

Действия, определяемые программистом (JSP Custom Actions)



Что такое Custom Actions?

- Механизм расширения языка JSP **новыми действиями**
- Действия выполняются **на сервере** при обработке запроса
- Действия распространяются в виде «**библиотек действий**» (tag libraries)
 - > JSTL
- Функции custom actions:
 - > генерация фрагмента ответа
 - > генерация отчета в JasperReports
 - > вывод диаграммы в JFreeChart
 - > обращение к БД для получения информации
 - > работа с переменными, управление потоком выполнения, циклы, ...
 - > Главное - **возможность повторного использования** функций



Возможности действий, определяемых программистом

- Передача атрибутов с вызывающей страницы
- Передача переменных обратно на вызывающую страницу
- Доступ ко всем объектам, доступным на JSP-страницах
- Взаимодействие между действиями
 - Можно в одном действии инициализировать JavaBeans-компонент и создать для него EL-переменную, а затем использовать ее в другом действии
- Вложение действий и взаимодействие через закрытые переменные
- Вычисление или другая обработка тела действия

Составляющие архитектуры Custom Actions

- Обработчик действия (Tag Handler)
 - > Определяет поведение действия
 - > Варианты реализации:
 - > Классический обработчик
 - > Простой обработчик (JSP 2.0)
 - > Тэг-файл (JSP 2.0)
- Дескриптор библиотеки действий (Tag Library Descriptor — TLD)
 - > Устанавливает соответствие между элементами и обработчиками
- JSP-страница
 - > Использует действия

Custom Actions на JSP-странице

- Используемые библиотеки действий объявляются директивой **taglib**
 - > `<%@ taglib prefix="myprefix" uri="myuri" %>`
 - > **prefix**: сокращенное обозначение библиотеки
 - > **uri**: указывает на дескриптор библиотеки (TLD)
- Действия вызываются следующим образом:

`<prefix:tag attr1="value" ... attrN="value" />`

или

`<prefix:tag attr1="value" ... attrN="value" >`

Тело действия

`</prefix:tag>`



Что происходит при компиляции JSP-страницы?

- Транслятор JSP-страниц находит TLD-файл с помощью атрибута `uri` директивы `taglib`
 - TLD-файл определяет соответствие между действием (`prefix:tag`) и его обработчиком
- При вызове действия на JSP-странице в PISC генерируется код вызова обработчика действия
 - Код зависит от типа обработчика
 - Последовательность вызовов методов обработчика рассматривается далее

Объявление библиотеки действий: атрибут uri

- Путь к TLD-файлу
 - `<%@ taglib prefix="tlt" uri="/WEB-INF/iterator.tld"%>`
- Косвенная ссылка на TLD-файл по логическому имени
 - `<%@ taglib prefix="tlt" uri="/tlt"%>`
 - Отображение логического имени на путь к TLD-файлу определяется в `web.xml`

```
<jsp-config>  
  <taglib>  
    <taglib-uri>/tlt</taglib-uri>  
    <taglib-location>/WEB-INF/iterator.tld</taglib-location>  
  </taglib>  
</jsp-config>
```



