

---

# ECサイト中級

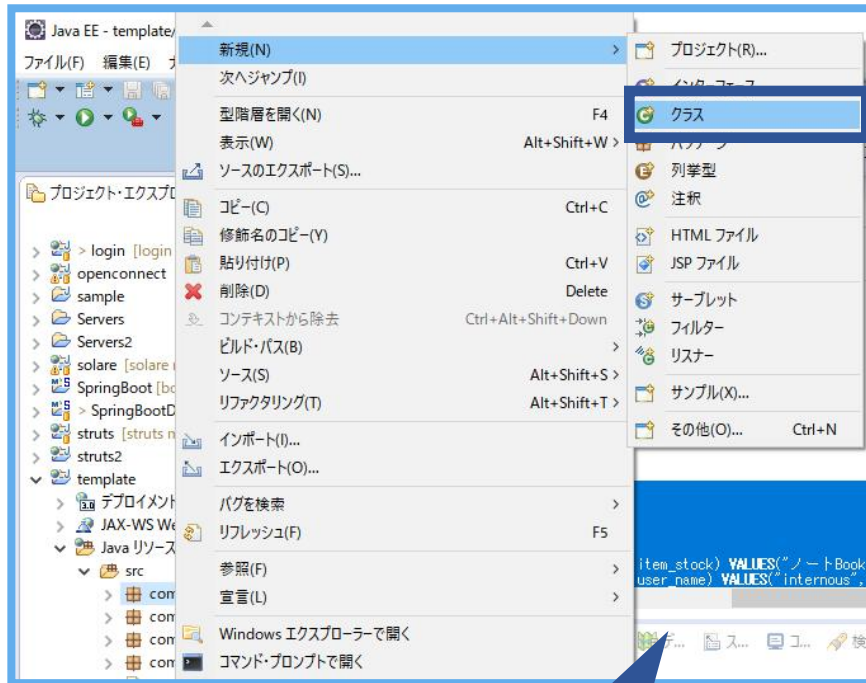
商品購入機能の作成

---

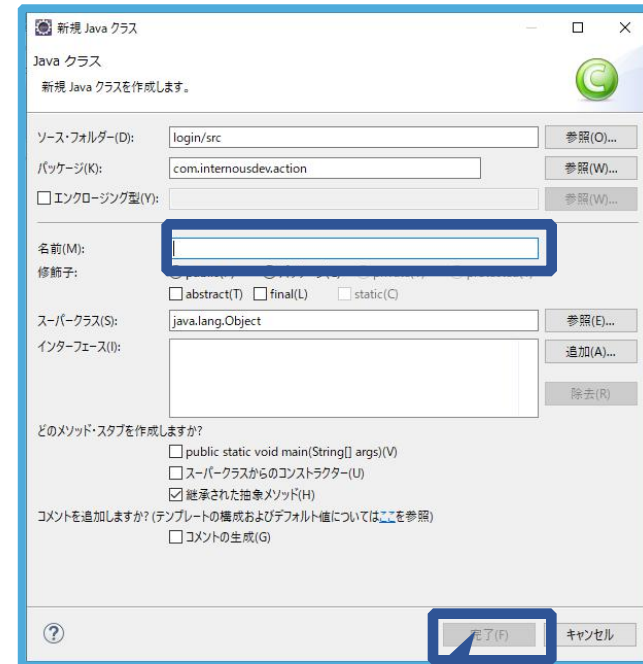
## 5 時間目

# 商品画面用Actionクラスの作成

## 1 Actionクラス



① 「com.internousdev.template.action」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「HomeAction」を入力し、完了ボタンをクリックします。

# HomeActionクラスの作成

HomeAction(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.ecsite.action;

import java.util.Map;
import org.apache.struts2.interceptor.SessionAware;
import com.opensymphony.xwork2.ActionSupport;

public class HomeAction extends ActionSupport implements SessionAware {

    public Map<String, Object> session;
```

次へ続きます。

# HomeActionの作成

前の続きです。

```
public String execute() {  
    String result = "login";  
    if (session.containsKey("id")) {  
        BuyItemDAO buyItemDAO = new BuyItemDAO();  
        BuyItemDTO buyItemDTO = buyItemDAO.getBuyItemInfo();  
        session.put("id", buyItemDTO.getId());  
        session.put("buyItem_name", buyItemDTO.getItemName());  
        session.put("buyItem_price", buyItemDTO.getItemPrice());  
  
