

---

# ECサイト中級

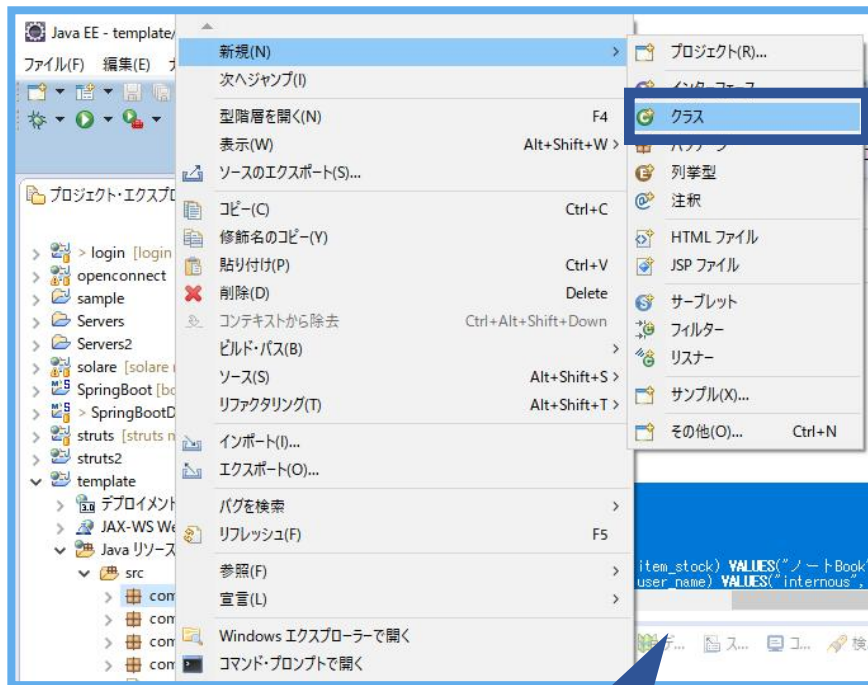
購入履歴機能の作成

---

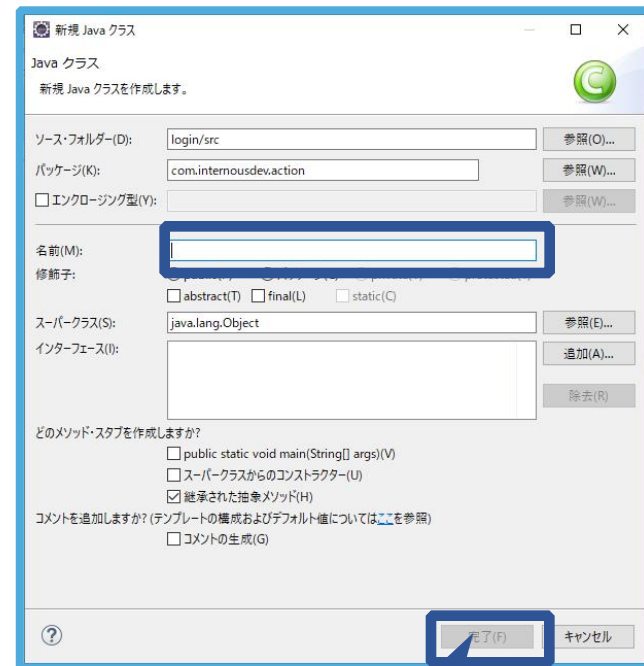
## 6 時間目

# MyPageActionの作成

## 1 Actionクラス



① 「com.internousdev.ecsite.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「MyPageAction」を入力し、完了ボタンをクリックします。

# MyPageActionの作成

MyPageAction(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.sql.SQLException;
import java.util.ArrayList;
import java.util.Map;
import com.internousdev.ecsite.dao.MyPageDAO;
import com.internousdev.ecsite.dto.MyPageDTO;

public class MyPageAction extends ActionSupport implements SessionAware {
    public Map<String, Object> session;
    private MyPageDAO myPageDAO = new MyPageDAO();
    private ArrayList<MyPageDTO> myPageList = new ArrayList<MyPageDTO>();
    private String deleteFlg;
    private String message;
```

次へ続きます。

# MyPageActionの作成

前の続きです。

