

# Webアプリケーション開発演習A

JavaBeansとは？

情報システム専攻科2年

A blue header area at the top left of the slide, featuring various white geometric shapes like circles, squares, and diamonds, some of which are nested or overlapping. The shapes are scattered across the blue background, which curves downwards towards the right.

**今日の目標！**

**JavaBeansを知ろう**

- **MVCモデル**とは、メンテナンス性が高いソースコードを組む為の**考え方**！  
M(モデル)、V(ビュー)、C(コントローラ)の3つの役割に沿ってソースコードを組んでいく。

**M(モデル)・・・Javaクラス(主たる処理を書く)**

**V(ビュー)・・・JSP(結果の表示(画面))**

**C(コントローラ)・・・サーブレット(処理の振り分け)**

- 通常はリクエストを**C**が受け、**M**に処理を依頼し結果を**V**で表示する。

A decorative blue header at the top left of the slide, featuring various white geometric shapes such as circles, squares, and diamonds, some of which are nested or overlapping.

# 目次

## JavaBeansとは？ 演習

A decorative blue header at the top left of the slide, featuring various white geometric shapes like circles, squares, and diamonds, some of which are nested or overlapping.

# 目次

## JavaBeansとは？

### 演習

A decorative blue header with a white curved border. It contains several light blue geometric shapes: circles, squares, and diamonds, some of which are nested or overlapping.

# JavaBeansとは？

**さっそくですが、JavaBeansとは  
何でしょうか？**

**教科書P.191～193**

# JavaBeansとは？

はい、よくわかりませんでしたね・・・



A blue header area at the top of the slide, featuring various white geometric shapes like circles, squares, and diamonds scattered across it.

# JavaBeansとは？

**どんな場面でjavaBeansを使うかを  
考えて見ます。**



# JavaBeansとは？

## 会員登録

### 入力画面

同意画面 > **入力画面** > 確認画面 > 登録完了 >

お客様情報を入力してください。入力が終わりましたら「次へ」を選択してください。

※は必須入力です。

|                    |   |                    |  |
|--------------------|---|--------------------|--|
| 氏名(カナ)※            | セイ <input type="text"/> メイ <input type="text"/><br>全角カタカナで入力してください。 例) シブヤ タロウ            | 携帯メールアドレス※         | <input type="text"/><br>@ドメインを選択 ▼<br>半角英数で入力してください。<br>確認のためもう一度入力してください。 |
| 氏名(漢字)※            | 姓 <input type="text"/> 名 <input type="text"/><br>全角で入力してください。 例) 渋谷 太郎                    |                    |  |
| 電話番号※              | <input type="text"/><br>ハイフンなしの半角で入力してください。 例) 031234                                     |                    |  |
| 生年月日※              | <input type="text"/>  |                    |  |
| 性別※                | <input type="text"/>  |                    |  |
| 郵便番号※              | <input type="text"/>  |                    |  |
| ご住所※<br>(都道府県)     | <input type="text"/>  |                    |  |
| ご住所※<br>(市区町村)     | <input type="text"/>  |                    |  |
| ご住所※<br>(地域名地番)    | <input type="text"/>  |                    |  |
| ご住所<br>(マンション・建物名) | <input type="text"/><br>全角で入力してください。ただし、ローマ数字(I, II, ...)<br>に置き換えて入力してください。 例) △△△マンション2 | ご希望のパスワード※         | お好きな半角英数字を入力してください。<br><input type="text"/><br>半角英数字4文字以上10文字以内            |
| ご職業※               | <input type="text"/><br>「その他」選択時のみ入力してください。(最大15文字)                                       | ご希望のパスワード<br>(確認)※ | 確認のためもう一度入力してください。<br><input type="text"/><br>半角英数字4文字以上10文字以内             |

入力項目が17個！

次へ

戻る

# JavaBeansとは？

**この17項目をセッションに保存する  
必要があるとしましょう。**

**そうするとソースコードは  
こんな感じになりますね・・・。**



# JavaBeansとは？

```
protected void doGet(HttpServletRequest request, HttpServletResponse re:  
    throws ServletException, IOException {
```

```
//JSPからのパラメータを取得
```

```
String nameKanji = request.getParameter("name_kanji");
```

```
String nameKana = request.getParameter("name_kana");
```

```
String tel = request.getParameter("tel");
```

```
...
```

```
String dm = request.getParameter("dm");
```

```
String s_password1 = request.getParameter("s_password1");
```

```
String s_password2 = request.getParameter("s_password2");
```

```
//セッションに保存
```

```
HttpSession session = request.getSession(true);
```

```
session.setAttribute("nameKanji", nameKanji);
```

```
session.setAttribute("nameKana", nameKana);
```

```
session.setAttribute("tel", tel);
```

```
...
```

```
session.setAttribute("dm", dm);
```

```
session.setAttribute("s_password1", s_password1);
```

```
session.setAttribute("s_password2", s_password2);
```

```
}
```

