

## 第5章 データベース連携

本章で学習した内容を確認します。

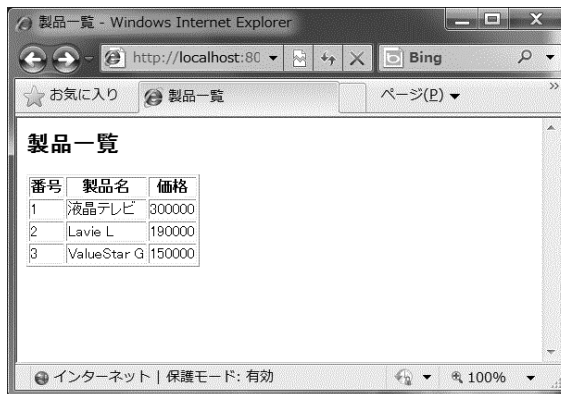
### 5.1 データベース連携（基本）

データベースと連携して製品一覧を表示するプログラムを作成します。次の（１）～（５）の作業を行なってください。作業の詳細は次ページ以降を参照してください。

- (1) データソースの設定
- (2) 製品情報を管理する JavaBeans のクラス、Product の作成
- (3) データベースアクセスを行う DAO クラス、ProductDAO の作成
- (4) 全体の制御を行うサーブレット、FindAllProductsServlet の作成
- (5) 製品一覧を表示する JSP、ProductList.jsp の作成

#### ■ 実行イメージ

URL : <http://localhost:8080/Labs/FindAllProductsServlet>



#### ■ テーブル構成

##### ◆商品（products）テーブル

列名	型	説明
no	INT AUTO_INCREMENT	商品の識別番号、主キー
name	VARCHAR(32)	商品名
price	INT	商品の単価（税抜き）

## (1) データソースの設定

データソースを利用するための設定を行ってください。本演習では、データソースの設定を「META-INF」フォルダの「context.xml」ファイルに記述するものとします。

## (2) 製品情報を管理する JavaBeans のクラス、Product の作成

次の要件に合うように Product クラスを作成します。

クラス名	beans.Product	
概要	製品情報を管理する JavaBeans のクラス。	
属性	private int no	製品番号
	private String name	製品名
	private int price	販売価格
操作	public Product()	
	コンストラクタ。処理は空。	
	public Product(int no, String name, int price)	
	コンストラクタ。引数の値をメンバ変数に代入する。	
	public int getNo()	
	no の getter。メンバ変数 no の値を返す	
	public void setNo(int no)	
	no の setter。引数の値をメンバ変数 no に代入する。	
	public String getName()	
	name の getter。メンバ変数 name の値を返す	
	public void setName(String name)	
	name の setter。引数の値をメンバ変数 name に代入する。	
	public int getPrice()	
price の getter。メンバ変数 price の値を返す		
public void setPrice(int price)		
price の setter。引数の値をメンバ変数 price に代入する。		

- (3) データベースアクセスを行う DAO クラス、ProductDAO の作成  
次の要件に合うように ProductDAO クラスを完成させてください。テンプレートがありません。

クラス名	dao.ProductDAO
概要	データベースアクセスを行う DAO クラス。products テーブルへアクセスする。
属性	private Connection con データベース接続を表す Connection オブジェクト。
操作	public ProductDAO(Connection con) コンストラクタ。引数で指定されたオブジェクトをメンバ変数に代入する。 public List<Product> findAll() throws SQLException データベースに SELEC 文を発行して製品一覧を検索し、検索結果を格納したリストを返す。

ProductDAO.java (未完成)

```

01: package dao;
02:
03: public class ProductDAO {
04:     // Connectionオブジェクトを保持するメンバ変数
05:
06:     // コンストラクタ
07:
08:     // 製品一覧を返す
09:     public List<Product> findAll() throws SQLException {
10:         // 製品一覧用リスト生成
11:
12:         // SQL文を準備
13:
14:         // SQL文を発行
15:
16:         // 検索結果を処理
17:
18:         // リソースの解放
19:
20:         // 戻り値を返却
21:
22:     }
23: }

```

- (4) 全体の制御を行うサーブレット、FindAllProductsServlet の作成  
次の要件に合うように FindAllProductsServlet を完成させてください。テンプレートがあります。

