

---

# ECサイト中級

ログイン認証機能の作成

---

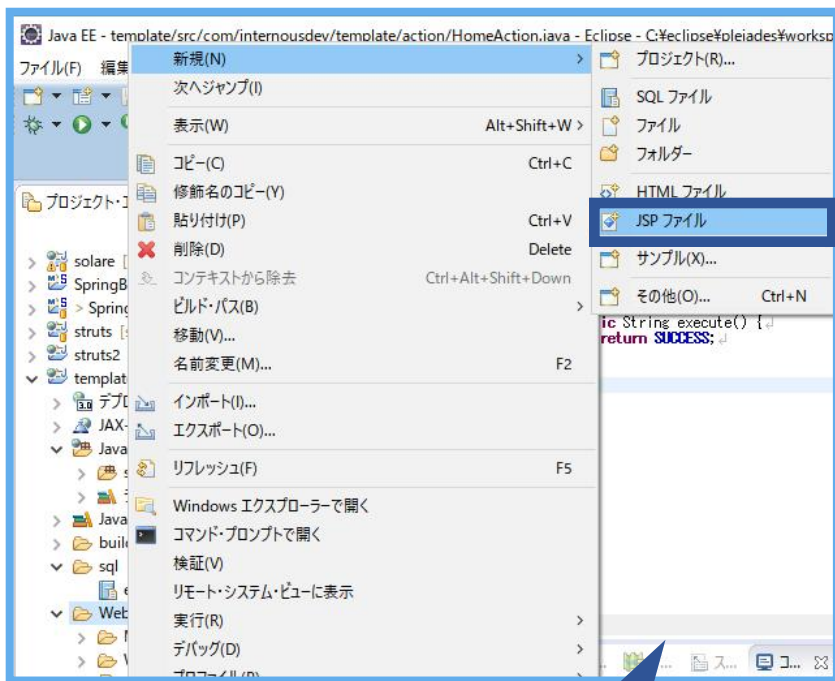
## 3 時間目

## 作成機能一覧

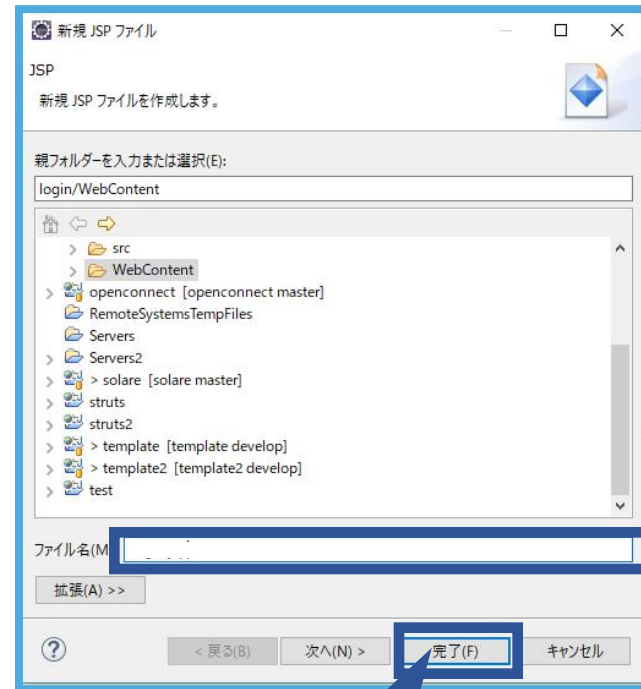
1. ログイン認証機能
2. ユーザー登録機能
3. 商品購入機能
4. 購入履歴機能

# JSPファイルの作成

## 1 home.jsp



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「home.jsp」を入力し、完了ボタンをクリックします。

# home.jspの作成

③ 以下の内容を写経します。

home.jsp(jspファイル)