```
public String execute() throws SQLException {  
    if (!session.containsKey("id")) {  
        return ERROR;  
    }  
  
    if(deleteFlg == null) {  
        String item_transaction_id = session.get("id").toString();  
        String user_master_id = session.get("login_user_id").toString();  
        myPageList = myPageDAO.getMyPageUserInfo(item_transaction_id, user_master_id);  
    } else if(deleteFlg.equals("1")) {  
        delete();  
    }  
  
    String result = SUCCESS;  
    return result;  
}
```

履歴の削除がされているか否か、  
チェックをしています。

「delete」メソッドを呼び出して、  
履歴の削除処理を行います。

DBから取得した履歴情報を、  
「myPageList」に格納しています。

次へ続きます。

# MyPageActionの作成

前の続きです。

```
public void delete() throws SQLException {
```

履歴の削除を行うためのメソッドです。

```
    String item_transaction_id = session.get("id").toString();  
    String user_master_id = session.get("login_user_id").toString();
```

```
    int res = myPageDAO.buyItemHistoryDelete(item_transaction_id, user_master_id);
```

```
    if(res > 0) {  
        myPageList = null;  
        setMessage("商品情報を正しく削除しました。");  
    } else if(res == 0) {  
        setMessage("商品情報の削除に失敗しました。");  
    }  
}
```

DBから削除した履歴情報の件数を、「res」に格納しています。

1件以上削除されたか否かで正常に削除処理がされたか判別しています。

次へ続きます。

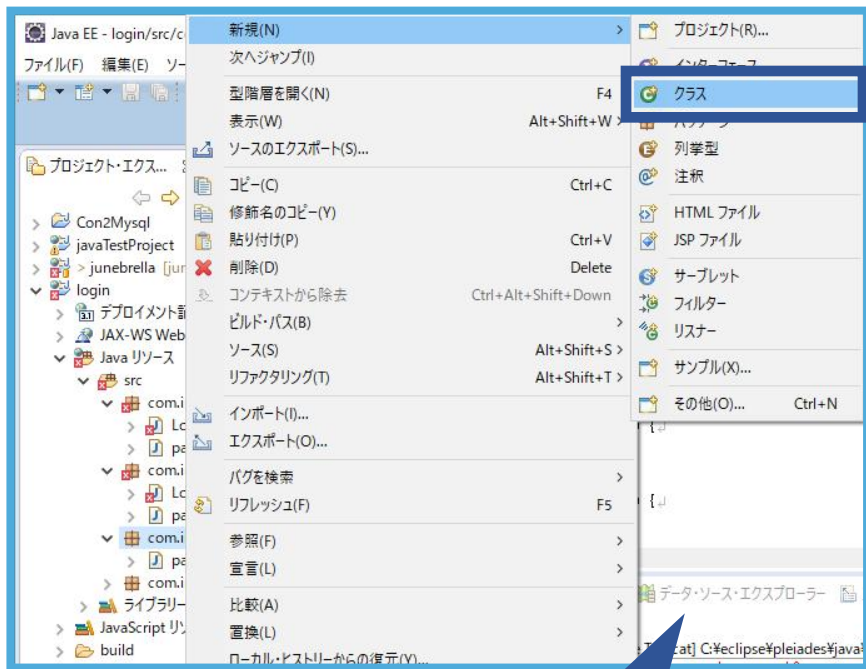
# MyPageActionの作成

前の続きです。

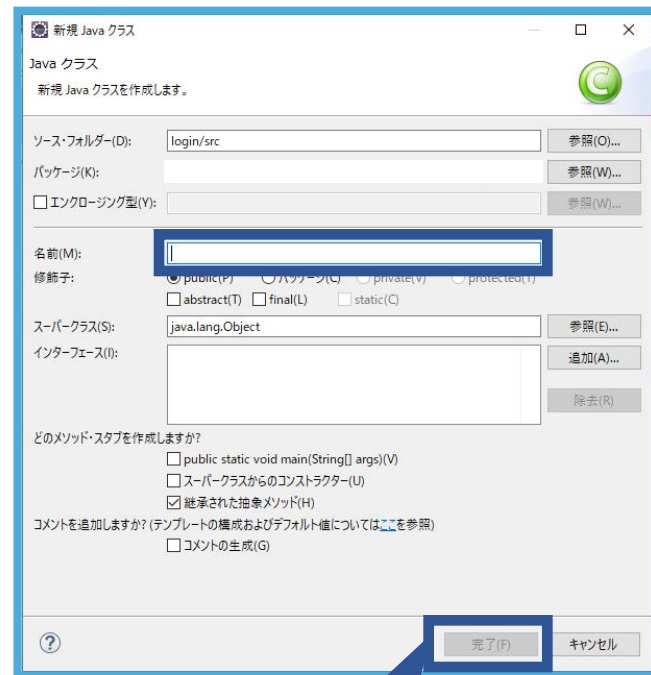
```
public void setDeleteFlg(String deleteFlg) {
    this.deleteFlg = deleteFlg;
}
@Override
public void setSession(Map<String, Object> session) {
    this.session = session;
}
public ArrayList<MyPageDTO> getMyPageList() {
    return this.myPageList;
}
public String getMessage() {
    return this.message;
}
public void setMessage(String message) {
    this.message = message;
}
}
```

# MyPageDAO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「MyPageDAO」を入力し、完了ボタンをクリックします。

# MyPageDAOの作成

MyPageDAO(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 java.sql.SQLException;
import java.util.ArrayList;
import com.internousdev.ecsite.dto.MyPageDTO;
import com.internousdev.ecsite.util.DBConnector;

public class MyPageDAO {

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

次へ続きます。



# MyPageDAOの作成

前の続きです。