Объявление библиотеки действий: атрибут uri

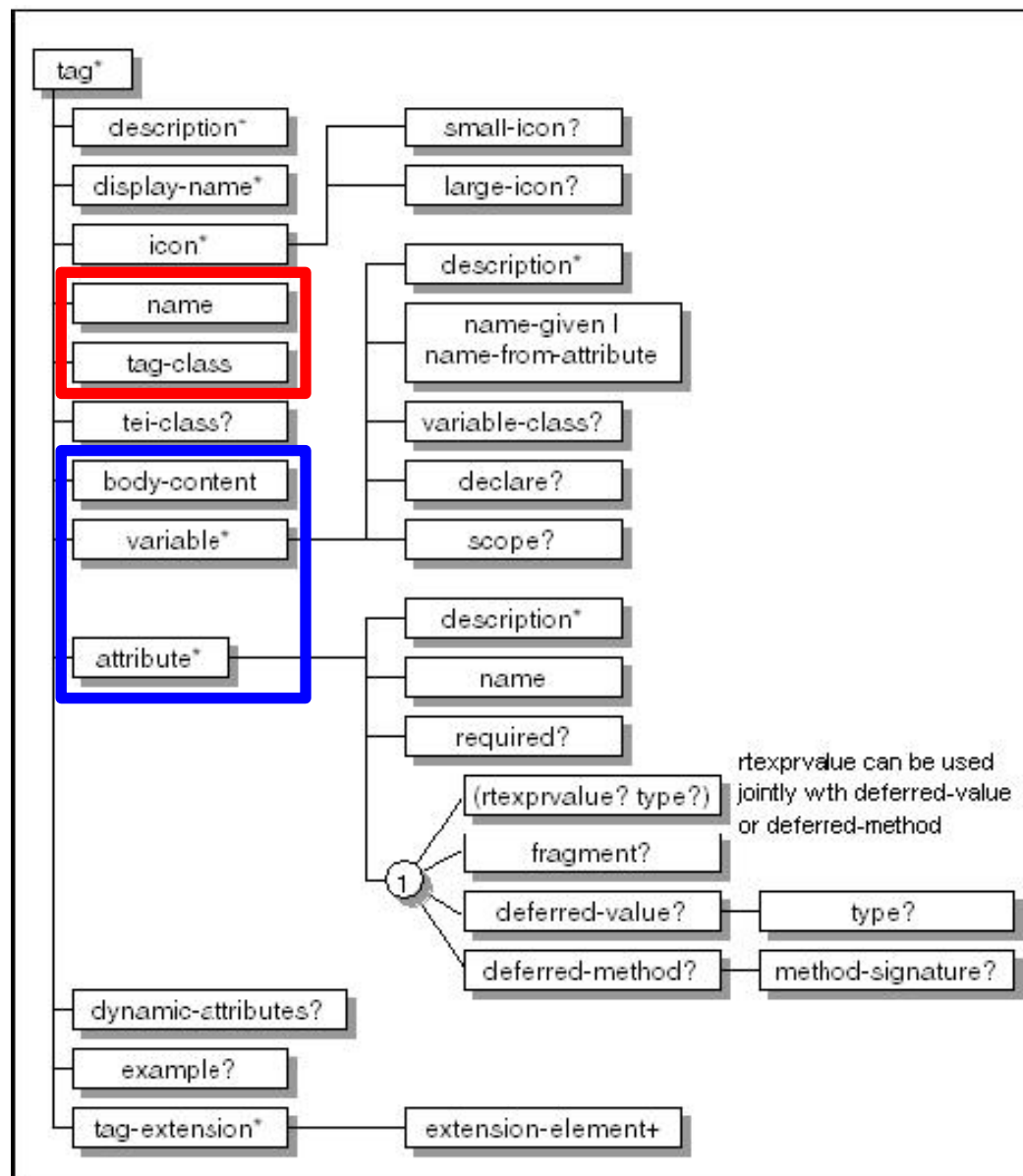
- Абсолютный URI — примеры для JSTL
 - > `<%@ taglib prefix="core" uri="http://java.sun.com/jsp/jstl/core"%>`
 - > `<%@ taglib prefix="xml" uri="http://java.sun.com/jsp/jstl/xml"%>`
 - > `<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>`
 - > `<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>`
 - > `<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>`
- URI библиотеки определяется в TLD-файле



Tag Library Descriptor (TLD)

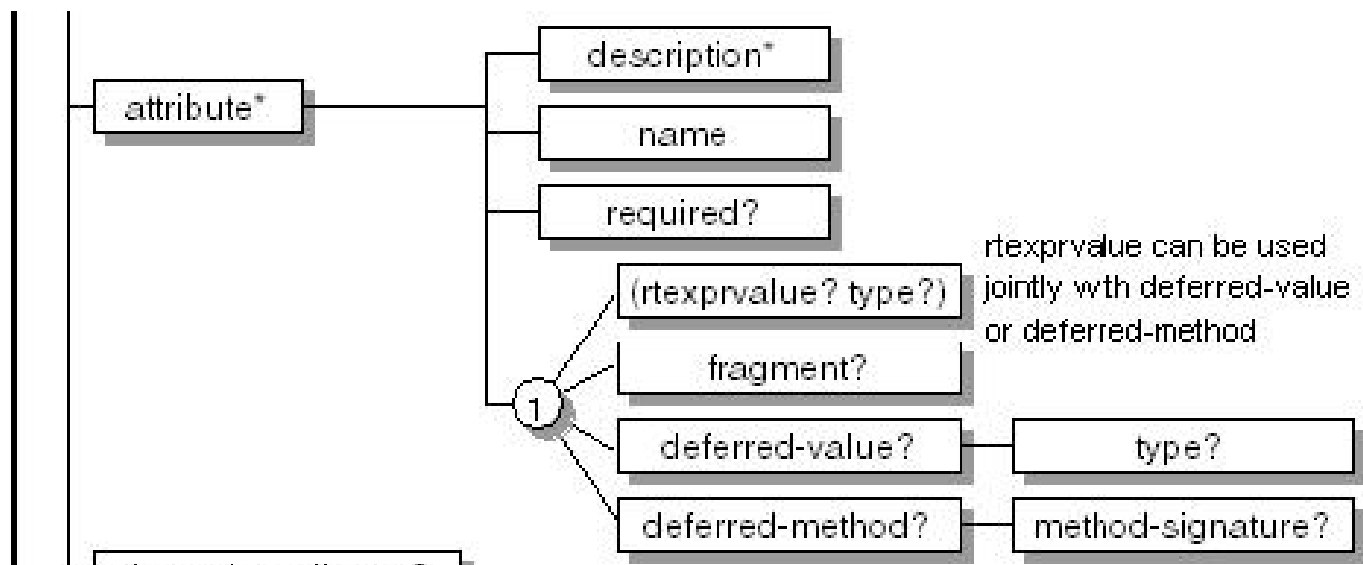
- XML-файл, который описывает
 - Имя действия
 - Тело действия
 - Атрибуты
 - Класс обработчика действия
- Размещается
 - В каталоге WEB-INF, либо
 - В файле META-INF/taglib.tld в JAR-архиве в каталоге WEB-INF/lib
- Размещение указывается в JSP-файле
 - Атрибут **uri** директивы **taglib**

