

プログラミング Java 3

第6回： JSPとJava Beans入門

Ivan Tanev

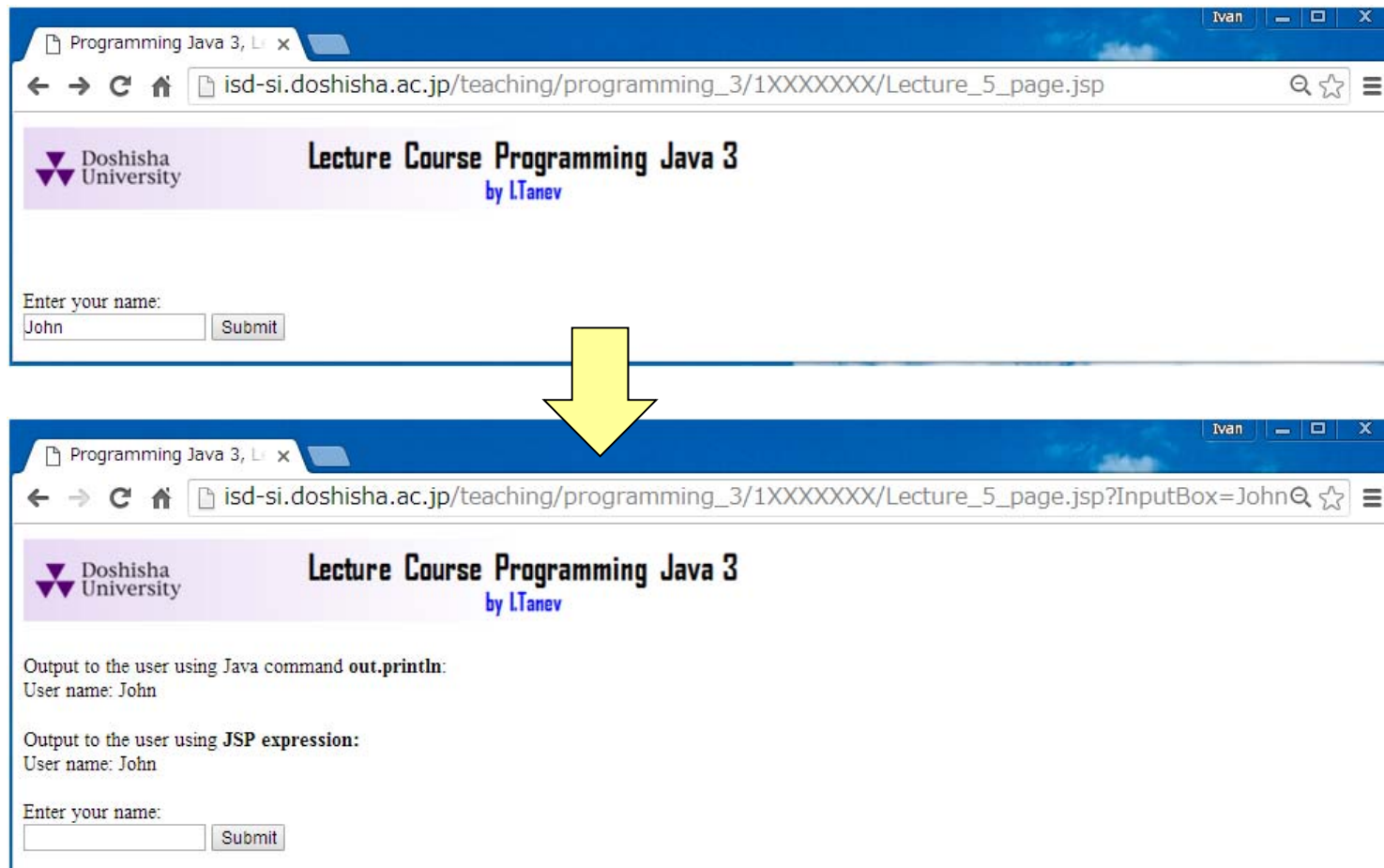


アウトライン

1. 第5回のまとめ
2. Java Beans 入門
3. JSPとJava Beans
4. ランタイム・ステップ
5. 演習

1. 第5回のまとめ

Lecture_5_page.jsp



1. 第5回のまとめ

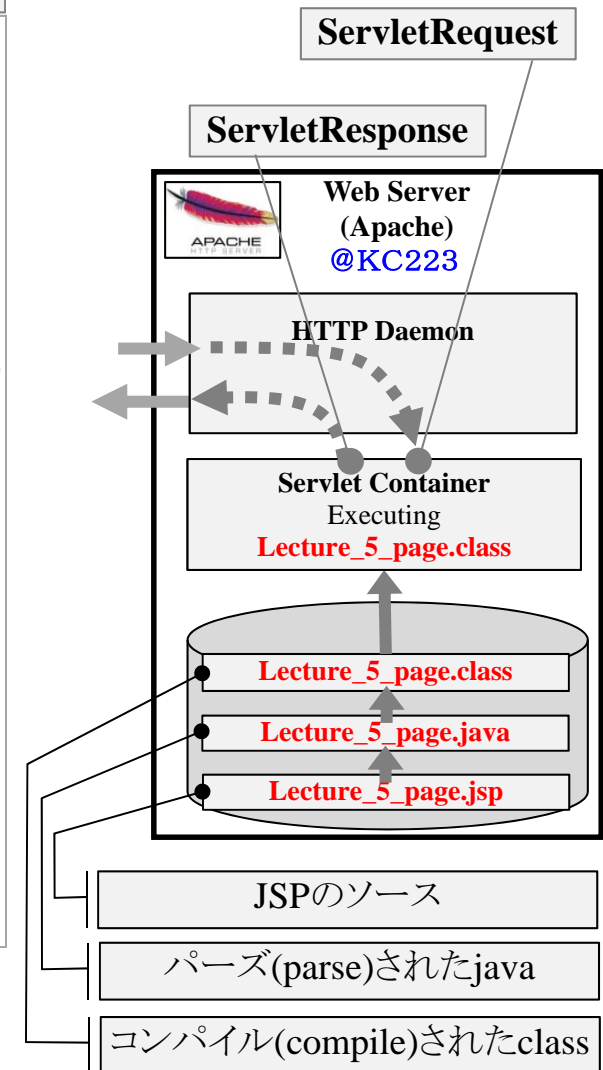
Lecture_5_page.jsp

```
<HTML>
<HEAD>
<TITLE> Programming Java 3, Lecture 5</TITLE>
<%@ page contentType= "text/html" %>
</HEAD>
<BODY>
<%@ include file = "logo.htm" %>
<% // Begin of Java code (scriptlet) in JSP
String userName = request.getParameter("InputBox");
if (userName != null) { // i.e., not the very first activation of JSP
// Output via Java command out.println:
out.println("Output to the user using Java command <B>out.println</B>:<BR>");
out.println("User name: "+ userName);
%> <!-- End of Java code (scriptlet) in JSP -->

<BR><BR> <!-- Two breaks (new lines) -->

<!-- Static HTML text: -->

Output to the user using <B>JSP expression:</B><BR>
User name: <%= userName %>
<% } %> <!-- End of the body of if statement -->
<BR><BR>
<FORM METHOD="GET" ACTION="Lecture_5_page.jsp" >
Enter your name:<BR>
<INPUT NAME = "InputBox">
<INPUT TYPE = "Submit" VALUE="Submit"/>
</FORM>
</BODY>
</HTML>
```



Q: JSPはStatic Contentをどのように作成しますか？

A: 直接、**HTMLタグがついているテキスト**で。Static Web page (htmlファイル)と同じです。

=> **JSP = HTML + <% %>のJava**

2. Java Beans入門



Q: ソフトウェアコンポーネントとは何ですか？

A: Modern **software** applications and systems are most often developed as assemblies of many smaller parts.