クラス名	Servlet.FindAllProductsServlet
概要	製品一覧の検索を制御するサーブレット。
属性	private DataSource source データベース接続を管理する DataSource オブジェクト。
操作	public void init() 初期化処理を行う。ネーミングサービスから DataSource オブジェクトを取得し、メンバ変数に代入する。 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException データソースから Connection オブジェクトを取得し、ProductDAO オブジェクトを生成する。生成した DAO オブジェクトの findAll() メソッドを利用して取得した製品一覧をリクエストに格納し、ProductList.jsp へフォワードする。

FindAllProductsServlet.java (未完成)

```

01: package Servlet;
02:
03:     :
04:
05:
06:
07:
08:
09:
10: public class FindAllProductsServlet extends HttpServlet {
11:     // メンバ変数の定義
12:
13:     // データソースを初期化
14:     public void init() throws ServletException {
15:         // ネーミングサービスに接続
16:
17:         // ネーミングサービスからデータソースを取得
18:
19:     }
20:
21:     protected void doGet(HttpServletRequest request,
22:         HttpServletResponse response) throws ServletException, IOException {
23:
24:         // データソースからConnectionを取得
25:
26:         // DAOの生成
27:
28:         // findAllメソッドを呼び出し
29:
30:         // 結果をリクエストに登録
31:
32:         // 次ページへフォワード
33:
34:     }
35: }

```

## (5) 製品一覧を表示する JSP、ProductList.jsp の作成

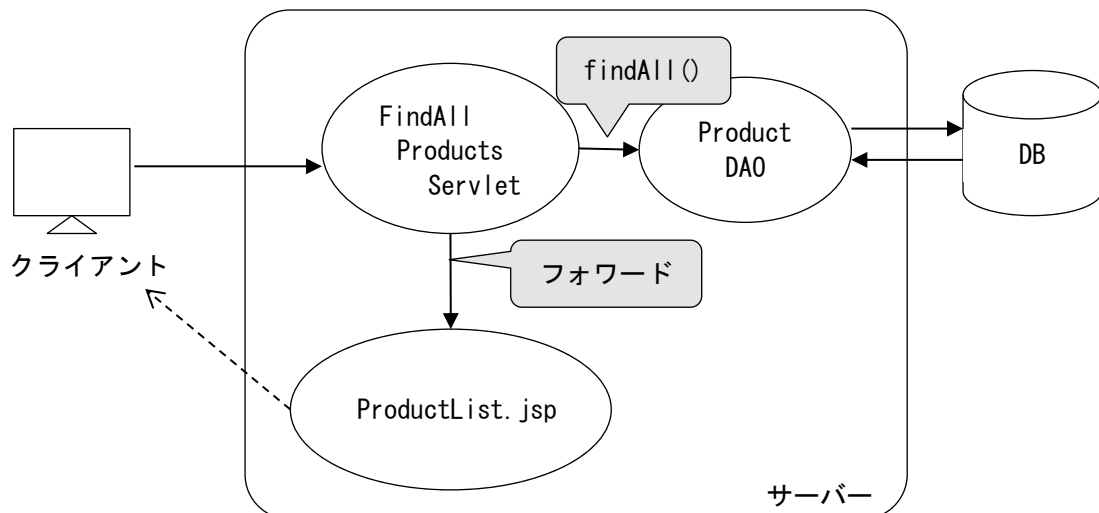
実行結果を参考に、ProductList.jsp ファイルを完成させてください。テンプレートがあります。

ProductList.jsp (未完成)

```

01: <%@ page language="java" contentType="text/html; charset=windows-31j"
02:     pageEncoding="windows-31j"%>
03: <%-- 必要なパッケージをインポート --%>
04:
05: <%-- リクエストから製品リストを取り出す --%>
06:
07: <html>
08: <head>
09: <meta http-equiv="Content-Type" content="text/html; charset=windows-31j">
10: <title>製品一覧</title>
11: </head>
12: <body>
13: <h2>製品一覧</h2>
14: <table border=1>
15: <tr><th>番号</th><th>製品名</th><th>価格</th></tr>
16:
17: <%-- 製品リストの内容をテーブルに表示する --%>
18:
19: </table>
20: </body>
21: </html>