        result = SUCCESS;  
    }  
    return result;  
}  
  
@Override  
public void setSession(Map<String, Object> session){  
    this.session = session;  
}  
}
```

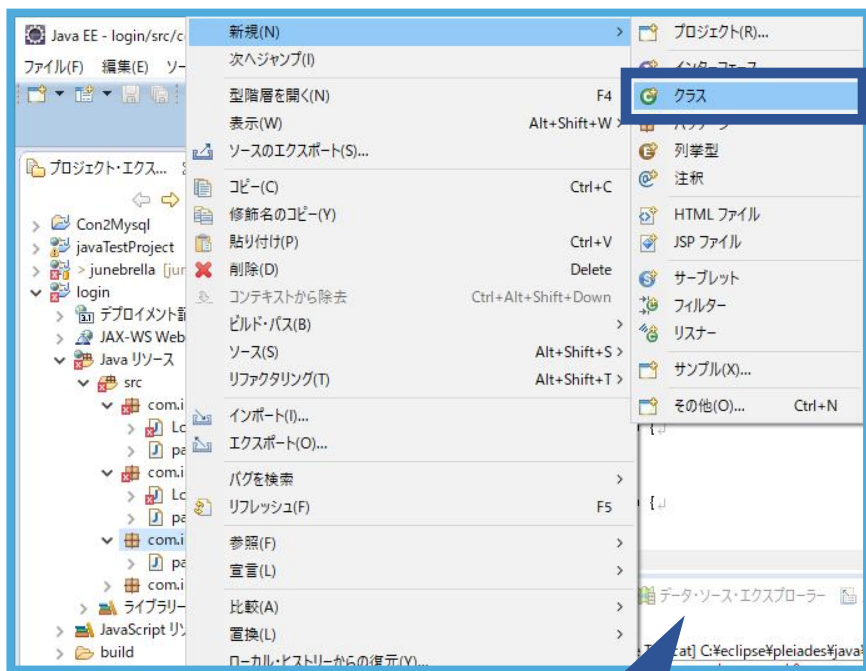
ログイン状態か否か、  
チェックをしています。

「execute」メソッドを実行した後は  
"success"文字列を返します。

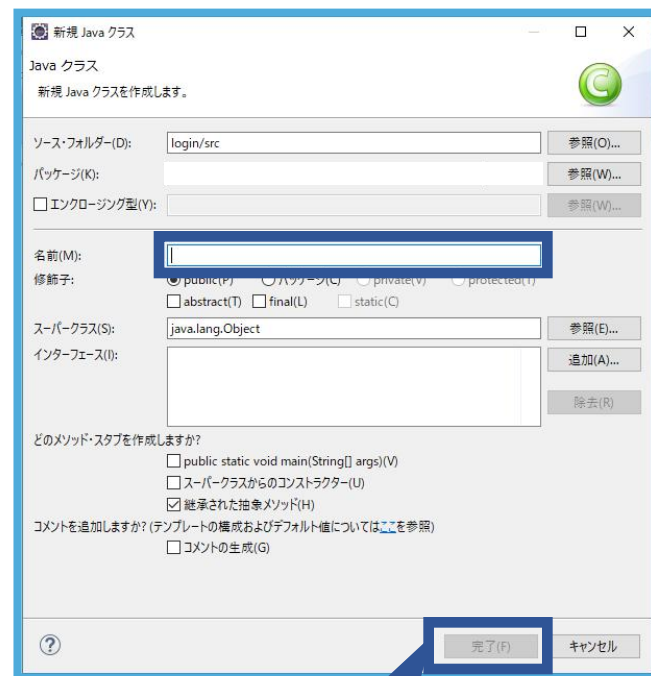
DBから取得した商品情報を、  
セッションに格納しています。

# BuyItemDAOの作成

## 2 DAOクラス



① 「com.internousdev.ecsite.dao」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「BuyItemDAO」を入力し、完了ボタンをクリックします。

# BuyItemDAOの作成

BuyItemDAO(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.ecsite.dao;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import com.internousdev.ecsite.dto.BuyItemDTO;
```

```
import com.internousdev.ecsite.util.DBConnector;
```

```
public class BuyItemDAO {
```

```
    private DBConnector dbConnector = new DBConnector();
```

```
    private Connection connection = dbConnector.getConnection();
```

```
    private BuyItemDTO buyItemDTO = new BuyItemDTO();
```

```
    public BuyItemDTO getBuyItemInfo() {
```

```
        String sql = "SELECT id, item_name, item_price FROM item_info_transaction";
```

商品情報をすべて取得する  
SQL文を書きます。

次へ続きます。

# BuyItemDAOの作成

前の続きです。

```
try {
    PreparedStatement preparedStatement = connection.prepareStatement(sql);
    ResultSet resultSet = preparedStatement.executeQuery();
    if(resultSet.next()) {
        buyItemDTO.setId(resultSet.getInt("id"));
        buyItemDTO.setItemName(resultSet.getString("item_name"));
        buyItemDTO.setItemPrice(resultSet.getString("item_price"));
    }
} catch(Exception e) {
    e.printStackTrace();
}
return buyItemDTO;
}

public BuyItemDTO getBuyItemDTO() {
    return buyItemDTO;
}
}
```