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

<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="./css/style.css">

<title>Home画面</title>

</head>
```

次へ続きます。

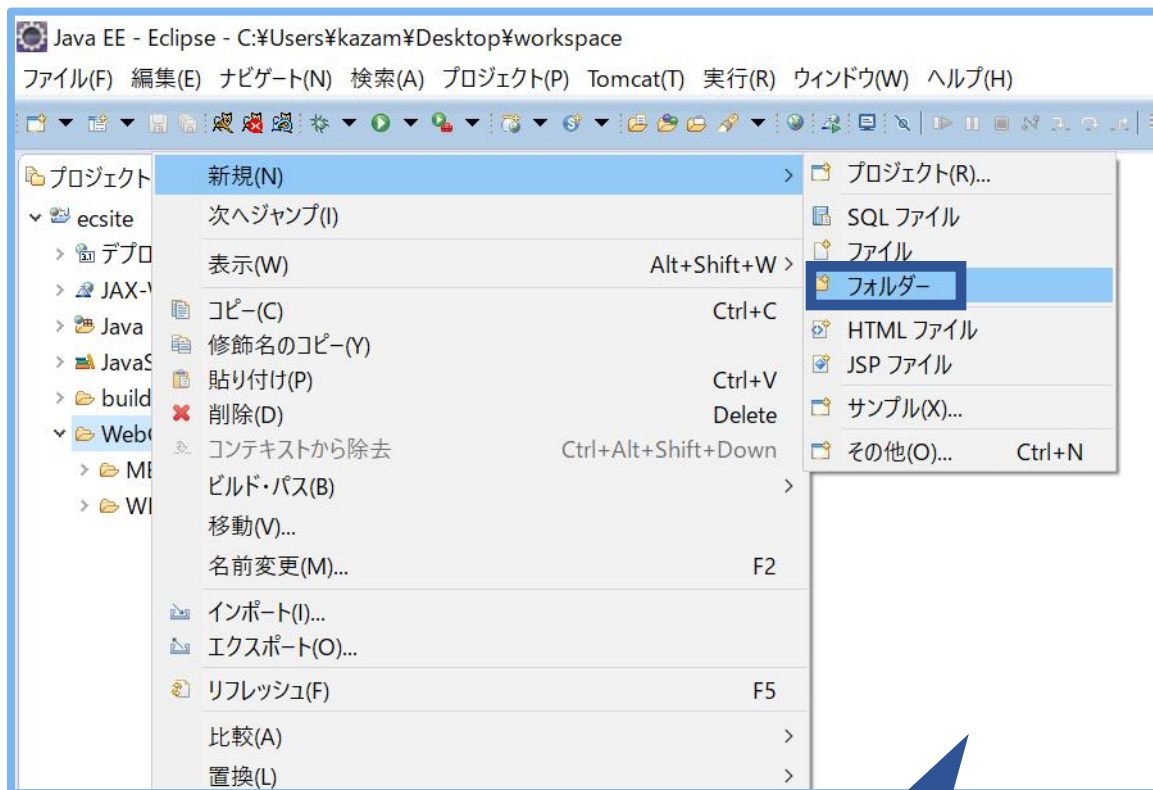
前の続きです。

```
<body>
  <div id="header">
  </div>
  <div id="main">
    <div id="top">
      <p>Home</p>
    </div>
    <div id="text-center">