```



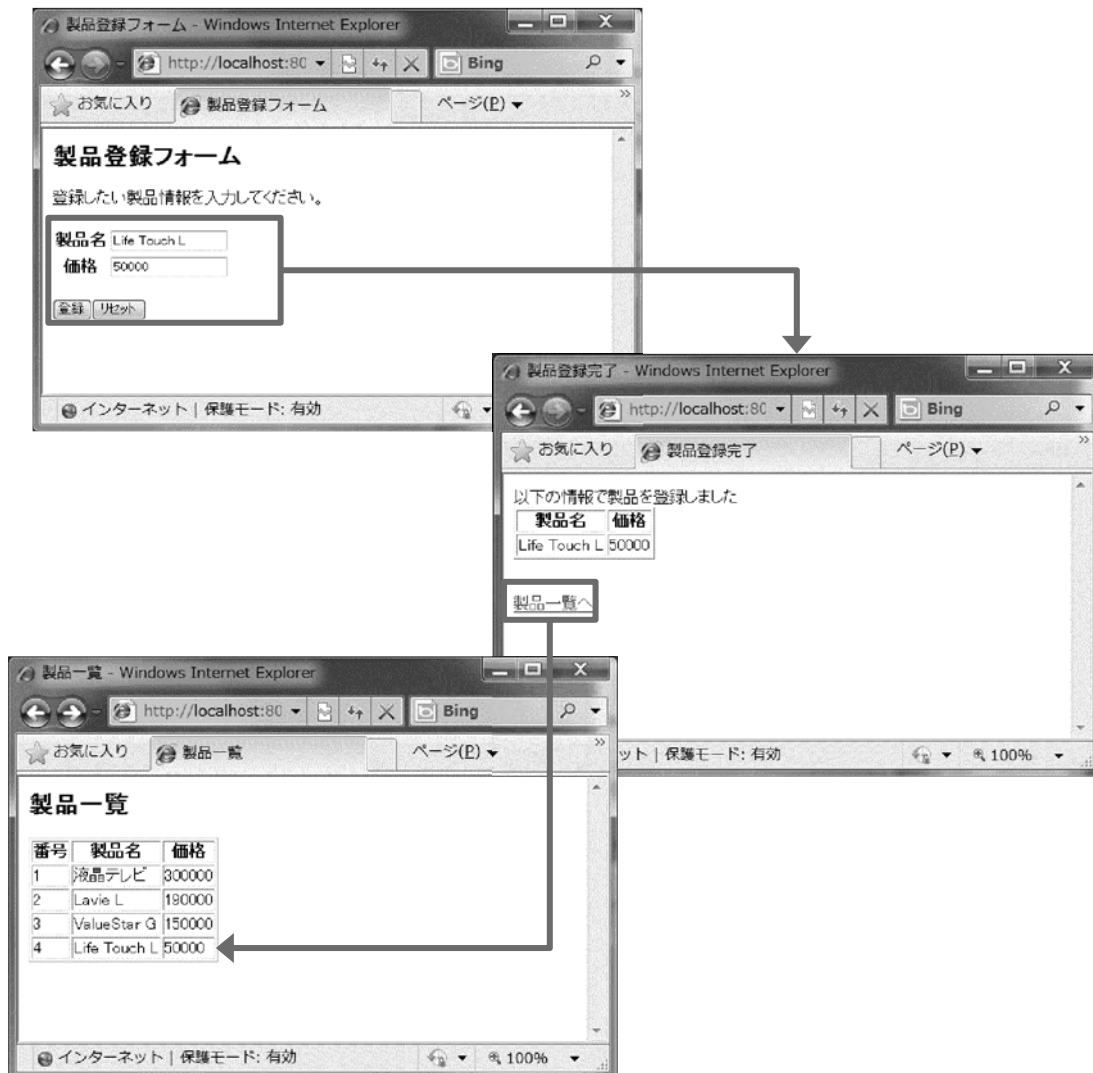
## 5.2 データベース連携（基本）

データベースと連携して製品情報を登録するプログラムを作成します。次の（１）～（４）の作業を行なってください。作業の詳細は次ページ以降を参照してください。

- （１） 製品情報登録用のフォーム insert-product-form.html を確認
- （２） DAO クラス、ProductDAO の修正
- （３） 全体の制御を行うサーブレット、ProductInsertServlet の作成
- （４） 製品登録結果を表示する JSP、ProductInsertOK.jsp の作成