Описание действия в TLD



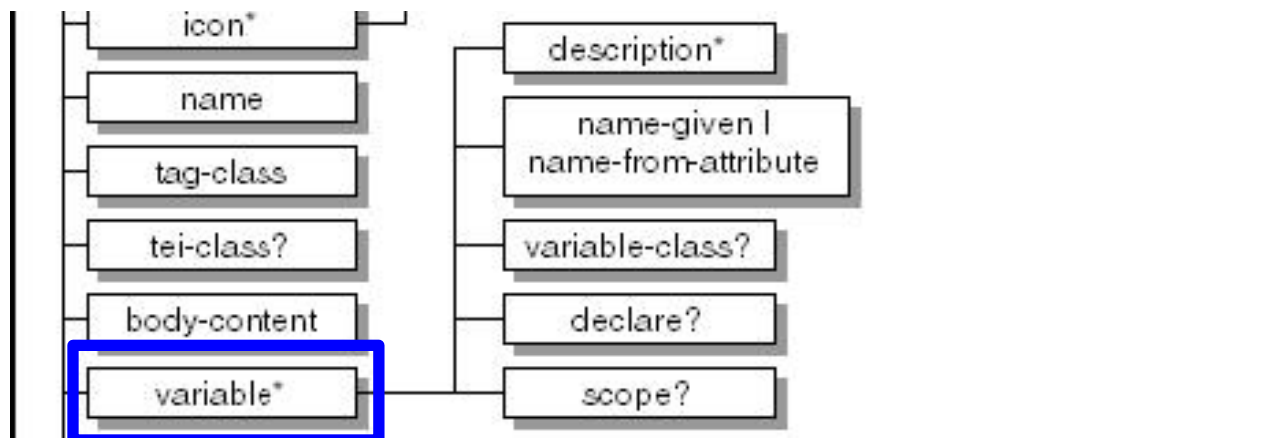
- Атрибуты
 - > от JSP-страницы к обработчику действия
- Переменные
 - > от обработчика действия к JSP-странице
- Тело действия
 - > вычисляется обработчиком или РС-ом
 - > ВОЗМОЖНО, в цикле

Описание атрибутов действия в TLD



- **rtexprvalue**: если **true**, то значение типа **type** может вычисляться при обработке запроса
- **fragment**: значение атрибута — фрагмент JSP-кода
- **deferred-value**: отложенное EL-выражение типа **type**, которое может вычисляться по мере надобности

Описание переменных действия в TLD

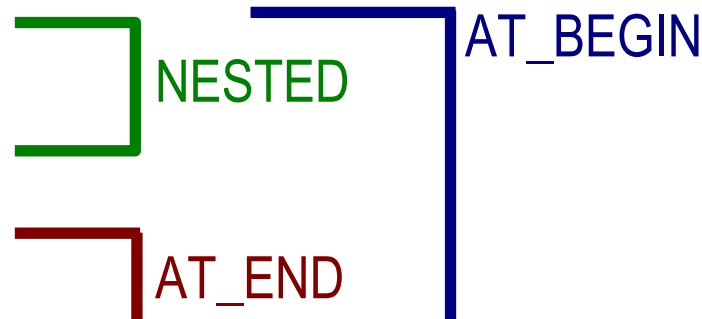


- Имя переменной фиксированное (**name-given**) или из атрибута действия (**name-from-attribute**)
- Область видимости переменной (**scope**):

`<prefix:tag attr1="value" ... attrN="value" >`

Тело действия

`</prefix:tag>`



...



Описание тела действия в TLD

- Элемент `<bodycontent>`
 - > **JSP**: тело интерпретируется транслятором как обычный JSP-фрагмент
 - > Вычисляются выражения и выполняются действия, вложенные в тело
 - > **empty**: тело действия должно быть пустым
 - > **scriptless**: в теле действия запрещено использовать скриптовые элементы
 - > **tagdependent**: тело интерпретируется на этапе выполнения обработчиком действия



Пример TLD

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

```
<taglib xmlns="http://java.sun.com/xml/ns/javaee"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
    http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd"  
  version="2.1">
```

```
<description>JSTL 1.2 core library</description>  
<display-name>JSTL core</display-name>  
<tlib-version>1.2</tlib-version>  
<short-name>c</short-name>  
<uri>http://java.sun.com/jsp/jstl/core</uri>
```



Пример TLD

```
<tag>
  <description>
    Subtag of &lt;choose> that includes its body if its
    condition evalutes to 'true'
  </description>
  <name>when</name>
  <tag-class>org.apache.taglibs.standard.tag.rt.core.WhenTag</tag-class>
  <body-content>JSP</body-content>
  <attribute>
    <description>The test condition that determines whether or not the body content
    should be processed.
    </description>
    <name>test</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
    <type>boolean</type>
  </attribute>
</tag>
```

```
...
</taglib>
```

Что такое библиотека действий (Tag Library)?