SQL文を実行します。

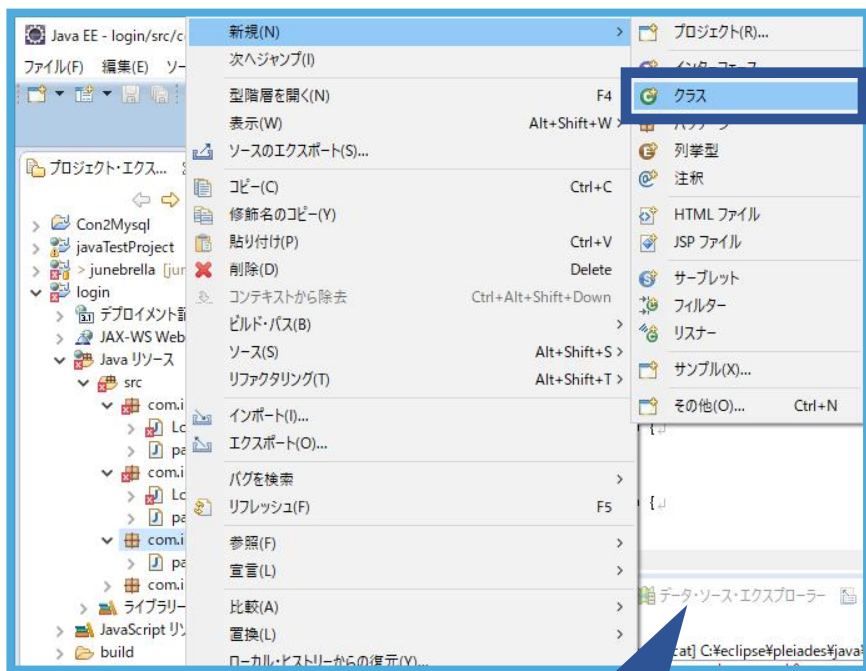
処理中にエラーが発生した際に  
実行する処理です。

Select文でDBから取得した情報を  
DTOクラスに格納します。

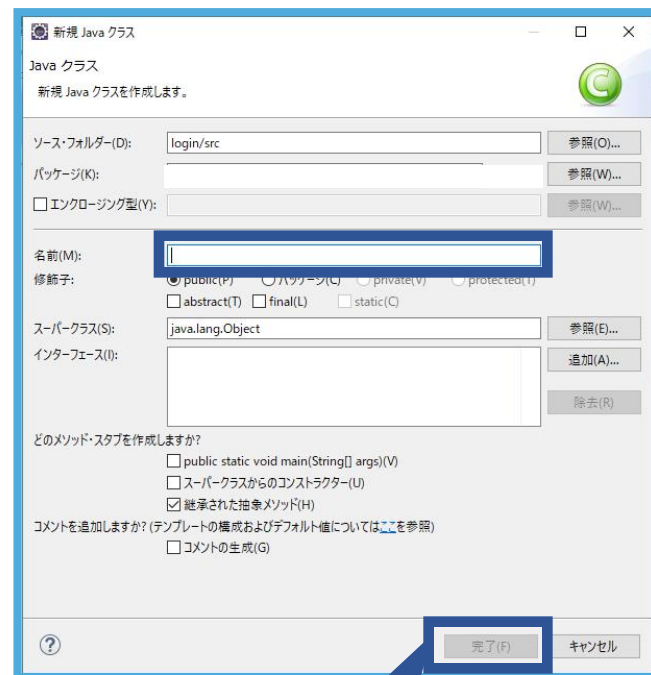
ActionクラスにDTOクラスを返します。

# BuyItemDTOの作成

## 3 DTOクラス



① 「com.internousdev.ecsite.dto」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「BuyItemDTO」を入力し、完了ボタンをクリックします。



# BuyItemDTOの作成

BuyItemDTO(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.ecsite.dto;
```

```
public class BuyItemDTO {  
    private int id;  
    private String itemName;  
    private String itemPrice;
```

テーブルカラムに対応したフィールド変数を宣言します。

```
    public String getItemName() {  
        return itemName;  
    }
```

フィールド変数に対応したGetter Setterを定義します。

```
    public void setItemName(String itemName) {  
        this.itemName = itemName;  
    }
```

```
    public String getItemPrice() {  
        return itemPrice;  
    }
```

次へ続きます。

# BuyItemDTOの作成

前の続きです。

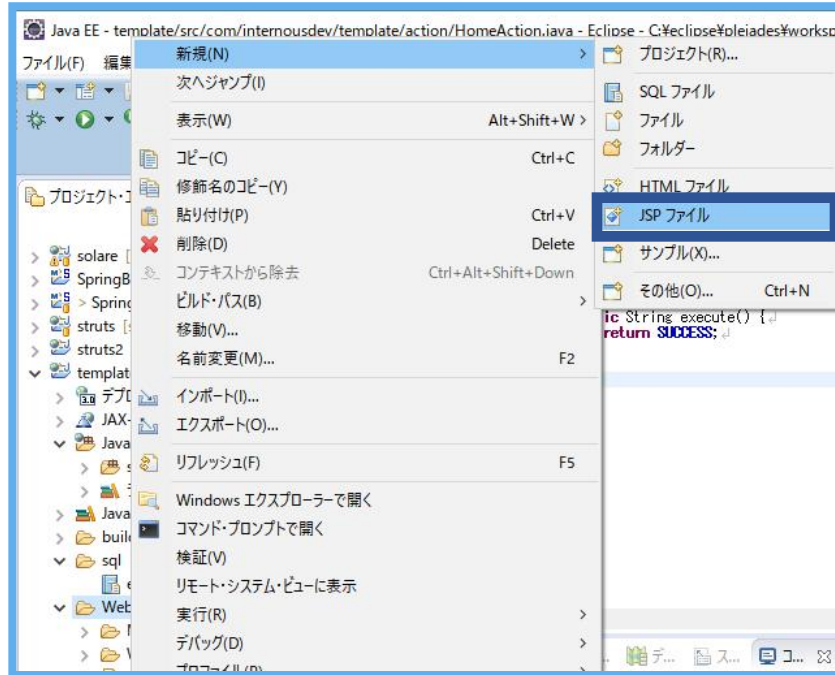
```
public void setItemPrice(String itemPrice) {  
    this.itemPrice = itemPrice;  
}
```

```
public int getId() {  
    return id;  
}
```

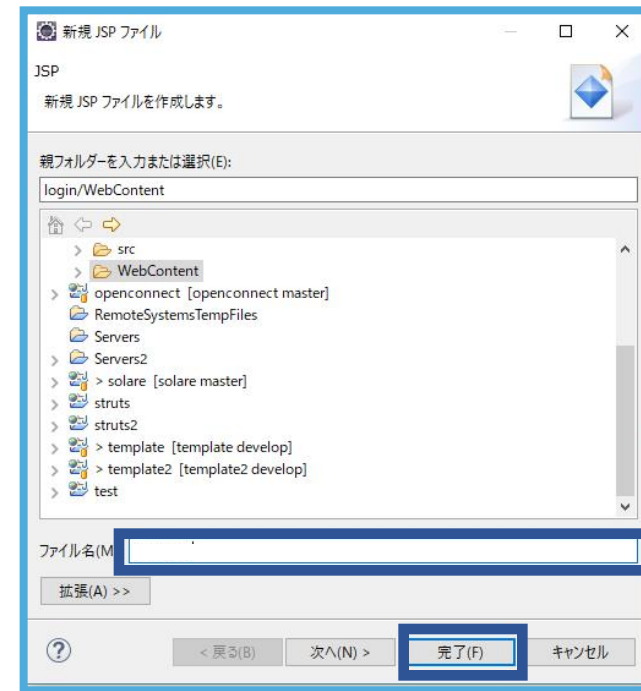
```
public void setId(int id) {  
    this.id = id;  
}
```

```
}
```

# buyItem.jspの作成



① 「プロジェクト」「WebContent」を選択し、「右クリック」「新規」「JSPファイル」を選択します。



② 「名前(M):」欄に「buyItem.jsp」を入力し、完了ボタンをクリックします。

# buyItem.jspの作成

buyItem.jsp(jspファイル)

③ 以下の内容を写経します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <meta http-equiv="imagetoolbar" content="no" />
    <meta name="description" content="" />
    <meta name="keywords" content="" />
    <title>BuyItem画面</title>
    <style type="text/css">
/* =====TAG LAYOUT===== */
```