      <s:form action="HomeAction">
        <s:submit value="商品購入"/>
      </s:form>
      <s:if test="#session.login_user_id != null">
        <p>ログアウトする場合は
          <a href='<s:url action="LogoutAction" />'>こちら</a></p>
        </s:if>
      </div>
    </div>
  </div>
  <div id="footer">
  </div>
</body>
</html>
```

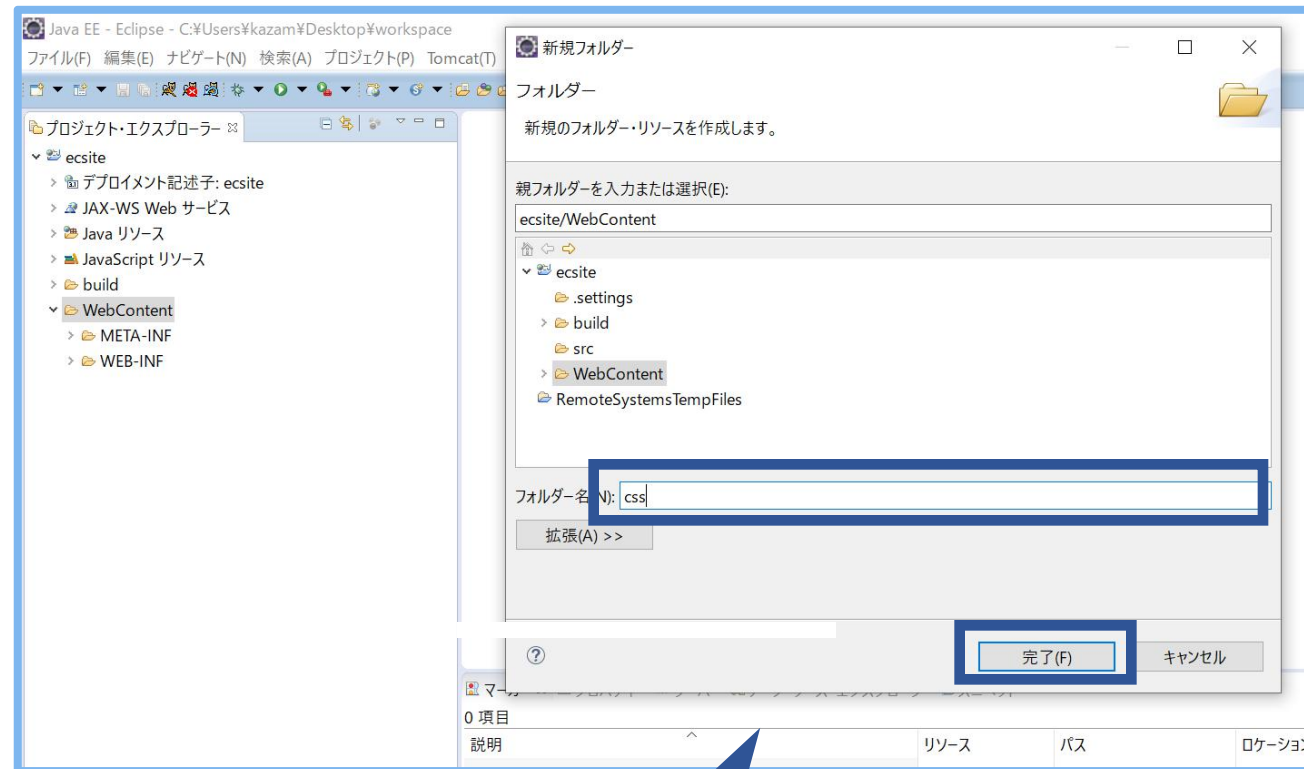
次へ続きます。