```
public ArrayList<MyPageDTO> getMyPageUserInfo
(String item_transaction_id, String user_master_id) throws SQLException {
    ArrayList<MyPageDTO> myPageDTO = new ArrayList<MyPageDTO>();
    String sql =
        "SELECT
            ubit.id, iit.item_name, ubit.total_price, ubit.total_count,
            ubit.pay, ubit.insert_date
        FROM
            user_buy_item_transaction ubit
        LEFT JOIN
            item_info_transaction iit
        ON
            ubit.item_transaction_id = iit.id
        WHERE
            ubit.item_transaction_id = ? AND ubit.user_master_id = ?
        ORDER BY
            insert_date DESC";
```

DBから  
購入履歴を  
取得するための  
メソッドです。

「LEFT JOIN」を用いて複数のテーブル  
を結合することによってユーザ情報と履  
歴情報を紐付けて一括して取得すること  
ができます。

次へ続きます。

前の続きです。

```
try {  
    PreparedStatement preparedStatement = connection.prepareStatement(sql);  
    preparedStatement.setString(1, item_transaction_id);  
    preparedStatement.setString(2, user_master_id);  
  
    ResultSet resultSet = preparedStatement.executeQuery();  
  
    while(resultSet.next()) {  
        MyPageDTO dto = new MyPageDTO();  
        dto.setId(resultSet.getString("id"));  
        dto.setItemName(resultSet.getString("item_name"));  
        dto.setTotalPrice(resultSet.getString("total_price"));  
        dto.setTotalCount(resultSet.getString("total_count"));  
        dto.setPayment(resultSet.getString("pay"));  
        dto.setInsert_date(resultSet.getString("insert_date"));  
        myPageDTO.add(dto);  
    }  
}
```

取得した結果を1件ずつ  
DTOに格納し、更にDTOを  
ArrayListに格納しています。

次へ続きます。

前の続きです。