次へ続きます。

前の続きです。

```
body {  
    margin:0;  
    padding:0;  
    line-height:1.6;  
    letter-spacing:1px;  
    font-family:Verdana, Helvetica, sans-serif;  
    font-size:12px;  
    color:#333;  
    background:#fff;  
}  
  
table {  
    text-align:center;  
    margin:0 auto;  
}  
  
/* =====ID LAYOUT===== */
```

次へ続きます。

前の続きです。

```
#top {  
  width:780px;  
  margin:30px auto;  
  border:1px solid #333;  
}  
  
#header {  
  width: 100%;  
  height: 80px;  
  background-color: black;  
}  
  
#main {  
  width: 100%;  
  height: 500px;  
  text-align: center;  
}
```

次へ続きます。

# buyItem.jspの作成

前の続きです。

```
#footer {  
    width: 100%;  
    height: 80px;  
    background-color: black;  
    clear: both;  
}  
</style>  
</head>  
<body>  
    <div id="header">  
        <div id="pr">  
        </div>  
    </div>  
    <div id="main">  
        <div id="top">  
            <p>BuyItem</p>  
        </div>
```

次へ続きます。

前の続きです。

```
<div>
<s:form action="BuyItemAction">
  <table>
    <tr>
      <td>
        <span>商品名</span>
      </td>
      <td>
        <s:property value="session.buyItem_name" /><br>
      </td>
    </tr>
  </table>
</div>
```

次へ続きます。



前の続きです。

```
<tr>
  <td>
    <span>値段</span>
  </td>
  <td>
    <s:property value="session.buyItem_price" /><span>円</span>
  </td>
</tr>
```

次へ続きます。

前の続きです。

```
<tr>
  <td>
    <span>購入個数</span>
  </td>
  <td>
    <select name="count">
      <option value="1" selected="selected">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
      <option value="5">5</option>
    </select>
  </td>
</tr>
```

次へ続きます。

# buyItem.jspの作成

前の続きです。

```
<tr>
  <td>
    <span>支払い方法</span>
  </td>
  <td>
    <input type="radio" name="pay" value="1" checked="checked">現金払い
    <input type="radio" name="pay" value="2">クレジットカード
  </td>
</tr>
<tr>
  <td>
    <s:submit value="購入"/>
  </td>
</tr>
</table>
</s:form>
```

次へ続きます。

# buyItem.jspの作成

前の続きです。

```
<div>
  <p>前画面に戻る場合は<a href='<s:url action="GoHomeAction" />'>こちら</a></p>
  <p>マイページは<a href='<s:url action="MyPageAction" />'>こちら</a></p>
</div>
</div>
</div>
<div id="footer">
  <div id="pr">
  </div>
</div>
</body>
</html>
```

# buyItem.jspの作成

## buyItem.jspの表示画面



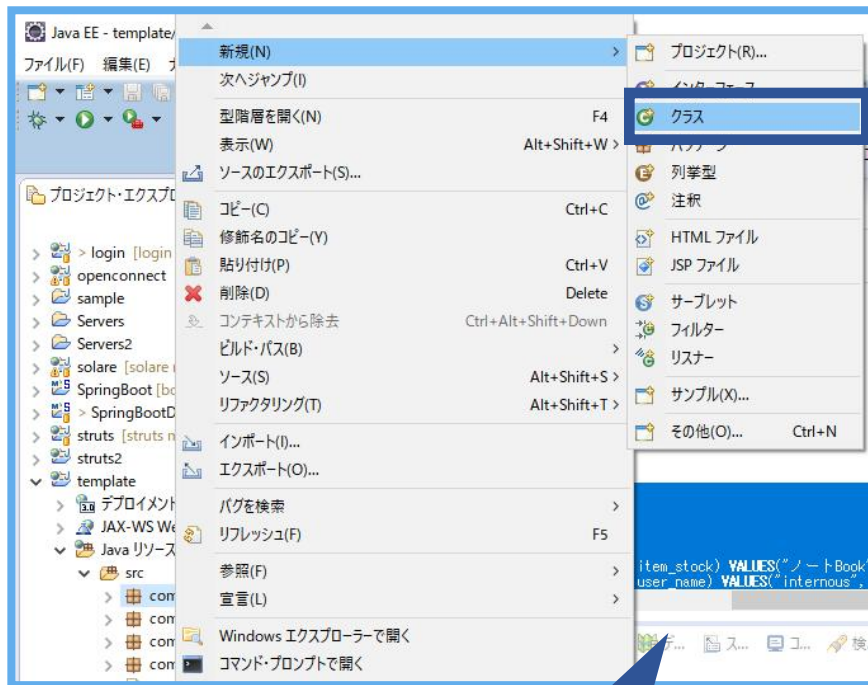
The screenshot displays a web page titled "BuyItem" within a black header bar. The page content is centered and includes the following elements:

- 商品名** (Product Name): Notebook
- 値段** (Price): 100円
- 購入回数** (Purchase Count): A dropdown menu showing "1".
- 支払い方法** (Payment Method): Radio buttons for ☒ 現金払い (Cash) and ☐ クレジットカード (Credit Card).
- 購入** (Purchase): A button to submit the form.
- 前画面に戻る場合は** (Return to previous screen): A link labeled [こちら](#).
- マイページは** (My page): A link labeled [こちら](#).

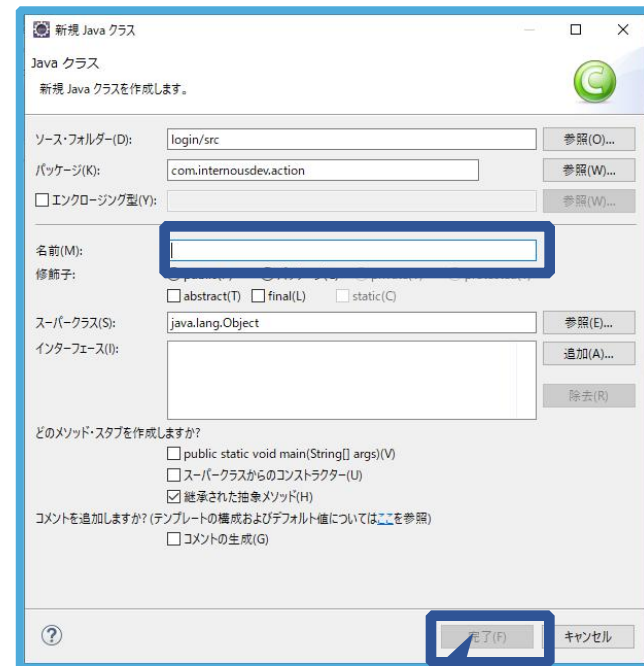
上記の画面が表示できれば成功です。

# BuyItemConfirmActionの作成

## 1 Actionクラス



① 「com.internousdev.template.action」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「BuyItemAction」を入力し、完了ボタンをクリックします。

