

- **page** – ссылка **this** для текущего экземпляра данной страницы является объектом **java.lang.Object**. Область видимости в пределах страницы.
- **exception** – представляет собой исключение одного из подклассов класса **java.lang.Throwable**, которое передается странице сообщения об ошибках и доступно только на ней.

JSTL core

Библиотека тегов JSTL состоит из четырёх групп тегов: основные теги – **core**, теги форматирования – **formatting**, теги для работы с SQL – **sql**, теги для обработки XML – **xml**.

| Library | Actions | Description |
|-------------------|---------|--|
| core | 14 | <u>Основные</u> : if/then выражения и switch конструкции; вывод; создание и удаление контекстных переменных; управление свойствами JavaBeans компонентов; перехват исключений; итерирование коллекций; создание URL и импортирование их содержимого. |
| formatting | 12 | <u>Интернационализация и форматирование</u> : установка локализации; локализация текста и структуры сообщений; форматирование и анализ чисел, процентов, денег и дат. |
| sql | 6 | <u>Доступ к БД</u> : описание источника данных; выполнение запросов, обновление данных и транзакций; обработка результатов запроса. |
| xml | 10 | <u>XML-анализ и преобразование</u> : преобразование XML; доступ и преобразование XML через XPath и XSLT. |

Библиотека **core** содержит в себе наиболее часто используемые теги.

```
<%@taglib uri="http://java.sun.com/jstl/core" prefix="c" %>
– для обычной JSP.
<jsp:root version="1.2" xmlns:c=
"http://java.sun.com/jstl/core"> ...</jsp:root> – для XML
формата JSP.
```

Теги общего назначения:

```
<c:out /> – вычисляет и выводит значение выражения;
<c:set /> – устанавливает переменную в указанную область видимости;
<c:remove /> – удаляет переменную из указанной области видимости;
<c:catch /> – перехватывает обработку исключения.
```

Теги условного перехода:

```
<c:if /> – тело тега вычисляется только в том случае, если значение выраже-
ния true;
<c:choose /> (<c:when />, <c:otherwise />) – то же что и <c:if />
с поддержкой нескольких условий и действия, производимого по умолчанию.
```

Итераторы:

<c:forEach /> — выполняет тело тега для каждого элемента коллекции;

<c:forTokens /> — выполняет тело тега для каждой лексемы в строке.

Теги обработки URL:

<c:redirect /> — перенаправляет запрос на указанный **URL**;

<c:import /> — добавляет на JSP содержимое указанного WEB-ресурса;

<c:url /> — формирует адрес с учётом контекста приложения **request.getContextPath()**.

<c:param /> — добавляет параметр к запросу, сформированному при помощи **<c:url />**.

Ниже приведено несколько примеров, иллюстрирующих применение основных тегов из группы **core**.

<!--пример #5 : демонстрация работы тегов c:set, c:remove, c:if, c:out: core1.jspx -->

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
          xmlns:c="http://java.sun.com/jsp/jstl/core"
          version="2.0">
<jsp:directive.page contentType=
    "text/html; charset=Utf-8" />
<html><head><title>Демонстрация тегов core</title></head>
<h3> Демонстрация работы тегов c:set, c:remove, c:if, c:out
<br/></h3>
    <form>
Новое значение переменной:<input type="text" name="set" />
<br/>
Удалить переменную:<input type="checkbox" name="del" />
<br/>
<input type="submit" name="send" value="принять"/><br/>
    </form>
    <c:if test="${not empty param.send }">
        <c:if test="${not empty param.set }">
            <c:set var="item" value="${param.set}"
                scope="session"></c:set>
        </c:if>
        <c:if test="${not empty param.del }">
            <c:remove var="item"/>
        </c:if>
    </c:if>
    <c:if test="${not empty item }">
        <jsp:text>Значение переменной :</jsp:text>
        <c:out value="${item }"/>
    </c:if>
    <c:if test="${empty item and not empty param.send}">
        <jsp:text>Значение переменной: </jsp:text>
        <c:out value="пусто"/>
    </c:if>
</jsp:root>
```