The idea of **software components** formalizes the definition of these “smaller parts”:

- A **software component** is basically a **software** unit with a well-defined interface and explicitly specified dependencies.
- A **software component** can be as small as a block of reusable code, or it can be as big as an entire application.

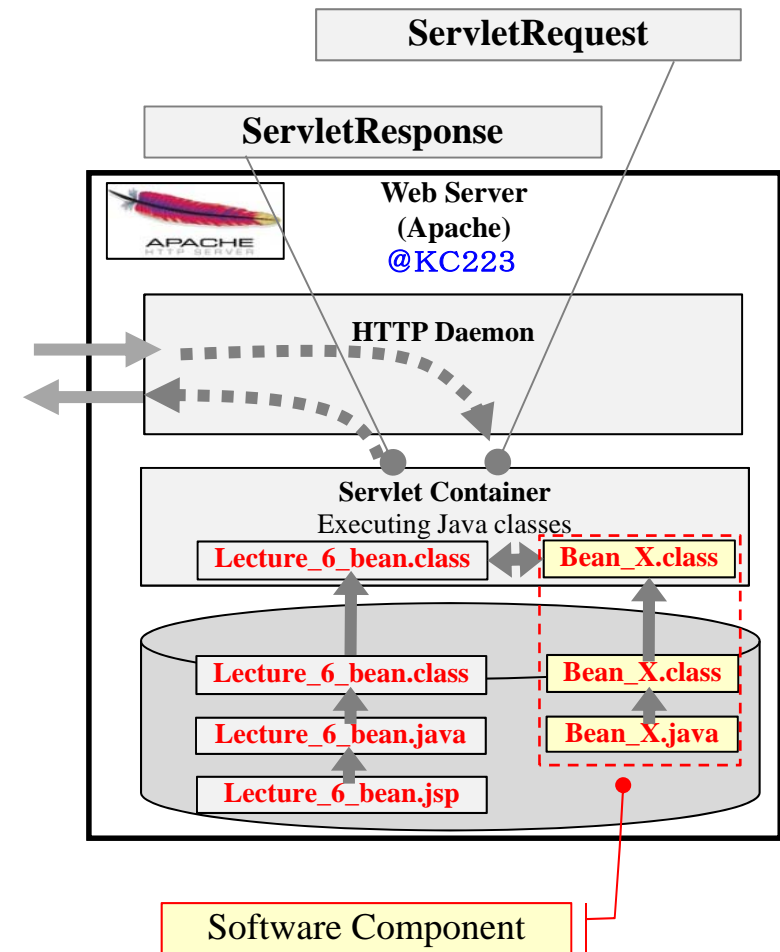
Component-oriented Programming

Component 1

Component 2

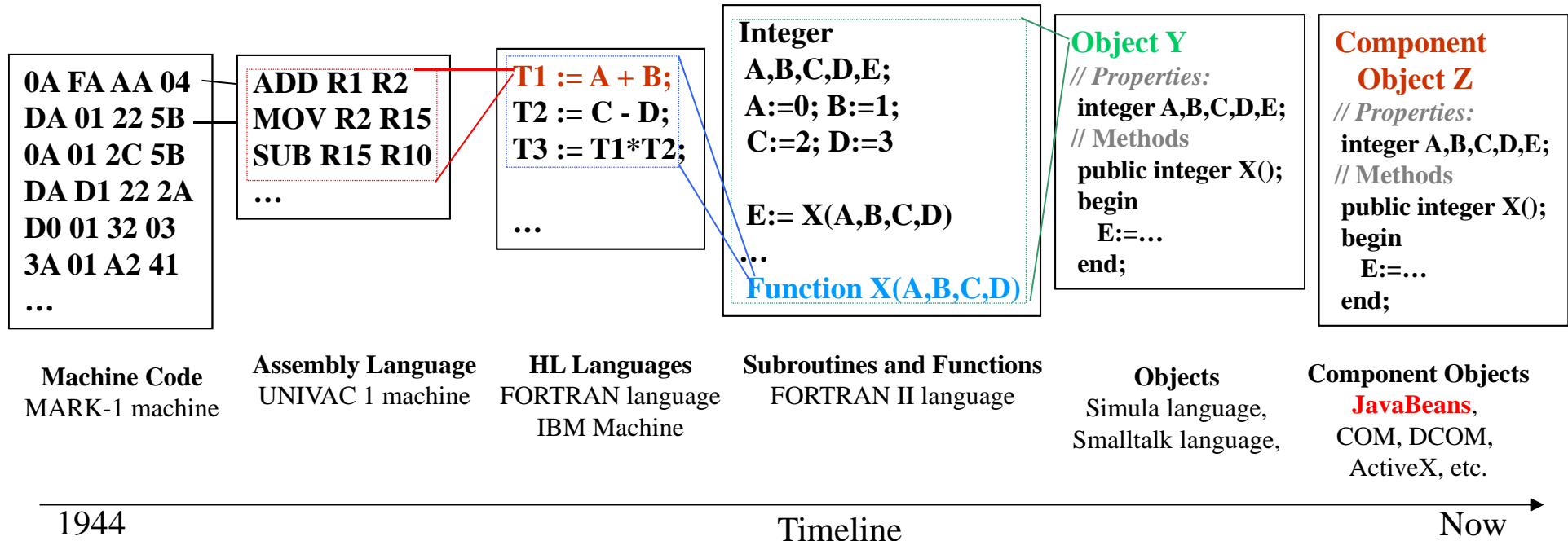
Component N

Software



2. Java Beans入門

ソフトウェア工学の進化



Q: オブジェクトとソフトウェアコンポーネントの違いは何ですか？

- A:
