

```

        width="250"
        height="250">
<jsp:params>
  <jsp:param name="bNumber" value="7" />
  <jsp:param name="state" value="true" />
</jsp:params>
<jsp:fallback>
  <p> unable to start plugin </p>
</jsp:fallback>
</jsp:plugin>

```

Код апплета находится в примере 5 главы 11, и пакет, в котором он объявлен, должен быть расположен в корне папки **/WEB-INF**, а не в папке **/classes**.

Элементы **<jsp:attribute>**, **<jsp:body>**, **<jsp:invoke>**, **<jsp:doBody>**, **<jsp:element>**, **<jsp:output>** используются в основном при включении в страницу пользовательских тегов.

JSP-документ

Предпочтительно создавать JSP-страницу в виде JSP-документа – корректного XML-документа, который ссылается на определенное пространство имен, содержит стандартные действия JSP, пользовательские теги и теги ядра JSTL, XML-эквиваленты директив JSP. В JSP-документе вышеперечисленные пять тегов неприменимы, поэтому их нужно заменять стандартными действиями и корректными тегами. JSP-документы необходимо сохранять с расширением **.jspx**.

Директива **taglib** для обычной JSP:

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>

```

для JSP-документа:

```

<jsp:root xmlns:c="http://java.sun.com/jsp/jstl/core"/>

```

Директива **page** для обычной JSP:

```

<%@ page contentType="text/html"%>

```

для JSP-документа:

```

<jsp:directive.page contentType="text/html" />

```

Директива **include** для обычной JSP:

```

<%@ include file="file.jspf"%>

```

для JSP-документа:

```

<jsp:directive.include file="file.jspf" />

```

Ниже приведены два примера, демонстрирующие различие применения стандартных действий и тегов при создании JSP-страниц и JSP-документов.

```

<!--пример #1 : обычная jsp-страница: page.jsp -->
<%@ page contentType="text/html; charset=Cp1251" %>
<html><head><title>JSP-страница</title></head>
<%! private int count = 0;
    String version = new String("J2EE 1.5");
    private String getName(){return "J2EE 1.6";} %>

```

```

<% out.println("Значение count: "); %>
<%= count++ %>
<br/>
<% out.println("Значение count после инкремента: " +
count); %>
<br/>
<% out.println("Старое значение version: "); %>
<%= version %>
<br/>
<% version=getName();
out.println("Новое значение version: " + version); %>
</html>

```

Версия в виде JSP-документа несколько более громоздка, но читать и искать ошибки в таком документе проще, нежели в предыдущем.

```

<!--пример # 2 : правильный jsp-документ : page.jspx -->
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
                                version="2.0">

    <jsp:directive.page contentType=
                                "text/html; charset=UTF-8" />

    <html><body>
    <jsp:declaration>
        private int count = 0;
        String version = new String("J2EE 1.5");
        private String getName(){return "J2EE 1.6";}
    </jsp:declaration>
    <jsp:scriptlet>
        out.println("Значение count: ");
    </jsp:scriptlet>
    <jsp:expression>
        count++
    </jsp:expression>
    <br />
    <jsp:scriptlet>
        out.println("Значение count после инкремента:"
                                + count);

    </jsp:scriptlet>
    <br/>
    <jsp:scriptlet>
        out.println("Старое значение version: ");
    </jsp:scriptlet>
    <jsp:expression> version </jsp:expression>
    <br/>
    <jsp:scriptlet> version=getName();
        out.println("Новое значение version: " + version);
    </jsp:scriptlet>
    </body></html>
</jsp:root>

```

Далее в главе примеры будут приведены в виде JSP-документов.