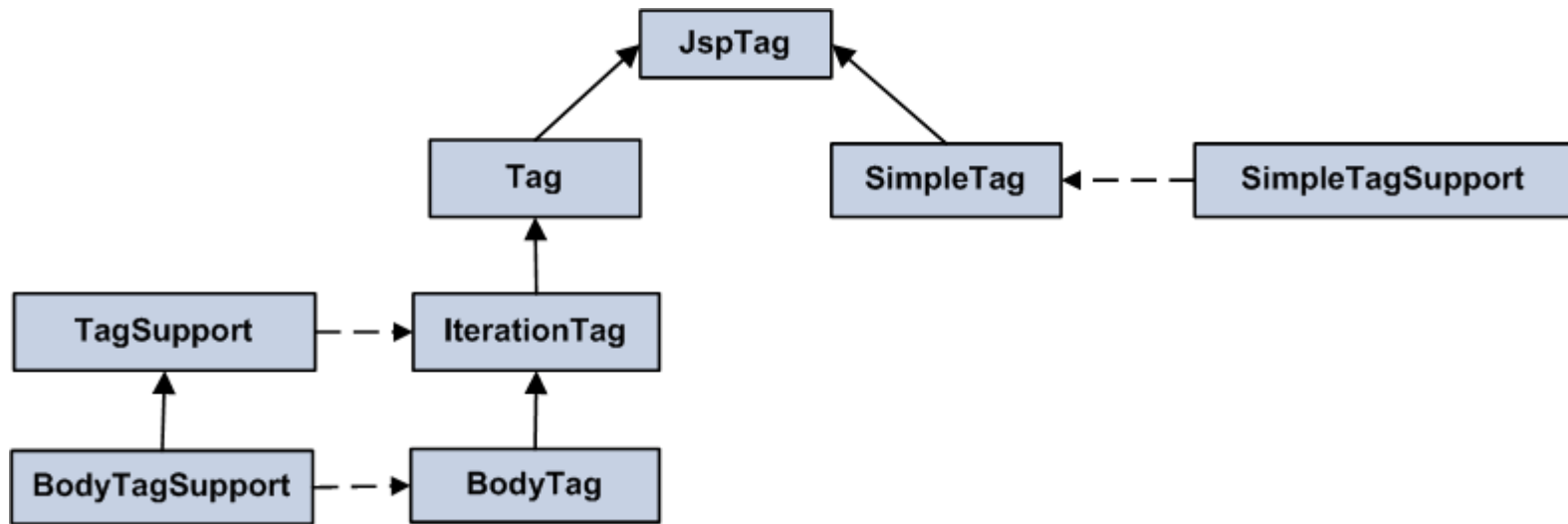
- Набор взаимосвязанных действий
 - > Действия собираются в **библиотеку действий**
- Обычно упаковывается в JAR-файл, содержащий
 - > Дескриптор библиотеки действий (TLD), например, META-INF/taglib.tld
 - > *.class-файлы для классов обработчиков действий
 - > Любые дополнительные ресурсы



Настройка библиотеки действий в отдельном веб-приложении

- В неупакованном виде
 - > Классы обработчиков действий размещаются в каталоге [/WEB-INF/classes/](#)
 - > TLD-файл в каталоге [/WEB-INF/](#)
- В упакованном виде (JAR-файл)
 - > JAR-файл в каталоге [/WEB-INF/lib/](#)