- (1) コンポーネントはアプリケーションのようなもので、OSではプロセスとして管理されます。
 - (2) コンポーネントには、別のアプリケーションから呼び出すことができるAPI (プロパティとメソッドのセット) があります。
 - (3) 呼び出し元のアプリケーションとコンポーネントは、異なるプログラム言語を使用して開発できます。

例:



Microsoft Agent

2. Java Beans入門

Java Beansの定義

Java Beansは、次のように定義されている。

"A Java Bean is a **reusable software component** written in **Java** that can be manipulated visually in a builder tool."

Categories of Beans: Visual Component Beans for GUI, data beans (data access), service beans (calculations),...

Java Beansの特徴

Java言語そのもの

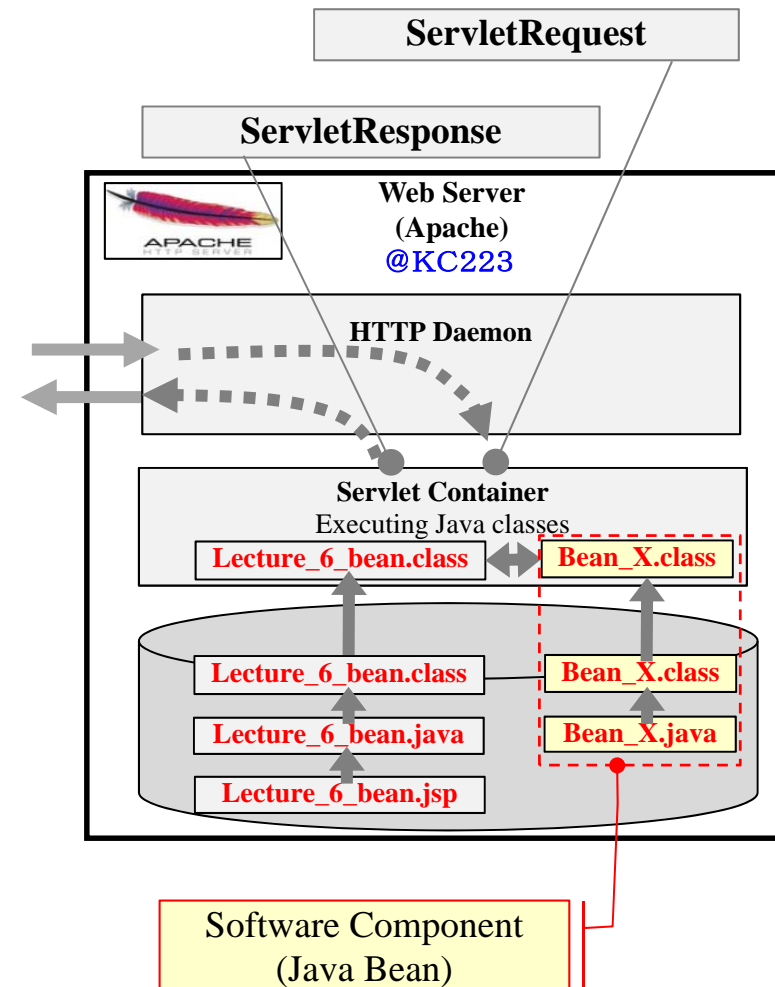
Java Beansを作る際に特別なものは、何も付加されておらず既存の**Java言語**そのものでよい。