```
    } catch(Exception e) {  
        e.printStackTrace();  
    } finally {  
        connection.close();  
    }  
  
    return myPageDTO;  
}  
  
public int buyItemHistoryDelete  
    (String item_transaction_id, String user_master_id) throws SQLException {
```

DBから  
購入履歴を  
削除するための  
メソッドです。

次へ続きます。

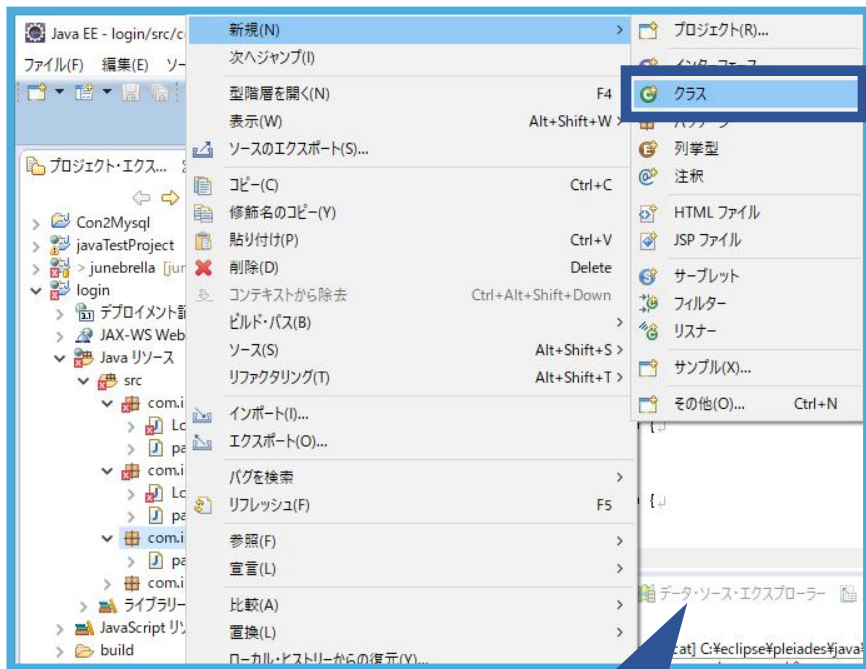
前の続きです。

```
String sql =
    "DELETE FROM
      user_buy_item_transaction
    WHERE
      item_transaction_id = ? AND user_master_id = ?";
PreparedStatement preparedStatement;
int result = 0;
try {
    preparedStatement = connection.prepareStatement(sql);
    preparedStatement.setString(1, item_transaction_id);
    preparedStatement.setString(2, user_master_id);
    result = preparedStatement.executeUpdate();
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    connection.close();
}
return result;
}
```

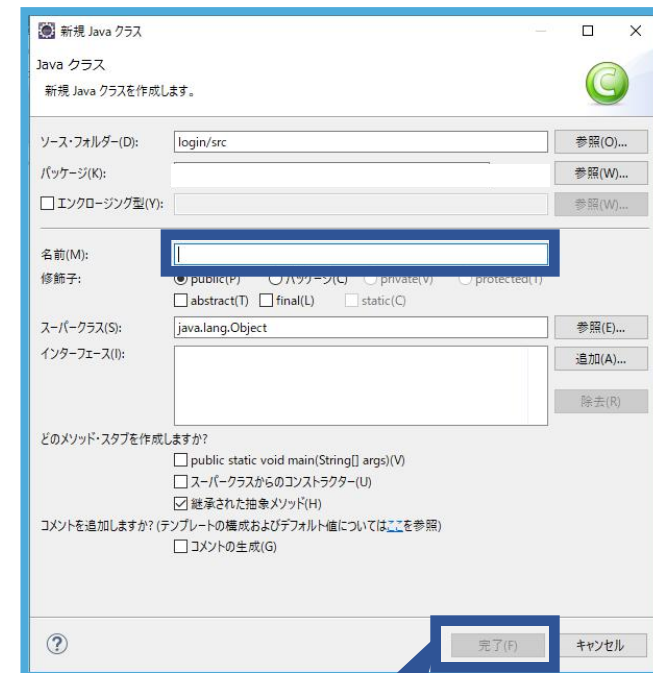
Actionクラスに削除した件数を返します。

# MyPageDTO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「MyPageDTO」を入力し、完了ボタンをクリックします。

# MyPageDTOの作成

MyPageDTO(javaファイル)

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