Классические и простые обработчики



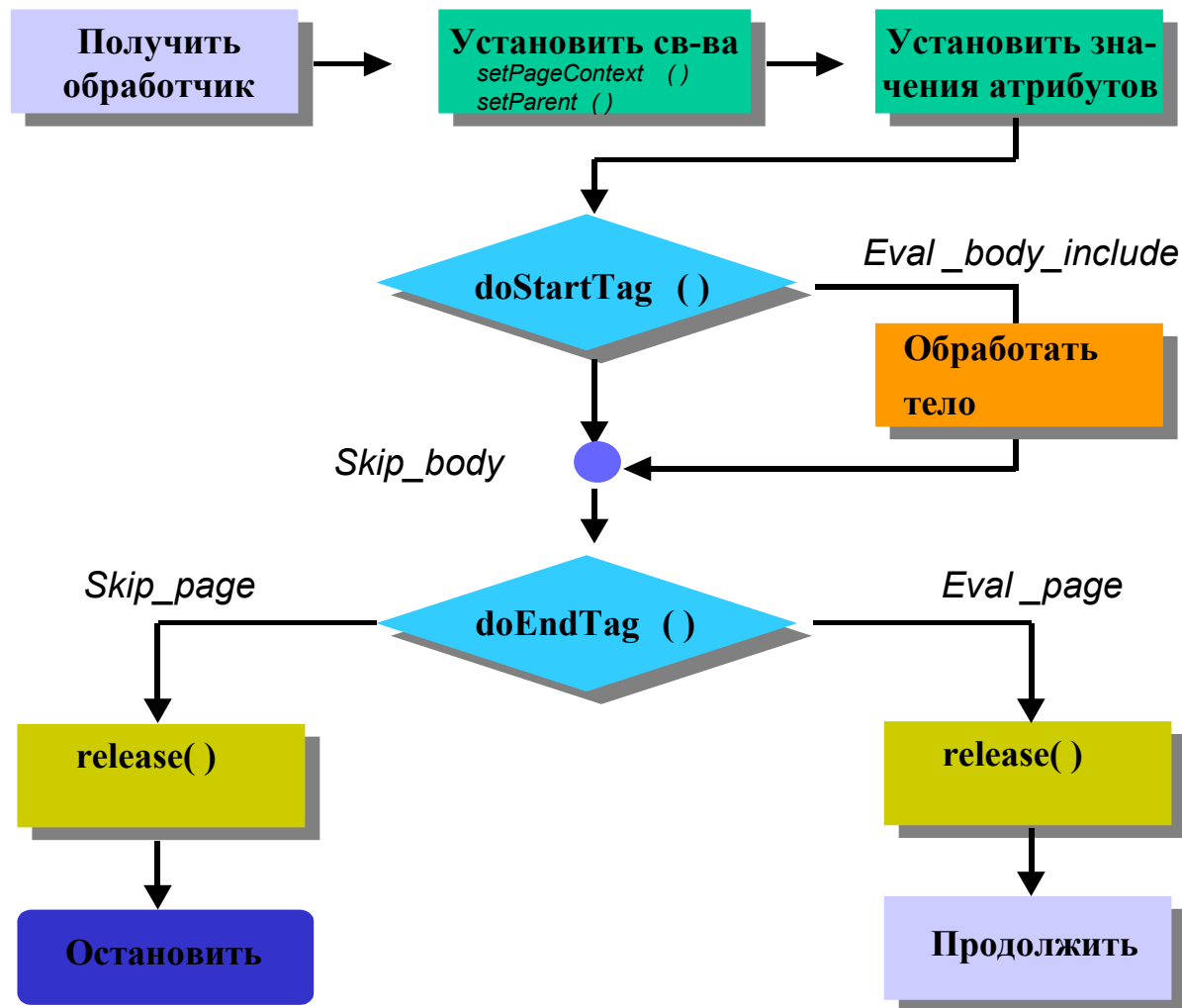
- Классические

- > **3 вида** действий: простые, циклические и обрабатывающие тело
- > **5 методов**, вызываемых для открывающего тега, тела и закрывающего тега

- Простые

- > **Единственный вид**
- > **Единственный метод** doTag() - после закрывающего тега
- > Начиная с JSP 2.0

Алгоритм вызова классического обработчика простого действия



Как вызывается классический обработчик действия, обрабатывающий тело?

```
<prefix:actionName  
  attr1="value1"  
  attr2="value2"  
>
```