コンパクトで簡単

Beanは**シンプル**であり、簡単に開発して、簡単に使用できる。

ポータブル

100% pure JavaのためJavaの実行環境をサポートするすべてのプラットフォームで動作する。すなわちJavaと同様に「**Write once - Run everywhere**」が成り立つ。



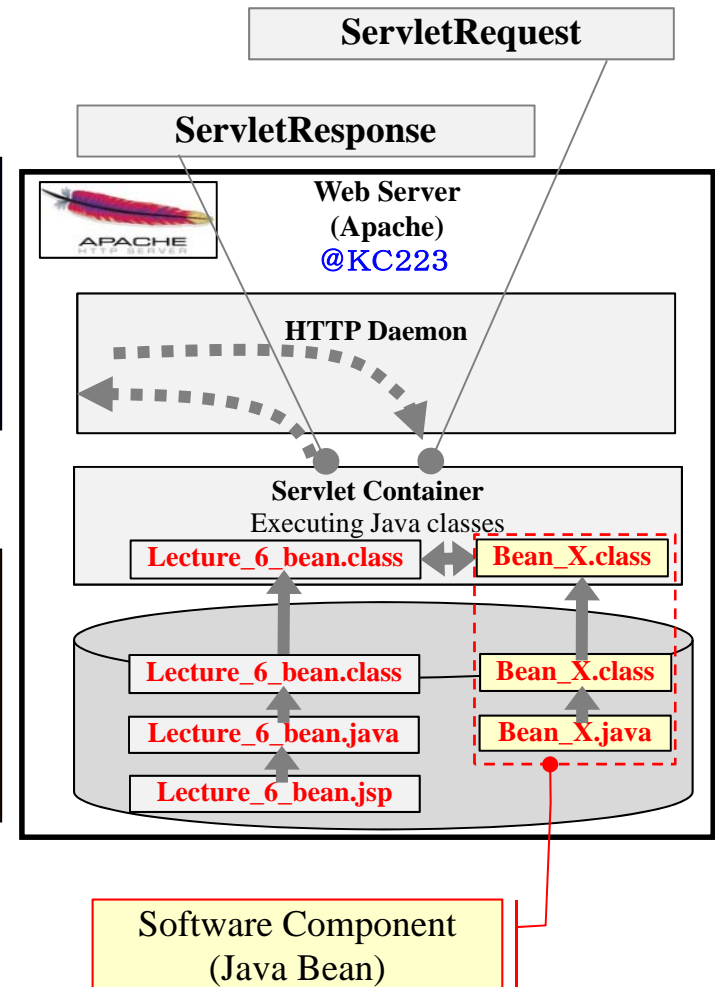
2. Java Beans入門


Java Beans: The Concept (例: NameBean.java)

```
package beans;  
public class NameBean {  
    String newName="";  
    public void NameBean() { }  
    public String getNewName() {  
        return newName;  
    }  
    public void setNewName(  
        String newName)  
    {  
        this.newName = newName;  
    }  
}
```