### ■ 実行イメージ

URL : <http://localhost:8080/Labs/insert-product-form.html>



## (1) 製品情報登録用のフォーム insert-product-form.html を確認

insert-product-form.html (作成済み)

```

01: <html>
02: <head>
03: <meta http-equiv="Content-Type" content="text/html; charset=windows-31j">
04: <title>製品登録フォーム</title>
05: </head>
06: <body>
07: <h2>製品登録フォーム</h2>
08: 登録したい製品情報を入力してください。<br>
09: <form action="InsertProductServlet" method="POST">
10: <table>
11: <tr><th>製品名</th><td><input type="text" name="name"></td></tr>
12: <tr><th>価格</th><td><input type="text" name="price"></td></tr>
13: </table><br>
14: <input type="submit" value="登録"><input type="reset" value="リセット">
15: </form>
16: </body>
17: </html>

```

## (2) DAO クラス、ProductDAO の修正

ProductDAO クラスに製品情報を登録する次のメソッドを追加してください。

クラス名	dao.ProductDAO (修正)
概要	データベースアクセスを行う DAO クラス。products テーブルへアクセスする。
操作	<pre>public int insert(Product product) throws SQLException</pre> <p>データベースに INSERT 文を発行して引数で指定された製品情報を登録する。製品番号については、データベースの自動採番機能を使う。戻り値として更新した行数を返す。</p>

※ 変更部分のみを記載

## (3) 全体の制御を行うサーブレット、InsertProductServlet の作成

次の要件に合うように InsertProductServlet を完成させてください。テンプレートがあります。

クラス名	servlet.InsertProductServlet
概要	製品登録を制御するサーブレット。
属性	private DataSource source データベース接続を管理する DataSource オブジェクト。
操作	public void init() 初期化処理を行う。ネーミングサービスから DataSource オブジェクトを取得し、メンバ変数に代入する。 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException データソースから Connection オブジェクトを取得して ProductDAO オブジェクトを生成し、生成した DAO オブジェクトの insert メソッドを利用して製品情報を登録する。登録した製品の情報をリクエストに格納して InsertProductOK.jsp へフォワードする。

InsertProductServlet.java (未完成)

```

01: package servlet;
   :
09:
10: public class InsertProductServlet extends HttpServlet {
11:     // メンバ変数の定義
12:
13:     public void init() throws ServletException {
14:         /* FindAllProductsServletのinitメソッドと同じ */
15:     }
16:
17:     protected void doPost(HttpServletRequest request,
18:         HttpServletResponse response) throws ServletException, IOException {
19:         // リクエストパラメータの読み込み
20:
21:         // 登録用のProductオブジェクトを生成
22:
23:         // データソースからConnectionを取得
24:
25:         // DAOの生成
26:
27:         // ProductDAOのinsertメソッドを実行
28:
29:         // DBIに登録した製品情報をリクエストに格納
30:
31:         // 次ページへフォワード
32:
33:     }
34: }

```



- (4) 製品登録結果を表示する JSP、ProductInsertOK.jsp の作成  
 実行結果を参考に ProductInsertOK.jsp を完成させてください。テンプレートがあります。

InsertProductOK.jsp (未完成)

```

01: <%@ page language="java" contentType="text/html; charset=windows-31j"
02:     pageEncoding="windows-31j"%>
03: <html>
04: <head>
05: <meta http-equiv="Content-Type" content="text/html; charset=windows-31j">
06: <title>製品登録完了</title>
07: </head>
08: <body>
09: 以下の情報で製品を登録しました<br>
10: <table border="1">
11: <tr><th>製品名</th><th>価格</th></tr>
12:
13: <%-- 登録した製品情報をリクエストから取り出して表示 --%>
14:
15: </table><br>
16: <br>
17: <a href="/Answers/FindAllProductsServlet">製品一覧へ</a>
18: </body>
19: </html>
    
```