# BuyItemActionの作成

BuyItemAction  
(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.ecsite.action;

import com.opensymphony.xwork2.ActionSupport;
import org.apache.struts2.interceptor.SessionAware;
import java.util.Map;

public class BuyItemAction extends ActionSupport implements SessionAware {
    public Map<String, Object> session;
    private int count;
    private String pay;
```

次へ続きます。

# BuyItemActionの作成

前の続きです。

```
public String execute() {  
    String result = SUCCESS;  
    session.put("count", count);  
    int intCount = Integer.parseInt(session.get("count").toString());  
    int intPrice = Integer.parseInt(session.get("buyItem_price").toString());  
    session.put("total_price", intCount * intPrice);  
    String payment;  
    if (pay.equals("1")) {  
        payment = "現金払い";  
        session.put("pay", payment);  
    } else {  
        payment = "クレジットカード";  
        session.put("pay", payment);  
    }  
    return result;  
}
```

次へ続きます。



# BuyItemActionの作成

前の続きです。

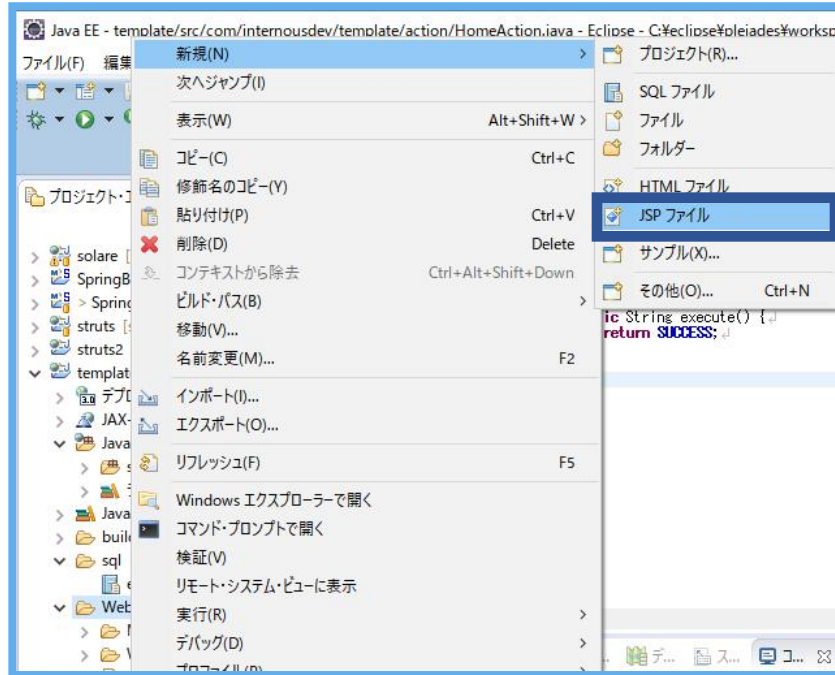
```
public void setCount(int count) {  
    this.count = count;  
}
```

```
public void setPay(String pay) {  
    this.pay = pay;  
}
```

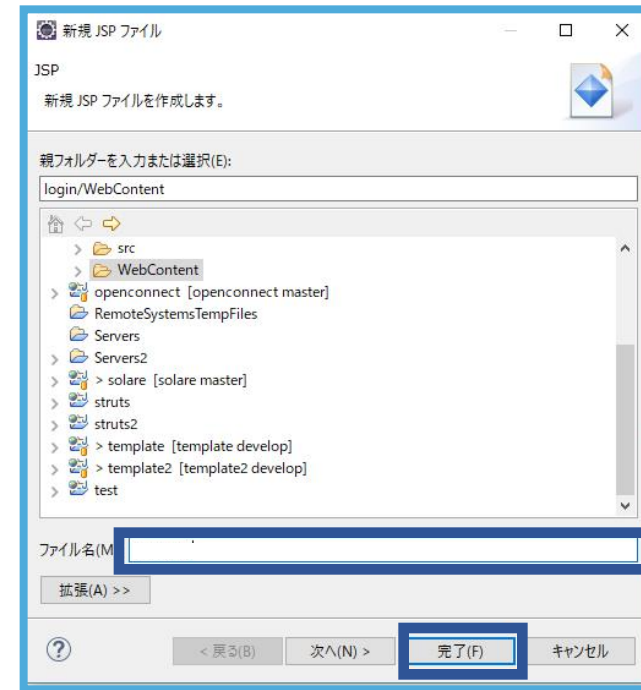
```
@Override  
public void setSession(Map<String, Object> session) {  
    this.session = session;  
}
```

```
}
```

# buyItemConfirm.jspの作成



① 「プロジェクト」「WebContent」を選択し、「右クリック」「新規」「JSPファイル」を選択します。



② 「名前(M):」欄に「buyItemConfirm.jsp」を入力し、完了ボタンをクリックします。

# buyItemConfirm.jspの作成

buyItemConfirm.jsp(jspファイル)

③ 以下の内容を写経します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <meta http-equiv="imagetoolbar" content="no" />
    <meta name="description" content="" />
    <meta name="keywords" content="" />
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <title>BuyItemConfirm画面</title>
    <style type="text/css">
/* =====TAG LAYOUT===== */
```