17回  
getParameter  
を書く

17回  
setAttribute  
を書く

# JavaBeansとは？

## JSPでセッションから値を取り出すときは...

17回  
getAttribute  
を書く

```
<body>
<%
//セッションからパラメータを取得
String nameKanji = (String)session.getAttribute("name_kanji");
String nameKana = (String)session.getAttribute("name_kana");
String tel = (String)session.getAttribute("tel");
.
String dm = (String)session.getAttribute("dm");
String s_password1 = (String)session.getAttribute("s_password1");
String s_password2 = (String)session.getAttribute("s_password2");
%>
</body>
```



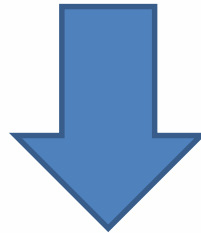
# JavaBeansとは？

**このコードをスッキリ、効率よく  
書くにはどうしたら良いでしょうか？**



# JavaBeansとは？

**17個の項目を「登録情報」として  
ひとつにまとめて扱ったらどうだろうか？**



**一つにまとめると  
setAttribute  
getAttribute  
は1回で済む！**



やってみよう!

# 演習

## javaBeansのプログラムを 作ってみよう

<https://github.com/nishino-naoyuki/2019Web-MVCサンプル2.pdf>



# JavaBeansとは？

具体的にはこういうことです。

- ① 17個の登録情報を持つクラスを作成
- ② サーブレットで、①のインスタンス作成
- ③ JSPから取得した値を①にセットする
- ④ セッションに①のインスタンスをセットする
- ⑤ セッションから①のインスタンスを取得する



# JavaBeansとは？

具体的な実装を見てみよう！

## ① 17個の登録情報を持つクラスを作成

```
public class RegisterInfo implements Serializable{
```

```
    private String nameKanji;
```

```
    private String nameKana;
```

```
    private String tel;
```

```
    .
```

```
    .
```

```
    private String dm;
```

```
    private String s_password1;
```

```
    private String s_password2;
```

```
    public String getNameKanji() {  
        return nameKanji;
```

```
    public void setNameKanji(String  
        this.nameKanji = nameKanji;
```

```
    public String getNameKana() {  
        return nameKana;
```

```
    }
```

```
    public void setTel(String tel) {  
        this.tel = tel;
```

```
    public String getDm() {  
        return dm;
```

```
    public void setDm(String dm) {  
        this.dm = dm;
```

```
    public String getS_password1() {  
        return s_password1;
```

```
    public void setS_password1(String s_password1) {  
        this.s_password1 = s_password1;
```

```
    public String getS_password2() {  
        return s_password2;
```

```
    }
```

# JavaBeansとは？

## ポイント

Serializableインターフェースを  
implementsする

```
public class RegisterInfo implements Serializable {
```

```
    private String nameKanji;
    private String nameKana;
    private String tel;
```

```
    private String dm;
    private String s_password1;
    private String s_password2;
```

変数は全てprivateする  
(カプセル化)

変数にアクセスする  
getter、setterを持つ

```
    public String getNameKanji() {
        return nameKanji;
    }
    public void setNameKanji(String nameKanji) {
        this.nameKanji = nameKanji;
    }
    public String getNameKana() {
        return nameKana;
    }
}
```

まとめ

今日の目標！

## MVCモデルを知ろう

- **MVCモデル**とは、メンテナンス性が高いソースコードを組む為の**考え方**！  
M(モデル)、V(ビュー)、C(コントローラ)の3つの役割に沿ってソースコードを組んでいく。

**M(モデル)・・・Javaクラス(主たる処理を書く)**

**V(ビュー)・・・JSP(結果の表示(画面))**

**C(コントローラ)・・・サーブレット(処理の振り分け)**

- 通常はリクエストを**C**が受け、**M**に処理を依頼し結果を**V**で表示する。