# style.cssの作成

## 1 CSSフォルダの作成

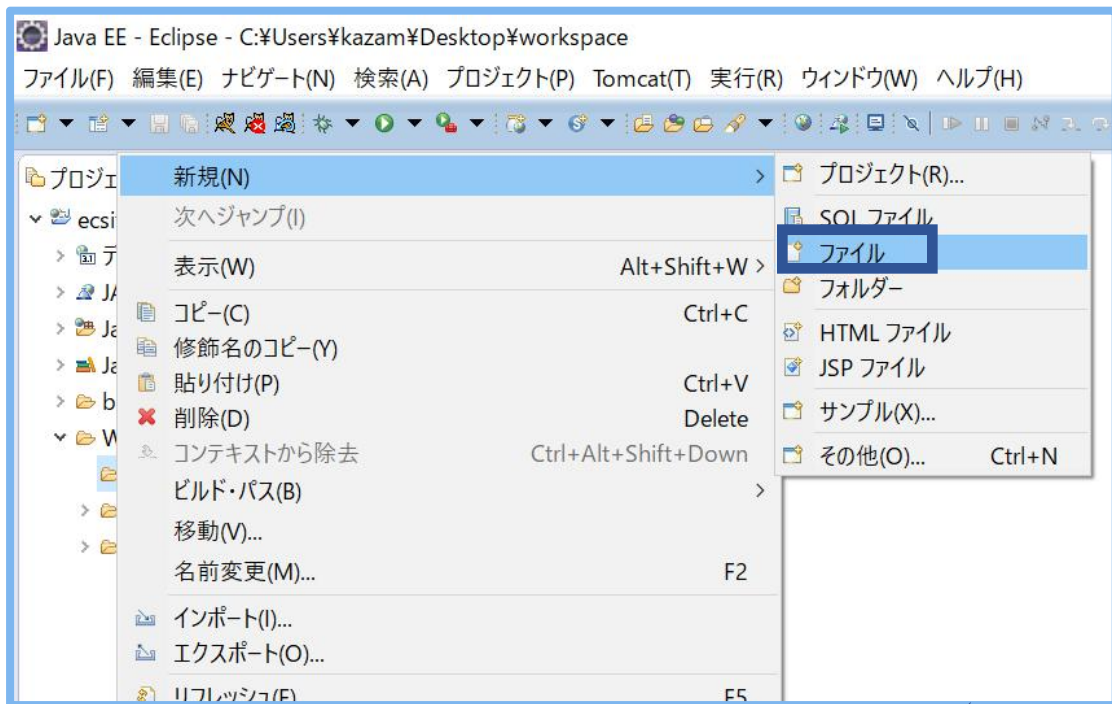


① 「プロジェクト」「WebContent」を選択し、  
「右クリック」「新規」「フォルダー」を選択します。

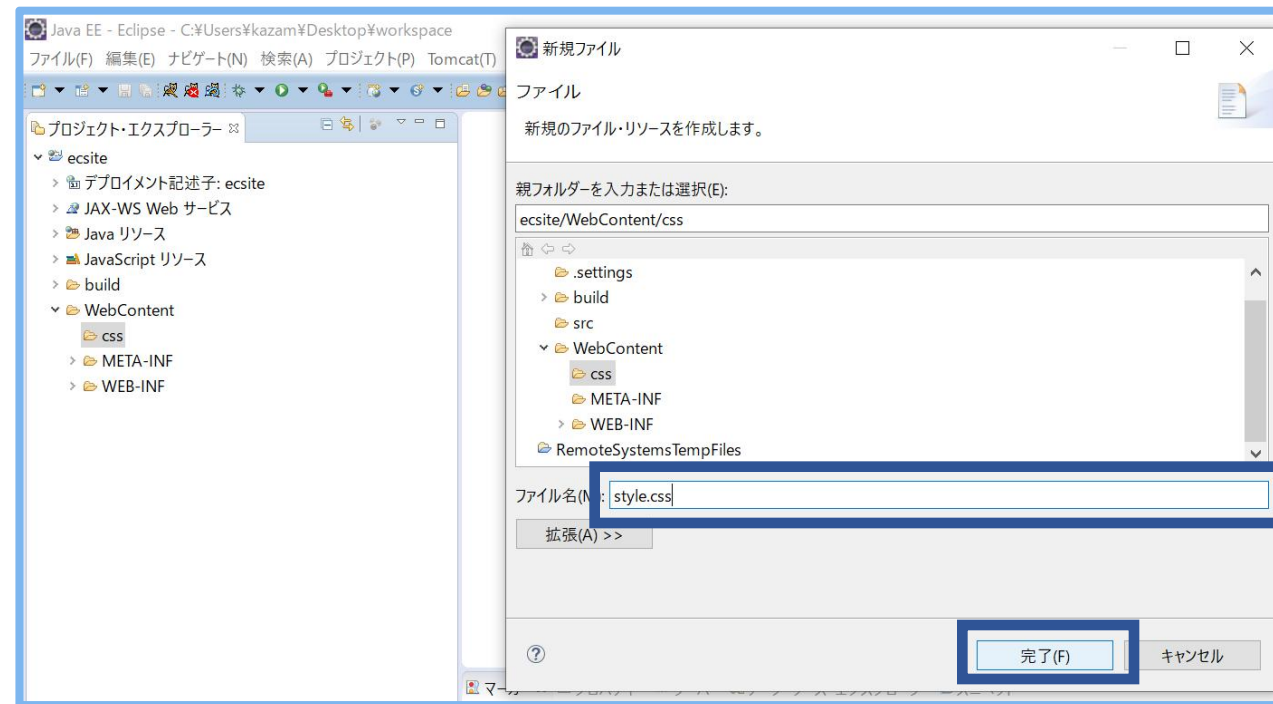


② 「名前(M):」欄に「css」と入力し、  
完了ボタンをクリックします。

## 2 CSSフォルダの作成



① 先ほど作ったcssフォルダを選択し、「右クリック」「新規」「ファイル」を選択します。



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

style.css(cssファイル)