The body

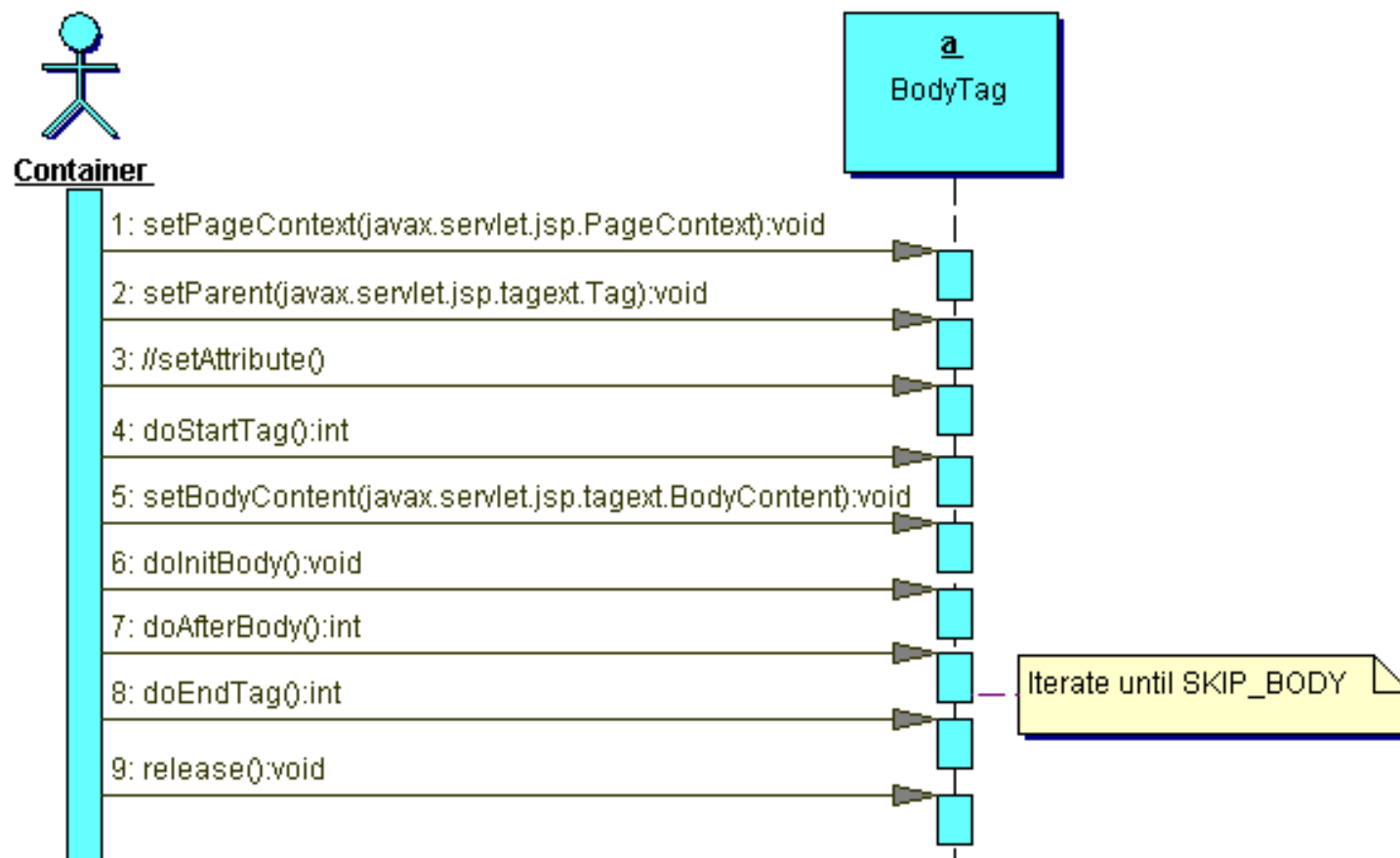
```
</prefix:actionName>
```

setPageContext ()

setParent ()

- setAttr1("values1")
- setAttr2("value2")
- doStartTag ()
- setBodyContent()
- doInitBody ()
- doAfterBody()
- doEndTag ()

Как вызывается классический обработчик действия, обрабатывающий тело?



Как вызывается простой обработчик действия?

```
// 1 действие на странице = 1 объект  
// класса обработчика  
ATag t = new ATag();  
t.setJSPContext(...);  
t.setParent(...);  
t.setAttribute1(value1);  
t.setAttribute2(value2);  
...  
t.setJspBody(new JspFragment(...))  
t.doTag();
```

Интерфейс JspFragment

- Представляет **фрагмент JSP-кода** в виде объекта, который можно вызывать столько раз, сколько нужно
 - > **Нельзя использовать скриптовые элементы**
- Способы определения JSP-фрагмента:
 - > **Тело действия**, для которого определен **простой** обработчик
 - > SimpleTag.getJspBody()
 - > **Тело** стандартного **действия** `<jsp:attribute>`, задающего значение атрибута вида **fragment**
- Вызов JSP-фрагмента
 - > `invoke(StringWriter writer)`
 - > `invoke(null)`, чтобы просто вывести вычисленный фрагмент

Тэг-файл

- Исходный файл, содержащий фрагмент JSP-кода
- Позволяет создать действие, определяемое программистом, с использованием **синтаксиса JSP**
 - > Авторы JSP-страниц, не владеющие Java, могут создавать библиотеки действий
 - > Упрощение разработки действий для вывода данных
- Автоматически транслируется в исходный код обработчика действия и компилируется
- TLD-файл не требуется
 - > Действие определяется директивами в самом тэг-файле

Тэг-файлы

- Простая и гибкая упаковка
 - > Достаточно поместить файл с расширением **.tag** в каталог **/WEB-INF/tags/** или какой-либо его **подкаталог**
 - > Для **/WEB-INF/tags/** и каждого его подкаталога автоматически генерируется отдельная библиотека действий
 - > Для дополнительной гибкости можно создать TLD
 - > Либо упаковать тэг-файл в JAR-архив в каталог **/META-INF/tags/** или его **подкаталог**
 - > TLD в этом случае **обязателен**

