上級プログラミング2（第11回）

工学部　情報工学科
木村昌臣

今日のテーマ

- 先回までの復習

- Webアプリケーションプログラム
  - JSP
  - JavaBeansをつかったプログラム

Webアプリケーションプログラミング

JavaBeansを使ってみる

JavaBeans

- Webアプリケーションで共通に使われるJavaクラス（Bean）群をまとめたもの
  - 引数なしのコンストラクタのみ
    - コンストラクタの引数に入るデータが何かは自明でないため、必要なデータのセットはsetterで行う
  - プロパティを扱うことにより処理を行う
    - privateなフィールドを持つ
    - getterやsetterを持ち、そのフィールドがカプセル化されている
  - シリアライズ（永続化）可能

JavaBeans

- Beanが持つ情報をプロパティと呼ぶ
- プロパティはフィールドとsetterメソッド、getterメソッドの組で実現される
  - フィールドはprivateで宣言
  - setフィールド名()・・・・setter(変数に値のセット)
  - getフィールド名()・・・・getter(変数の値を取得)

直接、変数にアクセスするのではなく、メソッド経由でアクセスするカプセル化が行われていることに注意

サンプルプログラムにJavaBeansの処理を付け加えてみる

- Beanの仕様：入力された名前（アルファベット）の文字数を数えて表示
- JSPからBeanのインスタンスを呼び出すときには、Beanのクラスは必ずパッケージにする（でないと少なくともTomcatではエラーとなる）
- JSPやサーブレットに加えるべき処理は後述

Bean

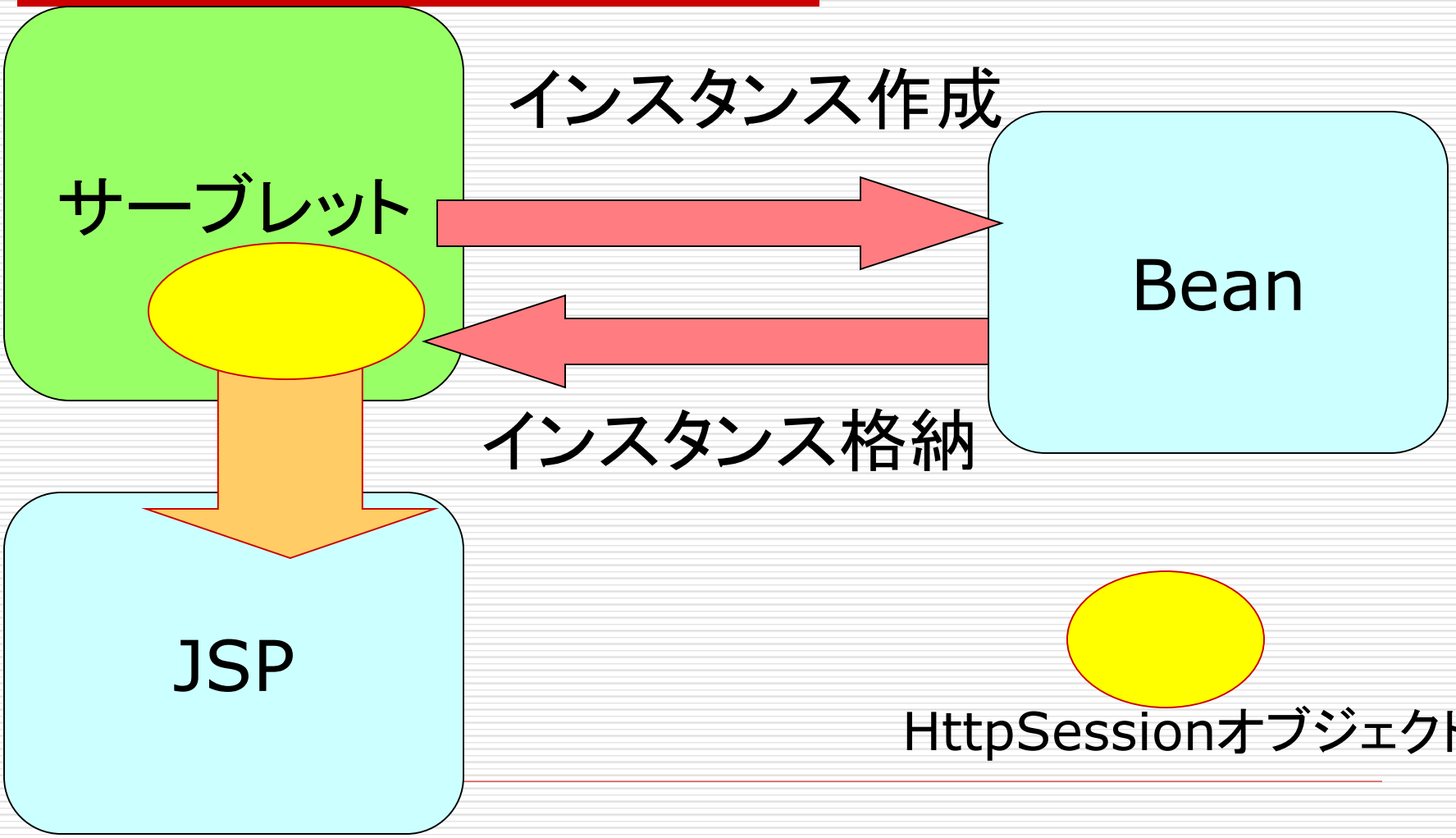
```
package simple;
public class SimpleBean {
	private String name="";
	private String message="";

	public void setName(String name){
		this.name=name;
	}
	public String getName(){
		return name;
	}
	public String getMessage(){
		return message;
	}
	public void calc(){
		int i=name.length();
		this.message="あなたの名前は、"+i+"文字です。";
	}
}
```