```
public class MyPageDTO {  
    public String itemName;  
  
    public String totalPrice;  
  
    public String totalCount;  
  
    public String payment;  
  
    public String userName;  
  
    public String id;  
  
    public String insert_date;
```

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。

③ 以下の内容を写経します。

次へ続きます。

# MyPageDTOの作成

前の続きです。

```
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 getTotalPrice() {  
    return totalPrice;  
}
```

```
public void setTotalPrice(String totalPrice) {  
    this.totalPrice = totalPrice;  
}
```

次へ続きます。

前の続きです。

```
public String getTotalCount() {  
    return totalCount;  
}  
  
public void setTotalCount(String totalCount) {  
    this.totalCount = totalCount;  
}  
  
public String getPayment() {  
    return payment;  
}  
  
public void setPayment(String payment) {  
    this.payment = payment;  
}
```

次へ続きます。



前の続きです。

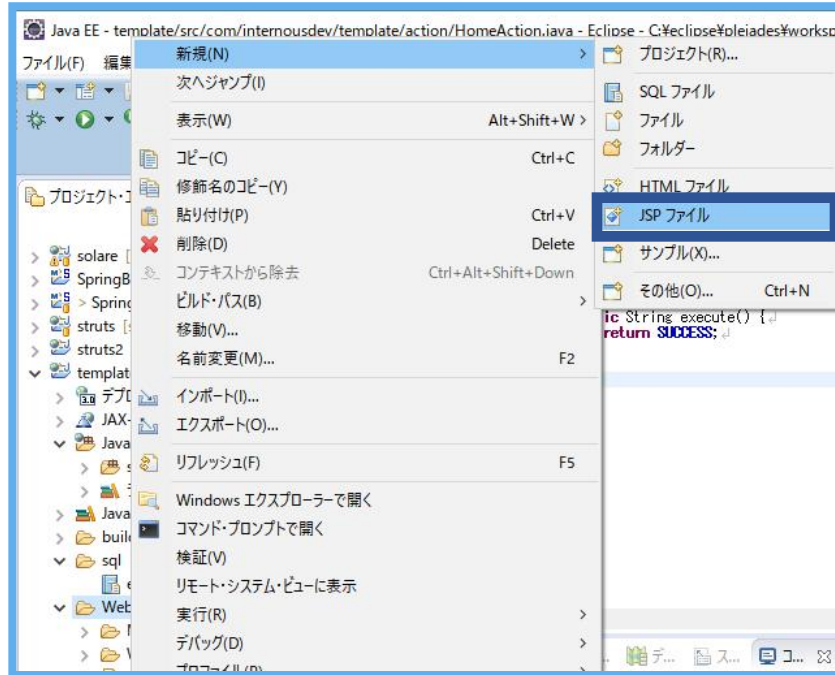
```
public String getUserName() {  
    return userName;  
}  
  