Тэг-файлы

- Альтернатива включению файлов с помощью `<%@include%>` и `<jsp:include>`
 - > Вызов действия компактнее, чем сочетание `<jsp:include>` и `<jsp:param>`
 - > Возможность вложения действий
 - > В теле действия нельзя использовать скриптовые элементы
 - > Обязательное объявление параметров тэг-файла (атрибутов)
 - > Переменные действия автоматически экспортируются и синхронизируются с переменными JSP-страницы



Использование библиотеки действий на базе тэг-файлов

- `<%@ taglib prefix="tt" tagdir="/WEB-INF/tags/dir" %>`
 - > Нужно указать размещение тэг-файлов с помощью атрибута `tagdir` директивы `taglib`
 - > Значение должно начинаться с `/WEB-INF/tags/`

```
<%@ taglib prefix="my"  
      tagdir="/WEB-INF/tags/" %>  
<my:repeat num="3">  
  tag body  
</my:repeat>
```

Директивы, используемые в тэг-файлах

- **taglib**
- **include**
- **tag**
 - > Аналогична директиве page для JSP-страниц, но предназначена для тэг-файлов
- **attribute**
 - > Описание атрибутов действия
- **variable**
 - > Описание **EL**-переменных действия, возвращаемых на вызывающую страницу

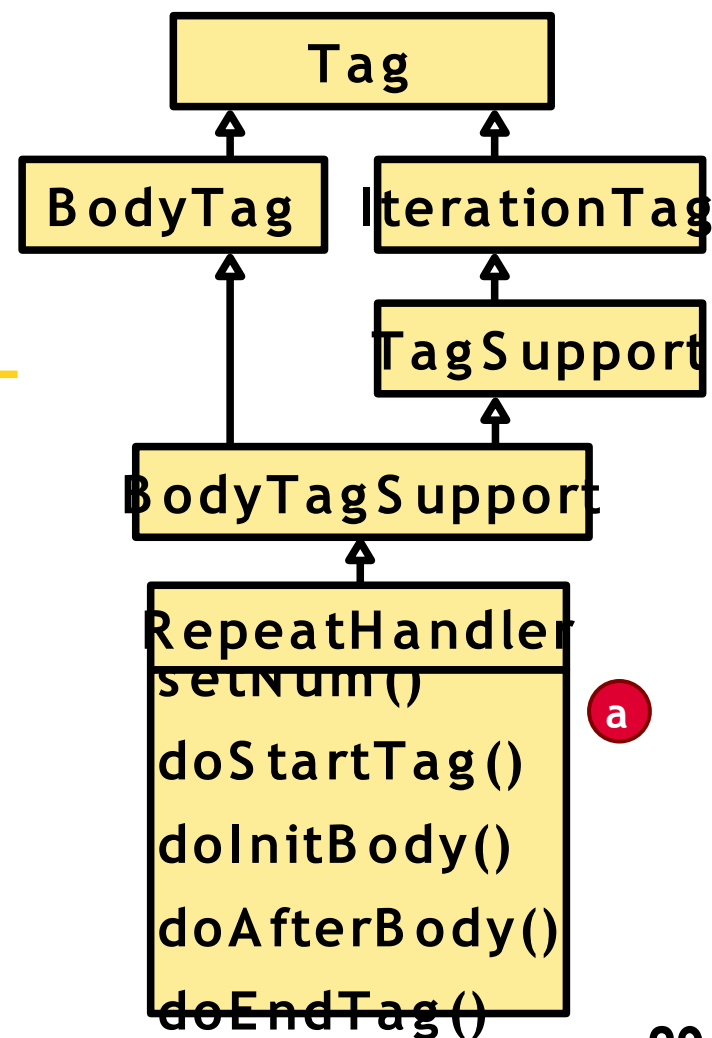
Циклическое действие: классический обработчик

Использование

```
<%@ taglib prefix="my"  
      uri="/mytags" %>  
<my:repeat num="3">  
  tag body  
</my:repeat>
```

Обработчик

```
int doStartTag() {  
    this.count = this.num;  
    return Tag.EVAL_BODY_INCLUDE;  
}  
  
int doAfterBody() {  
    this.count--;  
    return (this.count > 0) ?  
        Tag.EVAL_BODY_AGAIN :  
        Tag.SKIP_BODY;  
}
```



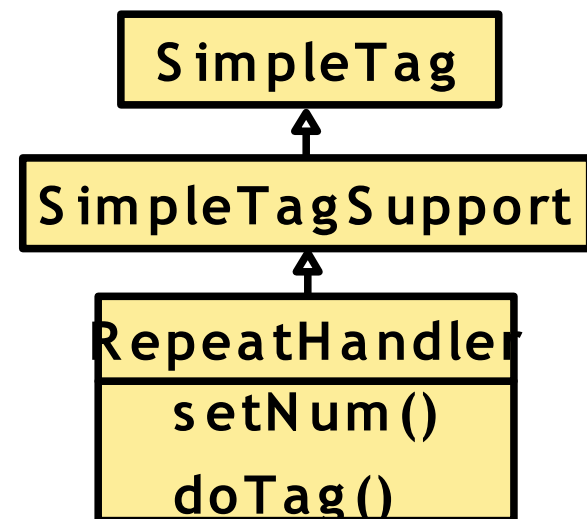
Циклическое действие: простой обработчик