```
/* =====TAG LAYOUT===== */  
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===== */  
#top {  
    width:780px;  
    margin:30px auto;
```

次へ続きます。



前の続きです。

```
border:1px solid #333;
}

#header {
  width: 100%;
  height: 80px;
  background-color: black;
}

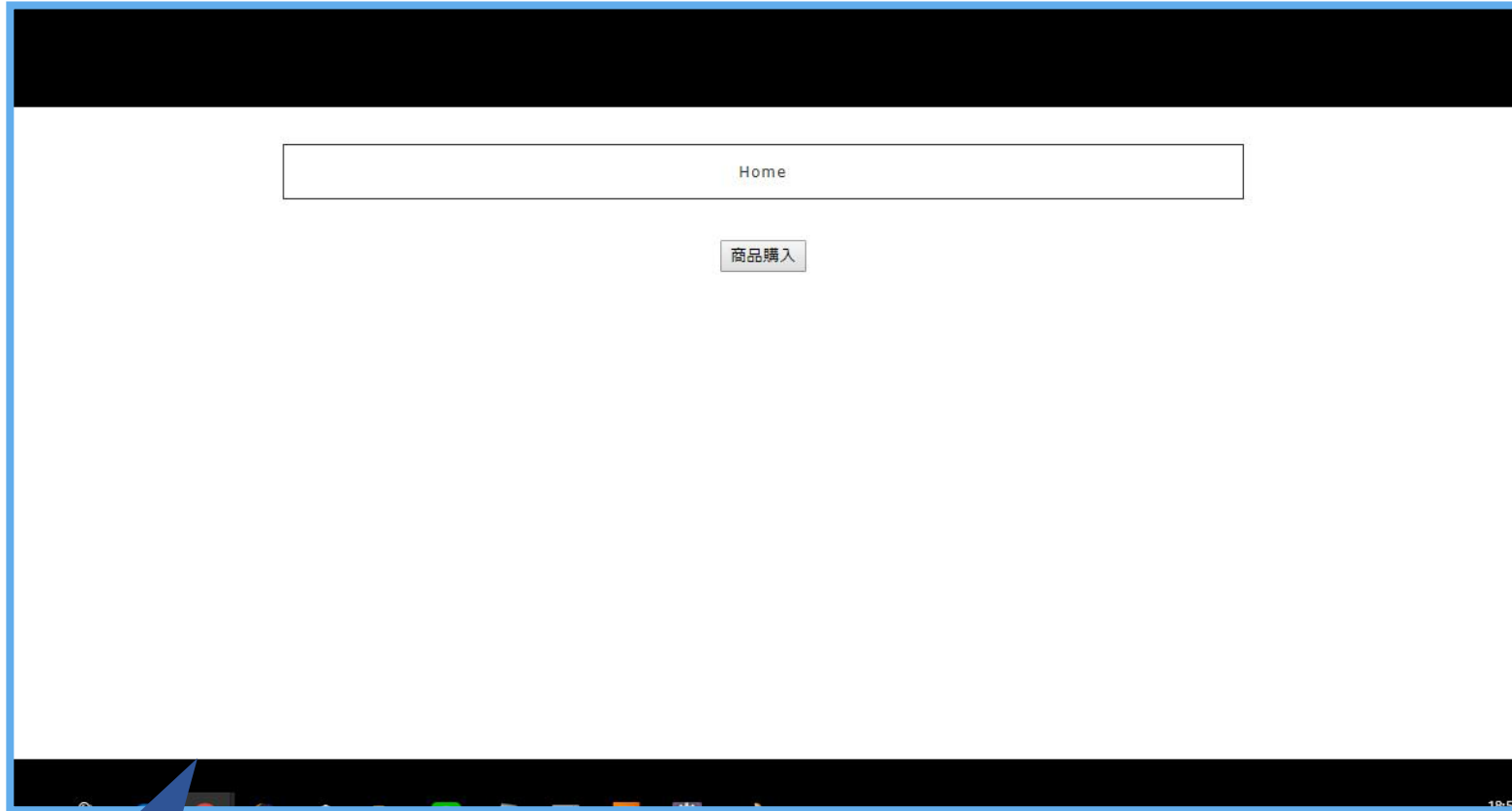
#main {
  width: 100%;
  height: 500px;
  text-align: center;
}