public void setUserName(String userName) {  
    this.userName = userName;  
}  
  
public String getId() {  
    return id;  
}  
  
public void setId(String id) {  
    this.id = id;  
}
```

次へ続きます。

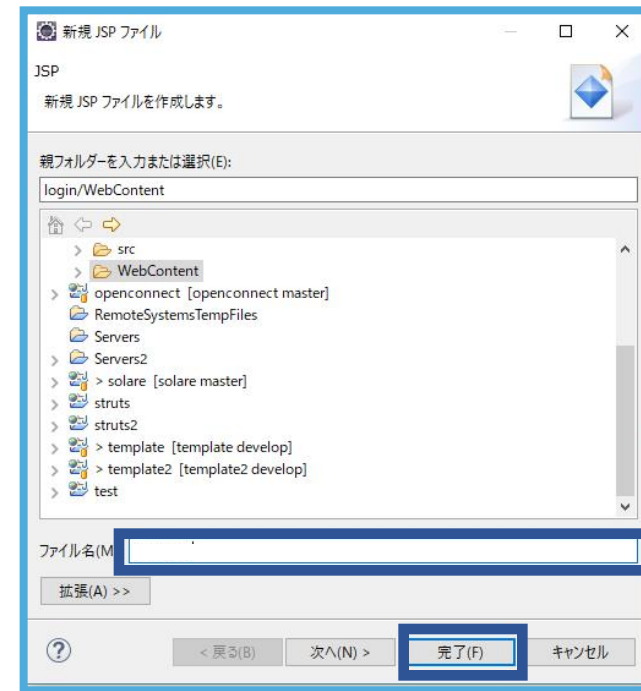
前の続きです。

```
public String getInsert_date() {  
    return insert_date;  
}  
  
public void setInsert_date(String insert_date) {  
    this.insert_date = insert_date;  
}  
  
}
```

# myPage.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「myPage.jsp」を入力し、完了ボタンをクリックします。

# myPage.jspの作成

MyPage.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>MyPage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。

前の続きです。

```
#footer {  
    width: 100%;  
    height: 80px;  
    background-color: black;  
    clear: both;  
}  
  
#text-right {  
    display: inline-block;  
    text-align: right;  
}  
  
</style>  
</head>  
<body>  
    <div id="header">  
        <div id="pr">  
            </div>  
    </div>
```

次へ続きます。

前の続きです。

```
<div id="main">
  <div id="top">
    <p>MyPage</p>
  </div>
  <div>
    <s:if test="myPageList == null">
      <h3>ご購入情報はありません。 </h3>
    </s:if>
    <s:elseif test="message == null">
      <h3>ご購入情報は以下になります。 </h3>
      <table border="1">
        <tr>
          <th>商品名</th>
          <th>値段</th>
          <th>購入個数</th>
          <th>支払い方法</th>
          <th>購入日</th>
        </tr>
```

次へ続きます。



前の続きです。

```
<s:iterator value="myPageList">
  <tr>
    <td><s:property value="itemName" /></td>
    <td><s:property value="totalPrice" /><span>円</span></td>
    <td><s:property value="totalCount" /><span>個</span></td>
    <td><s:property value="payment" /></td>
    <td><s:property value="insert_date" /></td>
  </tr>
</s:iterator>
</table>
<s:form action="MyPageAction">
  <input type="hidden" name="deleteFlg" value="1">
  <s:submit value="削除" method="delete" />
</s:form>
</s:elseif>
<s:if test="message != null">
  <h3><s:property value="message"/></h3>
</s:if>
```

次へ続きます。

前の続きです。

