submit бутон
    +"</form>\n"
    +"</body>\n"
    +"</html>\n";
```

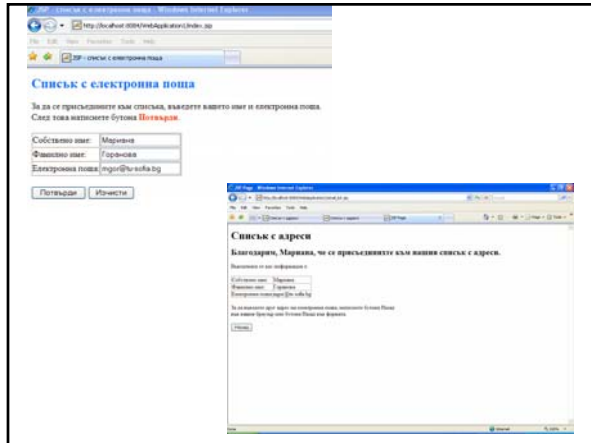
JSP (JavaServer Pages) технология

1. JSP – смесване на статично HTML съдържание и динамично съдържание, генерирано от сървлети

- при подаване на заявка от потребител JSP машината транслира JSP в сървлет и го компилира;
- сървлетът се изпълнява от сървлет машината.

Пример: Списък от електронни адреси

Включва HTML страница, намираща се на сървър, която съдържа текстови полета за въвеждане на имена и електронен адрес, и два бутона Submit и Reset. Потребител интерпретира тази страница на своя браузър. При натискане на бутона Submit HTML страницата извиква JSP, предава въведените данни от потребителя и връща HTML страница, която изобразява въведената информация от потребителя и бутон Return за повторно въвеждане на информация. При натискане на бутона Reset се изчистват текстовите полета.



index.jsp

```
<%@ page contentType="text/html;charset=windows-1251" %>
<%@page pageEncoding="windows-1251"%>
<%--
The taglib directive below imports the JSTL library. If you uncomment it,
you must also add the JSTL library to the project. The Add Library... action
on Libraries node in Projects view can be used to add the JSTL 1.1 library.
--%>
<%--
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
--%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
  charset=windows-1251">
  <title>JSP - списък с електронна поща</title>
</head>
<body>
<h2><font color=#0066FF>Списък с електронна поща</font></h2>
За да се присъедините към списъка, въведете вашето име и
електронна поща.<br>След това натиснете бутона <b>
<font color=#FF3300>Потвърди</font></b>.<br>
<form method="post" action="email_list.jsp">
  <table cellspacing="5" cellpadding="5" border="1">
    <tr><td align="left">Собствено име:</td>
      <td><input type="text" name="firstName"></td></tr>
    <tr><td align="left">Фамилно име:</td>
      <td><input type="text" name="lastName"></td></tr>
    <tr><td align="left">Електронна поща:</td>
      <td><input type="text" name="emailAddress"></td></tr>
  </table>
```

```
<p>
  <input type = "submit" value="Потвърди">
  <input type = "reset" value="Изчисти" >
</p>
</form>
</body>
</html>
```

```

email_list.jsp
<%@ page contentType="text/html;charset=windows-1251" %>
<%@page pageEncoding="windows-1251"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>JSP Page</title>
</head>
<body>
<h1>Списък с адреси</h1>
<%
response.setContentType("text/html;charset=windows-1251");
request.setCharacterEncoding(response.getCharacterEncoding());
String firstName = request.getParameter("firstName");
String lastName = request.getParameter("lastName");
String emailAddress = request.getParameter("emailAddress");
%>

```

Неявен обект request
JSP скриплет

```

<h2>Благодарим, <%= firstName %>, че се присъединихте към нашия
списък с адреси.</h2>
<p>Въведената от вас информация е:</p>
<table cellspacing="5" cellpadding="5" border="1">
<tr>
<td align="left">Собствено име: </td>
<td><%= firstName %></td>
</tr>
<tr>
<td align="left">Фамилно име: </td>
<td><%= lastName %></td>
</tr>
<tr>
<td align="left">Електронна поща: </td>
<td><%= emailAddress %></td>
</tr>
</table>

```

JSP израз

```

<p>За да въведете друг адрес на електронна поща,
натиснете бутона Назад <br>
във вашия браузър или бутона Назад във формата.</p>
<form action="index.jsp" method="post">
<input type="submit" value="Назад">
</form>
</body>
</html>

```

JSP скриплет
<% оператори на Java %>

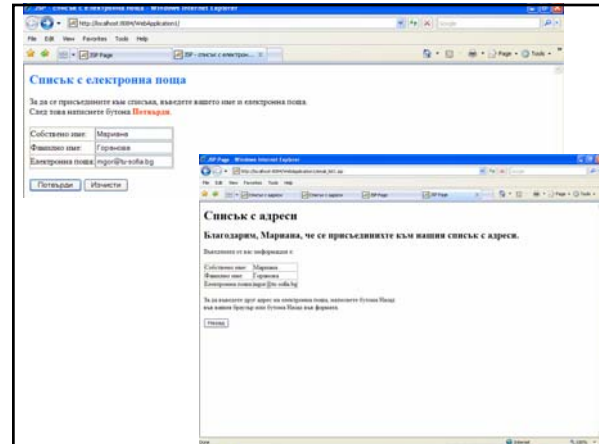
JSP израз
<%= израз на Java, който може да се конвертира в низ %>

Получаване на параметър от заявка чрез неявен обект request
request.getParameter(<име_на_параметър>);

Примери:
<%
String firstName = request.getParameter("firstName");
%>
First name: <%= firstName %>

или
First name: <%= request.getParameter("firstName") %>

Пример: Използване на други Java класове с JSP
Запис във файл на Списък от електронни адреси