次へ続きます。

# buyItemConfirm.jspの作成

前の続きです。

```
body {  
    margin:0;  
    padding:0;  
    line-height:1.6;  
    letter-spacing:1px;  
    font-family:Verdana, Helvetica, sans-serif;  
    font-size:12px;  
    color:#333;  
    background:#fff;  
}  
  
table {  
    text-align:center;  
    margin:0 auto;  
}  
  
/* =====ID LAYOUT===== */
```

次へ続きます。

前の続きです。

```
#top {  
  width:780px;  
  margin:30px auto;  
  border:1px solid #333;  
}  
  
#header {  
  width: 100%;  
  height: 80px;  
  background-color: black;  
}  
  
#main {  
  width: 100%;  
  height: 500px;  
  text-align: center;  
}
```

次へ続きます。

# buyItemConfirm.jspの作成

前の続きです。

```
#footer {  
    width: 100%;  
    height: 80px;  
    background-color: black;  
    clear: both;  
}  
</style>  
<script type="text/javascript">  
    function submitAction(url) {  
        $('form').attr('action', url);  
        $('form').submit();  
    }  
</script>
```

次へ続きます。

# buyItemConfirm.jspの作成

前の続きです。

```
</head>
<body>
  <div id="header">
    <div id="pr">
    </div>
  </div>
  <div id="main">
    <div id="top">
      <p>BuyItem</p>
    </div>
    <div>
      <s:form>
```

次へ続きます。

# buyItemConfirm.jspの作成

前の続きです。

```
<tr>
    <td>商品名</td>
    <td><s:property value="session.buyItem_name" /></td>
</tr>
<tr>
    <td>値段</td>
    <td><s:property value="session.total_price" /><span>円</span></td>
</tr>
<tr>
    <td>購入個数</td>
    <td><s:property value="session.count" /><span>個</span></td>
</tr>
<tr>
    <td>支払い方法</td>
    <td><s:property value="session.pay" /></td>
</tr>
<tr>
    <td><br></td>
</tr>
```

次へ続きます。



# buyItemConfirm.jspの作成

前の続きです。

```

        <tr>
            <td><input type="button" value="戻る"
                onclick="submitAction('HomeAction')" /></td>
            <td><input type="button" value="完了"
                onclick="submitAction('BuyItemConfirmAction')" /></td>
        </tr>
    </s:form>
</div>
</div>
<div id="footer">
    <div id="pr">
    </div>
</div>
</body>
</html>
```

次へ続きます。

# buyItemConfirm.jspの作成

前の続きです。

```
<div>
  <p>前画面に戻る場合は<a href='<s:url action="GoHomeAction" />'>こちら</a></p>
  <p>マイページは<a href='<s:url action="MyPageAction" />'>こちら</a></p>
</div>
</div>
</div>
<div id="footer">
  <div id="pr">
  </div>
</div>
</body>
</html>
```

# buyItemConfirm.jspの作成

## buyItemConfirm.jspの画面

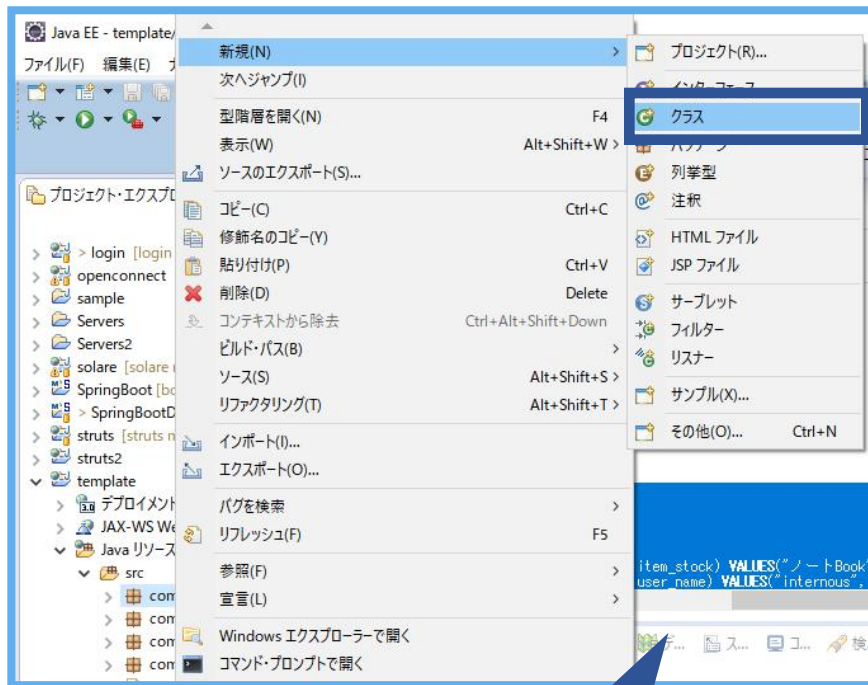


BuyItemConfirm	
商品名	NoteBook
値段	100円
購入個数	1個
支払い方法	現金払い
<input type="button" value="戻る"/>	<input type="button" value="完了"/>

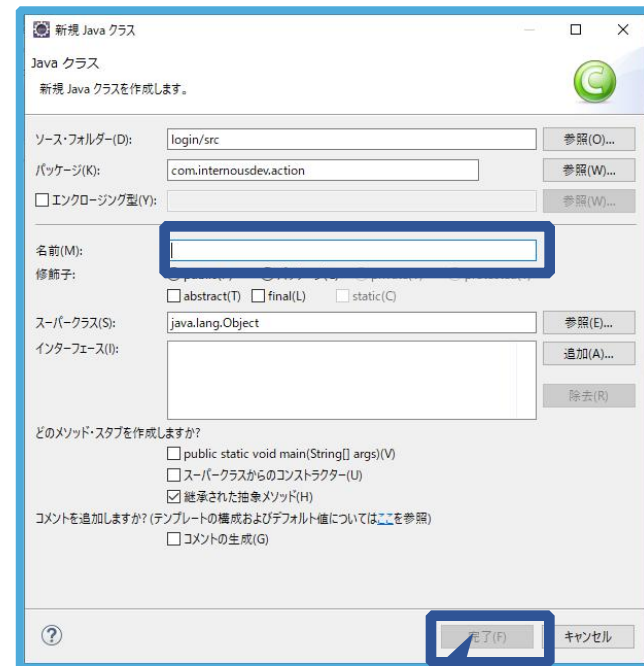
上記の画面が表示できれば成功です。

# BuyItemConfirmActionの作成

## 1 Actionクラス



① 「com.internousdev.template.action」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「BuyItemConfirmAction」を入力し、完了ボタンをクリックします。