プロパティ(Property)
プロパティは名前によって参照されるBeanの属性のこと。通常は、それぞれのプロパティの値を取得するメソッド(getter)、設定するメソッド(setter)をこのプロパティ毎に作る必要がある。

メソッド(Methods)
Beanのメソッドは通常のJavaプログラムのメソッドと変わるところはない。このメソッドはBeanのクラスにより公開されたpublicメソッドである。これらのメソッドにより目的のコンポーネントにアクセスし操作等を行う。



- 
- (1) この Bean は、ユーザー入力 (例: ユーザー名, Name) の「エコー, **Echo**」に使用されます。
 - (2) **Echo** (in computing): Also called an **echo check**, an **echo**, is data repeated back to the sender by the recipient, to verify that it was received correctly.

2. Java Beans入門

Java Beans: The Concept (例: NameBean.java)

```
package beans;  
public class NameBean {  
  
    String newName="";  
  
    public void NameBean() { }  
  
    public String getNewName() {  
        return newName;  
    }  
  
    public void setNewName(  
        String newName)  
    {  
        this.newName = newName;  
    }  
}
```

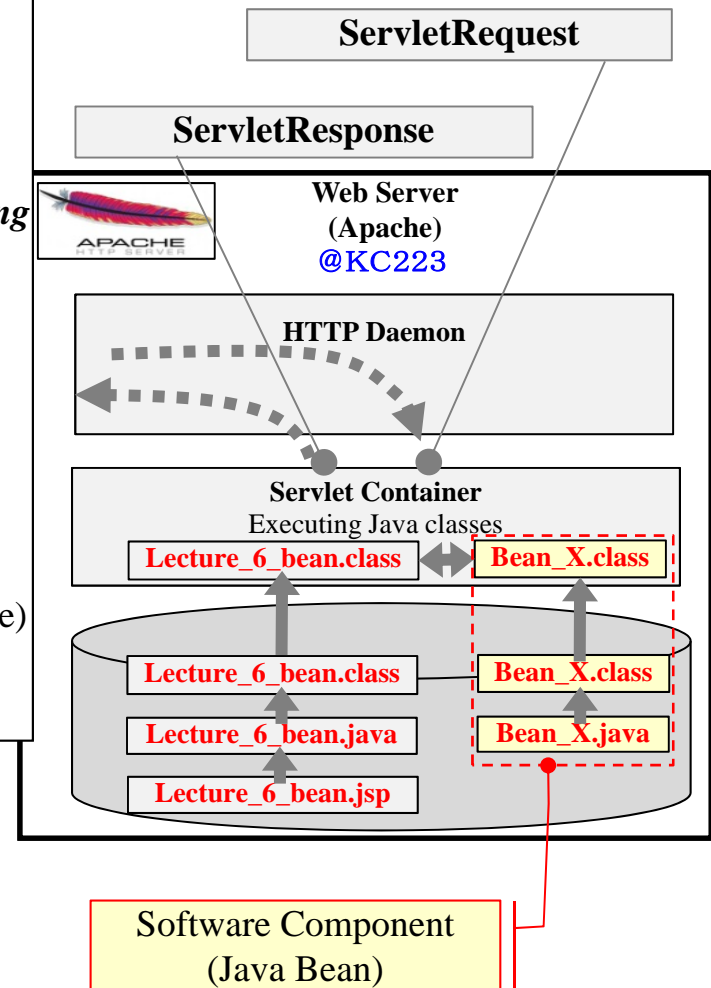
Conventions

- No-parameters constructor
- property *p* of type *T*
例: property *newName* of type *String*