#footer {
  width: 100%;
  height: 80px;
  background-color: black;
  clear:both;
}
```

次へ続きます。

# home.jspの作成

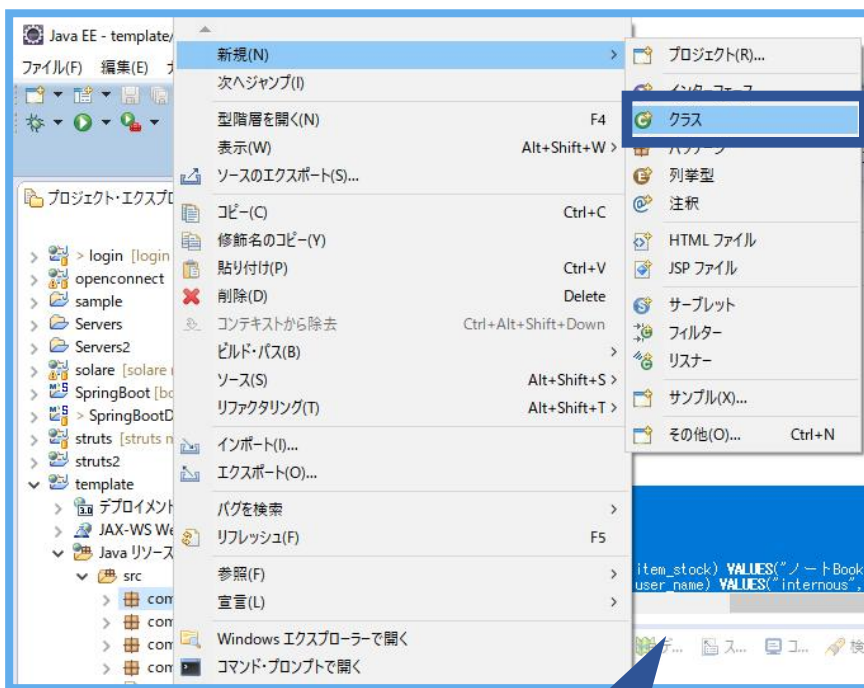
## home.jspの画面



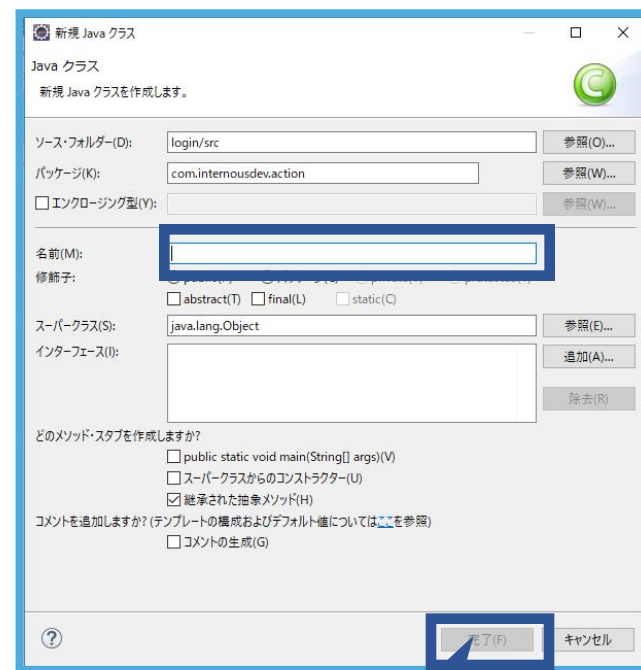
上記の画面が表示できれば成功です。

# HomeAction.javaの作成

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

# HomeAction.javaの作成

HomeAction(javaファイル)

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

```
import com.opensymphony.xwork2.ActionSupport;
import java.util.Map;
import org.apache.struts2.interceptor.SessionAware;
import com.internousdev.ecsite.dao.BuyItemDAO;
import com.internousdev.ecsite.dto.BuyItemDTO;
```

```
public class HomeAction extends ActionSupport implements SessionAware {
    private Map<String, Object> session;
```

```
    public String execute() {
```

```
        String result = "login";
```

```
        if (session.containsKey("login_user_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());
```

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。

③ 以下の内容を写経します。

ログイン済み判定を行います。  
一度ログインしている場合はログイン認証画面に遷移させることなく、商品画面へ遷移させます。

次へ続きます。

# HomeAction.javaの作成

前の続きです。