# BuyItemConfirmActionの作成

BuyItemConfirmAction  
(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.ecsite.action;

import com.opensymphony.xwork2.ActionSupport;
import org.apache.struts2.interceptor.SessionAware;
import com.internousdev.ecsite.dao.BuyItemCompleteDAO;
import java.sql.SQLException;
import java.util.Map;

public class BuyItemConfirmAction extends ActionSupport implements SessionAware {
    public Map<String, Object> session;
    private BuyItemCompleteDAO buyItemCompleteDAO = new BuyItemCompleteDAO();
}
```

次へ続きます。

# BuyItemConfirmActionの作成

前の続きです。

```
public String execute() throws SQLException {
    buyItemCompleteDAO.buyItemInfo(
        session.get("id").toString(),
        session.get("login_user_id").toString(),
        session.get("total_price").toString(),
        session.get("count").toString(),
        session.get("pay").toString());

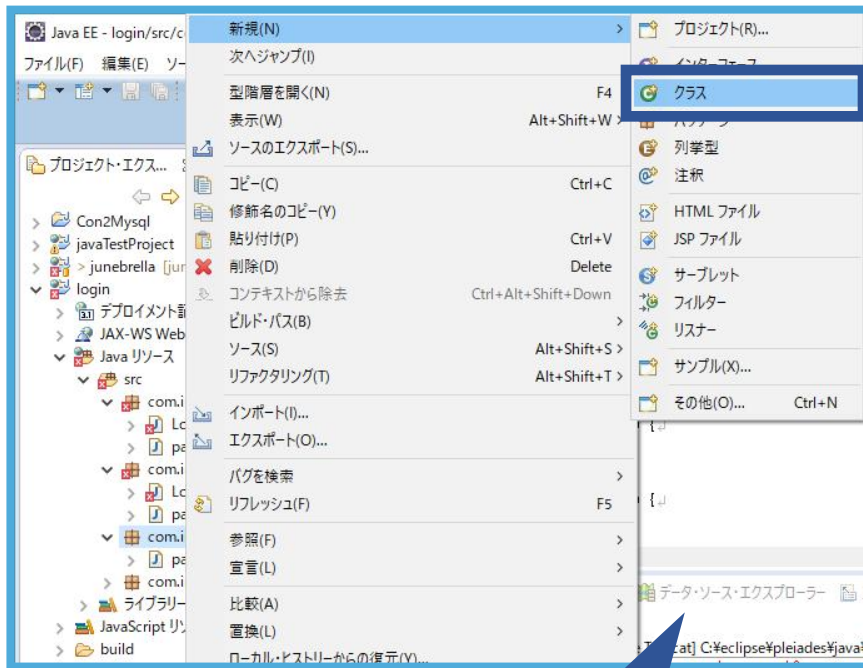
    String result = SUCCESS;
    return result;
}

@Override
public void setSession(Map<String, Object> session) {
    this.session = session;
}
}
```

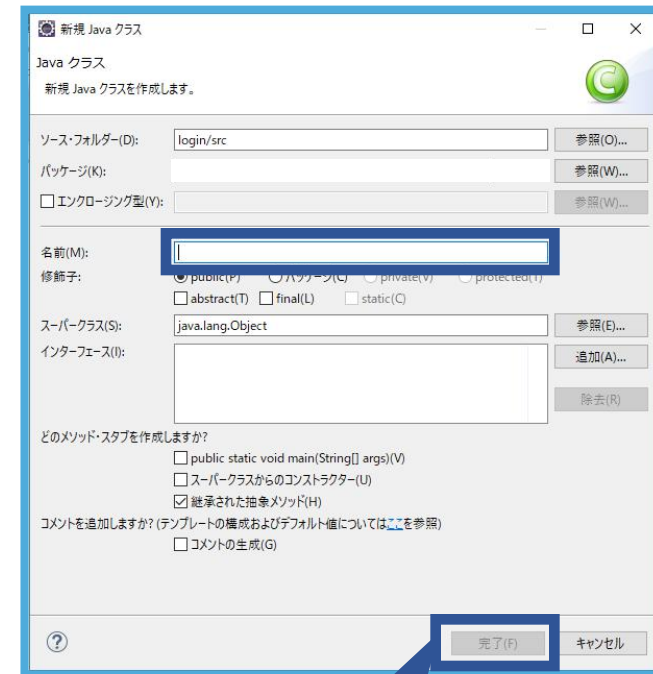
セッションから取得した商品情報を、  
DAOのbuyItemInfoメソッドに  
渡しています。

# BuyItemCompleteDAOの作成

## 2 DAOクラス



① 「com.internousdev.ecsite.dao」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「BuyItemCompleteDAO」を入力し、完了ボタンをクリックします。

# BuyItemCompleteDAOの作成

BuyItemCompleteDAO  
(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.template.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import com.internousdev.ecsite.util.DBConnector;
import com.internousdev.ecsite.util.DateUtil;

public class BuyItemCompleteDAO {
    private DBConnector dbConnector = new DBConnector();
    private Connection connection = dbConnector.getConnection();
    private DateUtil dateUtil = new DateUtil();
    private String sql = "INSERT INTO user_buy_item_transaction
        (item_transaction_id, total_price, total_count, user_master_id, pay, insert_date)
        VALUES(?, ?, ?, ?, ?, ?)";
```