サーブレットへの処理の追加

- Beanのインスタンスを作成（当然、Beanが属するパッケージはインポートしておく）。
- 作成したインスタンスをJSPから参照できるオブジェクトに格納する
  - HttpServletRequestオブジェクト
    - 一回のリクエストの範囲で利用可能
  - HttpSessionオブジェクト
    - 同じクライアントマシンからのリクエストの範囲内で利用可能
  - ServletContextオブジェクト
    - Webアプリケーション全体で有効（複数ユーザーで共有すべき情報を格納）

JSPへのBeanオブジェクトの渡し方



サーブレット

```
....
import simple.*;

public class SimpleServlet3 extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {   String name=request.getParameter("MYNAME");
        HttpSession session= request.getSession(true);
        SimpleBean sb = new SimpleBean();
        sb.setName(name);
        sb.calc();
        session.setAttribute("sb",sb);
        ....
```

暗黙オブジェクト

- JSPには、事前に宣言することなく利用できるオブジェクトがある（以下、主要なもの）
  - out
    - コンテンツを出力するためのオブジェクト
  - request (HttpServletRequestオブジェクトに対応)
    - リクエスト（要求）情報にアクセスするオブジェクト
  - session (HttpSessionオブジェクトに対応）
    - セッション内で共有可能な情報を格納するオブジェクト
  - response (HttpServletResponseオブジェクトに対応）
    - レスポンス(応答)情報にアクセスするオブジェクト
  - application (ServletContextオブジェクトに対応）
    - 複数ユーザー間で共有可能な情報を格納するオブジェクト

JSP

```
<%@ page contentType="text/html; charset=Shift_JIS" %>
<%@ page import="simple.SimpleBean" %>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
                                        charset=Shift_JIS">
<title>Hello world</title>
</head>
<body>
<h1> こんにちは </h1>
<%
  SimpleBean sb= (SimpleBean)session.getAttribute("sb");
%>
<h2>ようこそ！<%= sb.getName() %>さん、こんにちは！</h2>
余計なお世話ですが、<%= sb.getMessage() %>
</body>
</html>
```

jsp:useBeanタグとjsp:getPropertyタグ

- JSPはHTMLベースの記載方法をとるので、なるべくタグで表現したい。そのため、以下のスクリプトレットは次のように置き換えることができる。

```
<%
  SimpleBean sb= (SimpleBean)session.getAttribute("sb");
%>
<jsp:useBean id="sb" class="simple.SimpleBean"
                                        scope="session" />

<%= sb.getName() %>
<jsp:getProperty name="sb" property="name" />
```

スコープ

JSP

```
<%@ page contentType="text/html; charset=Shift_JIS" %>
<jsp:useBean id="sb" class=" simple.SimpleBean"
                                        scope="session" />
<html>
<head>
<title>Hello world</title>
</head>
<body>
<h1> こんにちは </h1>
<h2>ようこそ！
<jsp:getProperty name="sb" property="name" />
さん、こんにちは！</h2>
余計なお世話ですが、
<jsp:getProperty name="sb" property="message" />
</body>
</html>
```

スコープ

- page
  - JSP内のみで有効＝ローカル変数と同じ
- request
  - Webブラウザからの要求がでてから処理結果がブラウザに返るまで有効
- session
  - ユーザーが、そのWebアプリケーションを利用している間ずっと有効（ブラウザを閉じるまで）
- application
  - ユーザーによらずWebアプリケーション全体で有効

sessionがなぜ必要か

- HTTPプロトコルはステートレス
  - HTTPは、もともとWebのソース＝HTMLをサーバーからダウンロードすることが目的
  - よって、HTMLを要求するユーザーが同一か否かについて無頓着
  - ショッピングサイトのようにユーザーとサーバーの間で情報をなんどもやり取りする場合には不向き
  - そのようなHTTPの欠点を補うため、クッキーの機能を利用したHttpSessionオブジェクトを利用してアクセスしているユーザーの判別およびデータの引継ぎを行う

テストに向けて

テストは来週実施

- 持ち込み不可
- 各回の内容をきちんと復習しておくこと
- 範囲はまんべんなく
  - Javaの文法
  - ネットワークプログラミング、GUIプログラミング、データベースプログラミング、Webアプリケーションプログラミング

など