```

</c:if>
</html>
</jsp:root>
<!--пример # 6 : демонстрация работы тегов c:forEach, c:choose, c:when,
c:otherwise : core2.jspx -->
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
          xmlns:c="http://java.sun.com/jsp/jstl/core"
          version="2.0">
<jsp:directive.page contentType=
    "text/html; charset=UTF-8" />
<html><head>
<title>Демонстрация тегов core</title>
</head>
<h3>Демонстрация работы тегов <br/>c:forEach, c:choose,
c:when, c:otherwise</h3>
<jsp:text>Ниже приведены случайно сгенерированные элементы
массива <br/> и сделана их оценка по отношению к числу 50 :
<br/></jsp:text>
<jsp:useBean id="arr" class="chapt21.Arr" />
<c:set var="items" value="${arr.fillMap}"
    scope="session" />
    <c:forEach var="id" items="${items}">
        <c:out value="${id}" />
        <c:choose>
            <c:when test="${id > 50}" >
                <c:out value=" - число больше 50"/>
            </c:when>
            <c:otherwise>
                <c:out value=" - число меньше 50"/>
            </c:otherwise>
        </c:choose>
    </c:forEach>
</html>
</jsp:root>

```

Элемент `JavaBean`, используемый в данном примере, – класс **Arr**, генерирующий массив из пяти случайных чисел. Его описание хранится в файле **Arr.java**. Исходный Java-файл хранится в каталоге на верхнем уровне приложения (например каталог **build**, **src** или **JavaSource**) в зависимости от настроек web-приложения. Значение атрибута **class** равно **chapt21.Arr** тега **jsp:useBean** и означает, что файл находится в пакете **chapt21**, а имя класса – **Arr**. EL-выражение **arr.fillMap** означает вызов метода **getfillMap()**.

```

// пример # 7 : java bean класс : Arr.java
package chapt21;
public class Arr {
    public Arr() {}
    public String[] getfillMap() {

```

```
String str[] = new String[5];
for (int i =0; i < str.length ; i++){
String r = Integer.toString((int) (Math.random()*100));
    str[i] = r;
}
return str;
}
```

В результате в браузер будет выведено:

Демонстрация работы тегов

c:forEach, c:choose, c:when, c:otherwise

Ниже приведены случайно сгенерированные элементы массива и сделана их оценка по отношению к числу 50:

8 - число меньше 50

68 - число больше 50

84 - число больше 50

5 - число меньше 50

36 - число меньше 50

Следующие примеры показывают работу тегов по взаимодействию с другими документами и ресурсами.

<!--пример # 8 : демонстрация работы тегов c:import, c:url, c:redirect, c:param : url.jsp -->

```
<%@ page language="java" contentType=
    "text/html; charset=Cp1251"
    pageEncoding="Cp1251"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
    prefix="c" %>
<html><head><title>Переход по ссылке</title></head>
<body>
<h3>Данная страница демонстрирует работу тегов
<br/>c:import, c:url, c:param, c:redirect</h3><br/>
<c:import url="\WEB-INF\jspf\imp.jspf"
    charEncoding="Cp1251"/>
<c:url value="redirect.jspx" var="myUrl" />
<a href='<c:out value="\${myUrl}" />' />Перейти</a>
</body></html>
```

<!--пример # 9 : фрагмент, включаемый с помощью тега c:import (находится в каталоге WEB-INF/jspf/) : imp.jspf -->

```
<h5>importing by using c:import from jspf</h5>
```

В результате запуска страницы **url.jsp** в браузер будет выведено:

Данная страница демонстрирует работу тегов

c:import, c:url, c:param, c:redirect.

importing by using c:import from jspf

Перейти

В **url.jsp** был импортирован фрагмент **imp.jspf**, а также с помощью тега **c:url** была задана активная ссылка, при активации которой будет осуществлен переход на страницу **redirect.jspx**.

```

<!--пример # 10 : демонстрация работы тегов c:redirect, c:param: redirect.jspx -->
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
          xmlns:c="http://java.sun.com/jsp/jstl/core"
          version="2.0">
<jsp:directive.page contentType=
    "text/html; charset=UTF-8" />
<html><head><title>Демонстрация тегов core</title></head>
<body>
<c:redirect url="urldestination.jspx">
    <c:param name="fname" value="Ostap"/>
    <c:param name="lname" value="Bender"/>
</c:redirect>
</body></html>
</jsp:root>

```

В документе были объявлены два параметра и заданы их значения, а также был автоматически выполнен переход на документ **urldestination.jspx**.

<!--пример # 11 : конечная страница, на которую был перенаправлен запрос и переданы данные: urldestination.jspx -->

```

<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
          xmlns:c="http://java.sun.com/jsp/jstl/core"
          version="2.0">
<jsp:directive.page contentType=
    "text/html; charset=UTF-8"/>
<html><head>
    <title>Демонстрация работы тега c:url</title>
</head>
<body>
<jsp:text>
    Ваш запрос был перенаправлен на эту страницу<br/>
    Параметры, переданные с помощью тега c:param:<br/>
</jsp:text>
<c:forEach var="ps" items="${param}">
<c:out value="${ps.key} - ${ps.value}"/><br/>
</c:forEach>
</body></html>
</jsp:root>

```

В результате работы документа в браузер будет выведено:

Ваш запрос был перенаправлен на эту страницу.

Параметры, переданные с помощью тега c:param:

lname - Ostap

fname - Bender

JSTL fmt

Библиотека содержит теги форматирования и интернационализации.

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

— для обычной страницы JSP;

```
<jsp:root version="1.2" xmlns:fmt=
```