次へ続きます。



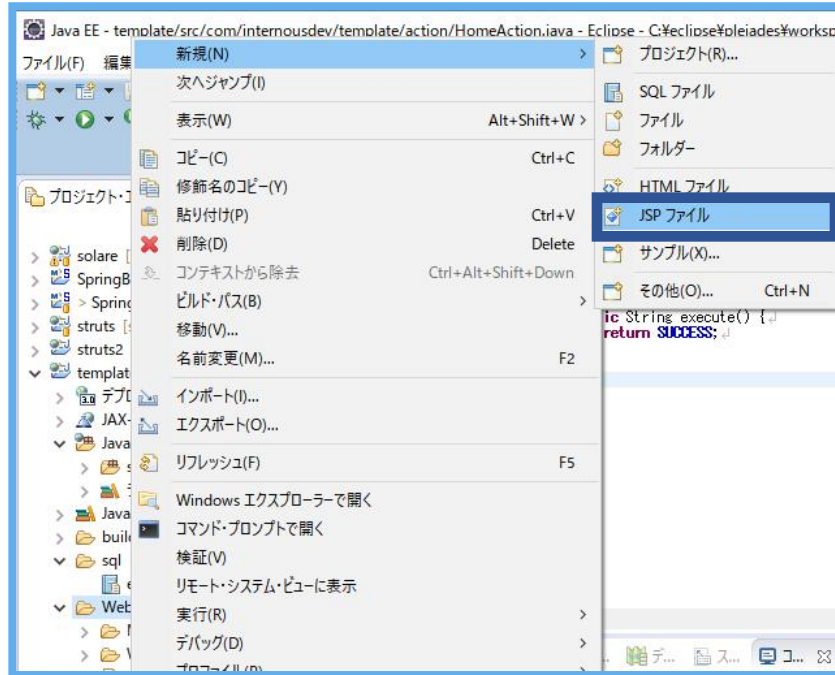
# BuyItemCompleteDAOの作成

前の続きです。

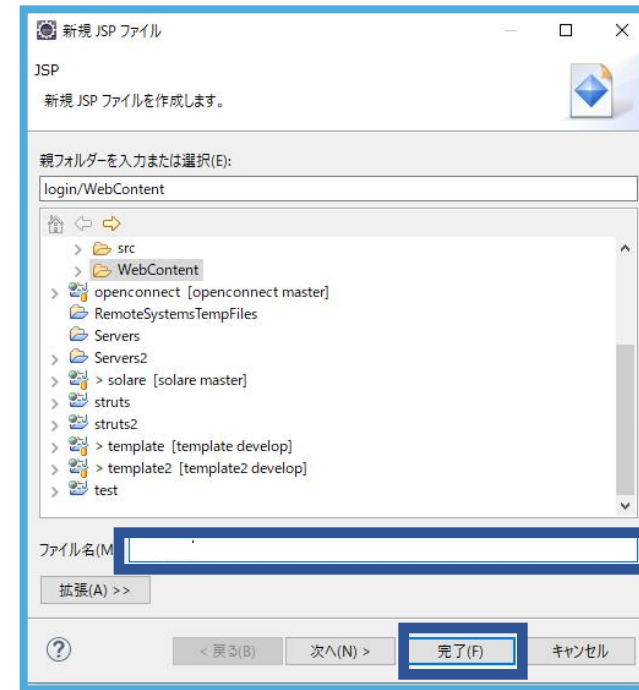
```
public void buyItemInfo(String item_transaction_id, String user_master_id,
    String total_price, String total_count, String pay) throws SQLException {
    try {
        PreparedStatement preparedStatement = connection.prepareStatement(sql);
        preparedStatement.setString(1, item_transaction_id);
        preparedStatement.setString(2, total_price);
        preparedStatement.setString(3, total_count);
        preparedStatement.setString(4, user_master_id);
        preparedStatement.setString(5, pay);
        preparedStatement.setString(6, dateUtil.getDate());
        preparedStatement.execute();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        connection.close();
    }
}
```

現在日時を取得しています。

# buyItemComplete.jspの作成



① 「プロジェクト」「WebContent」を選択し、「右クリック」「新規」「JSPファイル」を選択します。



② 「名前(M):」欄に「buyItemComplete.jsp」を入力し、完了ボタンをクリックします。

# buyItemComplete.jspの作成

BuyItemComplete.jsp  
(jspファイル)

③ 以下の内容を写経します。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <meta http-equiv="imagetoolbar" content="no" />
    <meta name="description" content="" />
    <meta name="keywords" content="" />
    <title>BuyItemComplete画面</title>
    <style type="text/css">
/* =====TAG LAYOUT===== */
```

次へ続きます。

# buyItemComplete.jspの作成

前の続きです。

```
body {  
    margin:0;  
    padding:0;  
    line-height:1.6;  
    letter-spacing:1px;  
    font-family:Verdana, Helvetica, sans-serif;  
    font-size:12px;  
    color:#333;  
    background:#fff;  
}  
  
table {  
    text-align:center;  
    margin:0 auto;  
}  
  
/* =====ID LAYOUT===== */
```

次へ続きます。

# buyItemComplete.jspの作成

前の続きです。

```
#top {  
  width:780px;  
  margin:30px auto;  
  border:1px solid #333;  
}  
  
#header {  
  width: 100%;  
  height: 80px;  
  background-color: black;  
}  
  
#main {  
  width: 100%;  
  height: 500px;  
  text-align: center;  
}
```

次へ続きます。

# buyItemComplete.jspの作成

前の続きです。

```
#footer {  
    width: 100%;  
    height: 80px;  
    background-color: black;  
    clear: both;  
}  
</style>  
</head>  
<body>  
    <div id="header">  
        <div id="pr">  
        </div>  
    </div>  
    <div id="main">  
        <div id="top">  
            <p>BuyItemComplete</p>  
        </div>
```

次へ続きます。

# buyItemComplete.jspの作成

前の続きです。

```
<div>
  <p>購入手続きが完了致しました。</p>
  <div>
    <a href='<s:url action="MyPageAction" />'>
      マイページ</a><span>から購入履歴の確認が可能です。</span>
    <p>Homeへ戻る場合は<a href='<s:url action="GoHomeAction" />'>
      こちら</a></p>
  </div>
</div>
</div>
<div id="footer">
  <div id="pr">
  </div>
</div>
</body>
</html>
```

# buyItemComplete.jspの作成

## buyItemComplete.jspの画面



上記の画面が表示できれば成功です。