Getter method:
public T getP()
例: **public String getNewName()**

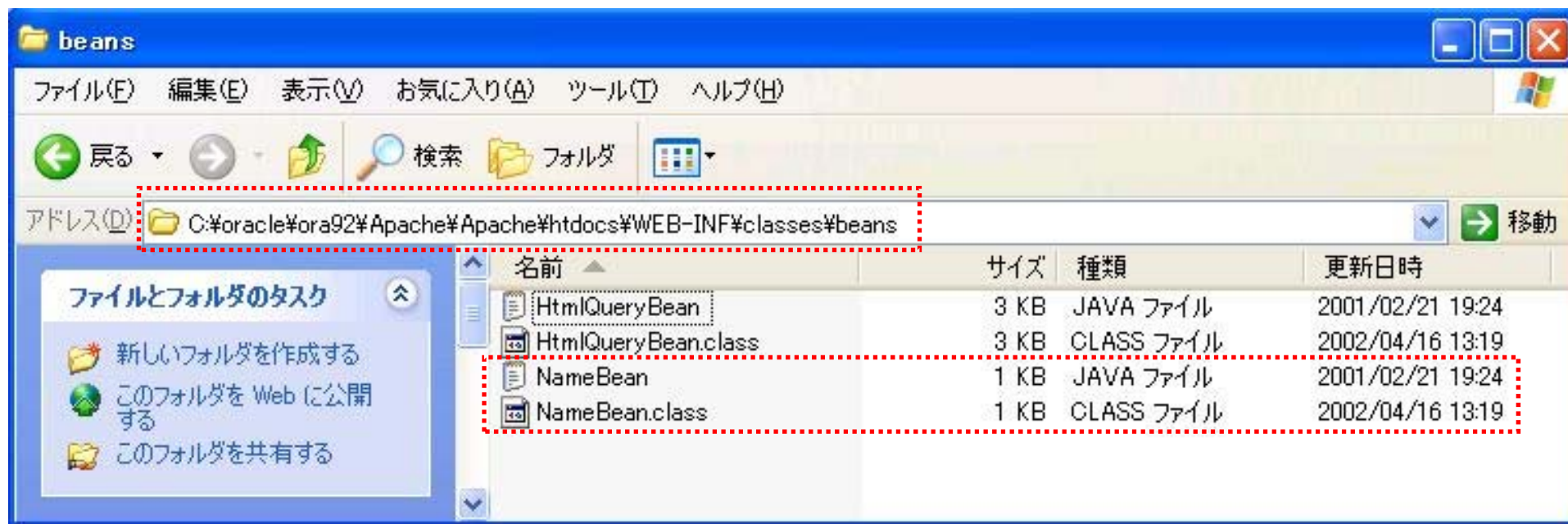
Setter method:
public void setP(T)
例: **public void setNewName(
String newName)**

- Methods should be **public**



2. Java Beans入門

Java BeanのLocation@ Apache Web Server



3. JSPとJava Beans

3.1 jsp:useBean(ビーンのインスタンスを作成)

形式 `<jsp:useBean id="beanInstanceName" scope="page|request|session|application" { class="package.class" | type="package.class" | class="package.class" type="package.class" | beanName="{package.class | <%= expression %>}" type="package.class" } { /> Other elements... </jsp:useBean> }`

例 `<jsp:useBean id = "pageBean" class = "beans.NameBean" scope = "page" />`

```
package beans;
public class NameBean {

    String newName="";

    public void NameBean() { }

    public String getNewName() {
        return newName;
    }

    public void setNewName(
        String newName)
    {
        this.newName = newName;
    }
}
```

3. JSPとJava Beans

3.2 `jsp:getProperty`(ビーンの`getXXX`メソッドを呼び出す)

形式 `<jsp:getProperty name="beanInstanceName" property="propertyName" />`

例 `<jsp:getProperty
 name = "pageBean"
 property = "newName" />`

または

`<% pageBean.getNewName() %>`

```
package beans;  
public class NameBean {  
    String newName="";  
  
    public void NameBean() { }  
  
    public String getNewName() {  
        return newName;  
    }  
  
    public void setNewName(  
        String newName)  
    {  
        this.newName = newName;  
    }  
}
```