```
<div id="text-right">
  <p>Homeへ戻る場合は<a href='<s:url action="GoHomeAction" />'>こちら</a></p>
  <p>ログアウトする場合は<a href='<s:url action="LogoutAction" />'>こちら</a></p>
</div>
</div>
</div>
<div id="footer">
  <div id="pr">
  </div>
</div>
</body>
</html>
```

# myPage.jspの作成

## MyPage.jspの画面

MyPage

ご購入情報は以下になります。

商品名	値段	購入個数	支払い方法	購入日
NoteBook	100円	1個	現金払い	2017-11-15 21:44:40.0
NoteBook	500円	5個	現金払い	2017-11-15 11:19:40.0

削除

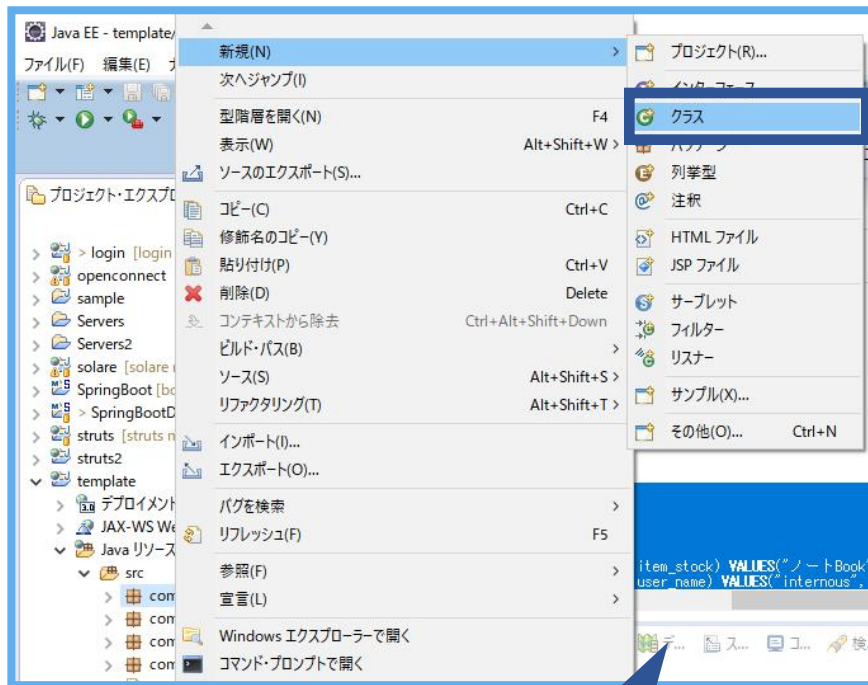
Homeへ戻る場合は[こちら](#)

ログアウトする場合は[こちら](#)

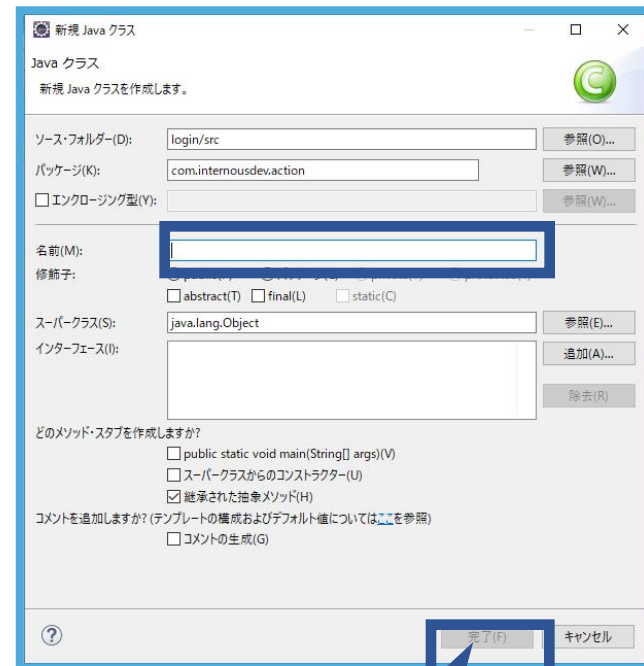
上記の画面が表示できれば成功です。

# LogoutAction.javaの作成

## 3 LogoutAction.java



① 「com.internousdev.ecsite.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「LogoutAction」を入力し、完了ボタンをクリックします。

# LogoutAction.javaの作成

③ 以下の内容を写経します。

LogoutAction.java  
(javaファイル)

```
package com.internousdev.ecsite.action;
import java.util.Map;
import org.apache.struts2.interceptor.SessionAware;
import com.opensymphony.xwork2.ActionSupport;
public class LogoutAction extends ActionSupport implements SessionAware {

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