Использование

```
<%@ taglib prefix="my"  
      uri="/mytags" %>  
<my:repeat num="3">  
  tag body  
</my:repeat>
```

Обработчик

```
void doTag() {  
    for( int i = 0; i < num; i++ ) {  
        getJspBody().invoke( null );  
    }  
}
```



Циклическое действие: обработчик в виде тэг-файла

Использование

```
<%@ taglib prefix="my"
      tagdir="/WEB-INF/tags/" %>
<my:repeat num="3">
  tag body
</my:repeat>
```

Обработчик

```
<%-- /WEB-INF/tags/repeat.tag --%>
<%@ attribute name="num" %>
<%@ taglib prefix="c"
      uri="http://java.sun.com/jsp/jstl/core" %>

<c:forEach begin="1" end="${num}">
  <jsp:doBody />
</c:forEach>
```



Пример тэг-файла с атрибутами и переменными (catalog.tag - 1)

<%-- Определение действия с помощью директив --%>

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

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

<%@ attribute name="bookDB" required="true" type="database.BookDB" %>

<%@ attribute name="color" required="true" %>

<%@ variable name-given="price" %>

<%@ variable name-given="salePrice" %>

<%@ attribute name="normalPrice" fragment="true" %>

<%@ attribute name="onSale" fragment="true" %>

Пример тэг-файла с атрибутами и переменными (catalog.tag - 2)

```
<table>
<c:forEach var="book" items="{bookDB.books}">
  <tr>
    <c:set var="bookId" value="{book.bookId}" />
    <td bgcolor="{color}">
      <c:url var="url" value="/bookdetails" >
        <c:param name="bookId" value="{bookId}" />
      </c:url>
      <a href="{url}"><strong>{book.title}&nbsp;</strong></a></td>
    <td bgcolor="{color}" rowspan=2>
      <c:set var="salePrice" value="{book.price * .85}" />
      <c:set var="price" value="{book.price}" />
    </td>
  </tr>
  <c:choose>
    <c:when test="{book.onSale}" >
      <jsp:invoke fragment="onSale" />
    </c:when>
    <c:otherwise>
      <jsp:invoke fragment="normalPrice" />
    </c:otherwise>
  </c:choose> &nbsp; </td>
</c:forEach>
</table>
```

Атрибуты,
переданные с
вызывающей
страницы

Переменные

Атрибуты с
JSP-фрагментами



Пример тэг-файла с атрибутами и переменными (bookcatalog.jsp - 3)

```
<%@ taglib prefix="sc" tagdir="/WEB-INF/tags" %>
```

```
...
```

```
<jsp:useBean id="bookDB" class="database.BookDB" scope="page" >  
  <jsp:setProperty name="bookDB" property="database" value="${bookDBAO}" />  
</jsp:useBean>
```

```
<sc:catalog bookDB="${bookDB}" color="#cccccc">  
  <jsp:attribute name="normalPrice">  
    <fmt:formatNumber value="${price}" type="currency"/>  
  </jsp:attribute>  
  <jsp:attribute name="onSale">  
    <strike><fmt:formatNumber value="${price}" type="currency"/></strike><br/>  
    <font color="red"><fmt:formatNumber value="${salePrice}"  
      type="currency"/></font>  
  </jsp:attribute>  
</sc:catalog>
```



Пример тэг-файла с атрибутами и переменными (результат вывода)

Please choose from our selections:

| | | |
|--|-------------------------------|------------------------------------|
| <u>Duke: A Biography of the Java Evangelist</u> | \$45.00 \$38.25 | <u>Add to Cart</u> |
| by Hazel Tu | | |
| <u>From Oak to Java: The Revolution of a Language</u> | \$10.75 | <u>Add to Cart</u> |
| by Kevin Novation | | |
| <u>Java Intermediate Bytecodes</u> | \$30.95 | <u>Add to Cart</u> |
| by James Gosling | | |
| <u>My Early Years: Growing up on "7"</u> | \$30.75 \$26.14 | <u>Add to Cart</u> |
| by Duke | | |
| <u>The Green Project: Programming for Consumer Devices</u> | \$30.00 | <u>Add to Cart</u> |
| by Ben Thrilled | | |
| <u>Web Components for Web Developers</u> | \$27.75 \$23.59 | <u>Add to Cart</u> |
| by Webster Masterson | | |
| <u>Web Servers for Fun and Profit</u> | \$40.75 | <u>Add to Cart</u> |
| by Jeeves | | |

Copyright © 2003 Sun Microsystems, Inc