Bean名(Beans ID) は`pageBean`です。下記の命令で設定されています:

`<jsp:useBean id="pageBean" class="beans.NameBean" scope = "page" />`

参考: 前のSlide #11

3. JSPとJava Beans

3.3 jsp:setProperty(ビーンのsetXXXメソッドを呼び出す)

形式 `<jsp:setProperty name="beanInstanceName" property="propertyName" />`

例 `<jsp:setProperty
name = "pageBean"
property = "newName"
param = "newName" />`

または

`<% pageBean.setNewName() %>`

```
package beans;  
public class NameBean {  
  
    String newName="";  
  
    public void NameBean() { }  
  
    public String getNewName() {  
        return newName;  
    }  
  
    public void setNewName(  
        String newName)  
    {  
        this.newName = newName;  
    }  
}
```

3. JSPとJava Beans 例 #1

```
<HTML>
<HEAD>
  <TITLE> Programming 3, Lecture 6 (1)</TITLE>
</HEAD>
<BODY>
  <%@ include file = "logo.htm" %>
  <BR>
  JSP and Java Beans: using methods <B>jsp:setProperty</B> and <B>jsp:getProperty</B>
  (Case 1)
  <BR><BR>
  <%@ page import="beans.NameBean" %>
  <% if (request.getParameter("newName") == null) { %>
    Enter your name:
  <% } else { %>
    <jsp:useBean id="pageBean" class="beans.NameBean" scope="page" />
    <jsp:setProperty name="pageBean" property="newName" param="newName" />
    Hello
    <B><jsp:getProperty name="pageBean" property="newName" /></B>
  <% } %>
  <FORM METHOD="GET" ACTION="Lecture_6_bean_1.jsp" >
    <INPUT NAME = "newName"/>
    <INPUT TYPE = "submit" VALUE="Submit Your Name"/>
  </FORM>
</BODY>
</HTML>
```

`<jsp:getProperty>`
`<jsp:setProperty>`

```
package beans;
public class NameBean {

  String newName="";

  public void NameBean() { }

  public String getNewName()
  {
    return newName;
  }

  public void setNewName(
  String newName)
  {
    this.newName = newName;
  }
}
```

3. JSPとJava Beans 例 #2

```
<HTML>
<HEAD>
  <TITLE> Programming 3, Lecture 6 (2)</TITLE>
</HEAD>
<BODY>
  <%@ include file = "logo.htm" %>
  <BR>
  JSP and Java Beans: using methods <B>pageBean.setNewName</B> and <B>pageBean.setNewName</B>
  (Case 2)
  <BR><BR>
  <%@ page import="beans.NameBean" %>
  <% String name = request.getParameter("newName");
  if (name == null) { %>
    Enter your name:
  <% } else { %>
    Hello
    <jsp:useBean id="pageBean" class="beans.NameBean" scope="page" />
    <% pageBean.setNewName(name); %><B>
    <%= pageBean.getNewName() %></B>
    !
  <% } %>
  <FORM METHOD="GET" ACTION="Lecture_6_bean_2.jsp" >
    <INPUT NAME = "newName" />
    <INPUT TYPE = "submit" VALUE="Submit" />
  </FORM>
</BODY>
</HTML>
```

pageBean.setNewName
pageBean.getNewName

```
package beans;
public class NameBean {

  String newName="";

  public void NameBean() { }

  public String getNewName()
  {
    return newName;
  }

  public void setNewName(
  String newName)
  {
    this.newName = newName;
  }
}
```

5. 演習

Lecture_6_Appendix.pdf

Pages 8～11



http://isd-si.doshisha.ac.jp/teaching/programming_3/Lectures/Lecture_6_Appendix.pdf

締め切り:

5月28日(木曜日), 23:59

The End