```
email_list1.jsp

<%@ page contentType="text/html;charset=windows-1251" %>
<%@page pageEncoding="windows-1251"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>JSP Page</title>
</head>
<body>
```

```
<h1>Списък с адреси</h1>
<%@ page import = "web.*" %>
<%
response.setContentType("text/html;charset=windows-1251");
request.setCharacterEncoding(response.getCharacterEncoding());
String firstName = request.getParameter("firstName");
String lastName = request.getParameter("lastName");
String emailAddress = request.getParameter("emailAddress");
User user = new User(firstName, lastName, emailAddress);
String filename = getServletContext().getRealPath("UserEmail.txt");
UserIO.addRecord (user, filename);
%>
<h2>Благодарим, <%= user.getFirstName() %>, че се присъединихте
към нашия списък с адреси.</h2>
```

Директива за импортиране на Java класове

```
<p>Въведената от вас информация е:</p>
<table cellspacing="5" cellpadding="5" border="1">
<tr>
<td align="left">Собствено име: </td>
<td><%= user.getFirstName() %></td>
</tr>
<tr>
<td align="left">Фамилно име: </td>
<td><%= user.getLastName() %></td>
</tr>
<tr>
<td align="left">Електронна поща: </td>
<td><%= user.getEmailAddress() %></td>
</tr>
</table>
<p>За да въведете друг адрес на електронна поща, натиснете бутона
Назад <br>във вашия браузър или бутона Назад във формата.</p>
<form action="index.jsp" method="post">
<input type="submit" value="Назад">
</form>
</body>
</html>
```

JSP директива за импортиране на Java класове

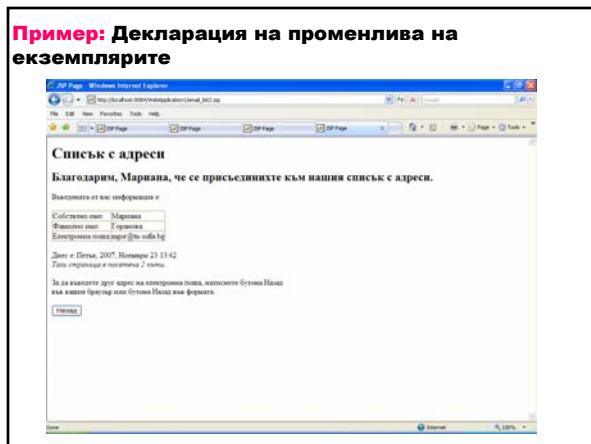
```
<%@ page import="<пакет>.*, ..." %>
```

JSP коментар

```
<%-- коментар --%>
```

JSP декларация на променливи и методи на екземплярите

```
<%! %>
```



```
email_list2.jsp

<%@ page contentType="text/html;charset=windows-1251" %>
<%@page pageEncoding="windows-1251"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<title>JSP Page</title>
</head>
<body>
```

```
<h1>Списък с адреси</h1>
<%@ page import = "web.*, java.util.*, java.text.*" %>
<%! int counter = 0; %>
<%
    response.setContentType("text/html;charset=windows-1251");
    request.setCharacterEncoding(response.getCharacterEncoding());
    String firstName = request.getParameter("firstName");
    String lastName = request.getParameter("lastName");
    String emailAddress = request.getParameter("emailAddress");
    User user = new User(firstName, lastName, emailAddress);
    String filename = getServletContext().getRealPath("UserEmail.txt");
    UserIO.addRecord (user, filename);
    int i = 0;
    synchronized (this) {
        counter++;
        i = counter;
    }
%>
<h2>Благодарим, <%= user.getFirstName() %>, че се присъединихте
към нашия списък с адреси.</h2>
```

```
<p>Въведената от вас информация е:</p>
<table cellspacing="5" cellpadding="5" border="1">
<tr>
<td align="left">Собствено име: </td>
<td><%= user.getFirstName() %></td>
</tr>
<tr>
<td align="left">Фамилно име: </td>
<td><%= user.getLastName() %></td>
</tr>
<tr>
<td align="left">Електронна поща: </td>
<td><%= user.getEmailAddress() %></td>
</tr>
</table>
<%
    Locale locale = request.getLocale();
    DateFormat full =
        DateFormat.getDateInstance(DateFormat.FULL,
            DateFormat.SHORT, locale);
    String date = full.format(new Date());
%>
```

```
<p>Днес е: <%= date %> <br>
<i><%= i %> пъти.</i>
</p>
<p>За да въведете друг адрес на електронна поща, натиснете
бутона Назад <br>
във вашия браузър или бутона Назад във формата.</p>
<form action="index.jsp" method="post">
<input type="submit" value="Назад">
</form>
</body>
</html>
```