```
        result = SUCCESS;
    }
    return result;
}

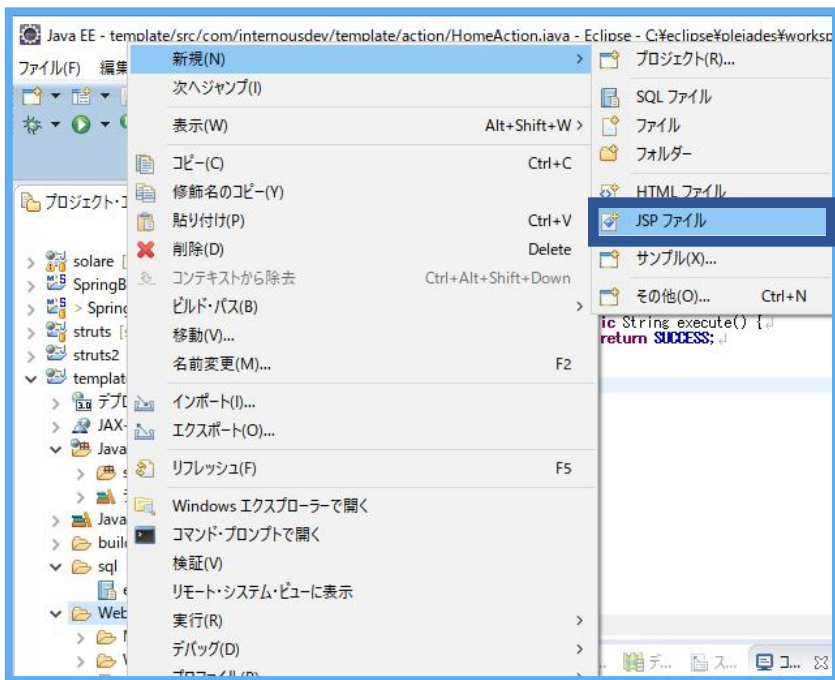
public Map<String, Object> getSession() {
    return this.session;
}

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

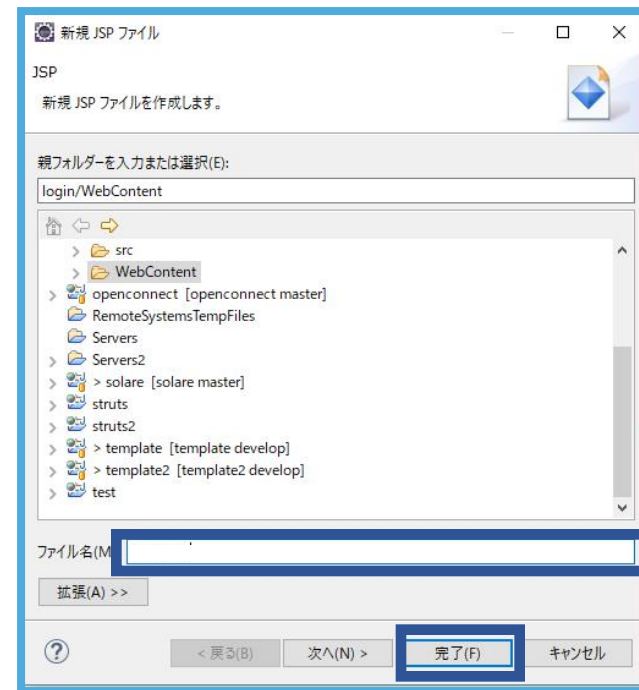
「execute」メソッドの戻り値として、  
ログイン状態の場合は「SUCCESS」  
ログインしていない場合は「login」を返します。

# login.jspの作成

## 3 login.jsp



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「login.jsp」を入力し、完了ボタンをクリックします。

# login.jspの作成

login.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 charset="UTF-8">
<link rel="stylesheet" type="text/css" href="./css/style.css">

<title>Login画面</title>

</head>
```

次へ続きます。

前の続きです。

```
<body>
  <div id="header">
  </div>
  <div id="main">
    <div id="top">
      <p>Login</p>
    </div>
    <div>
      <h3>商品を購入する際にはログインをお願いします。</h3>
      <s:form action="LoginAction">
        <s:textfield name="loginUserId"/>
        <s:password name="loginPassword"/>
        <s:submit value="ログイン"/>
      </s:form>
      <br/>
    </div>
  </div>
</body>
```

次へ続きます。



前の続きです。

```
<div id="text-link">
  <p>新規ユーザー登録は
    <a href='<s:url action="UserCreateAction" />'>こちら</a></p>
  <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>

</body>

</html>
```

# login.jspの作成

## login.jspの画面

Login

商品を購入する際にはログインをお願いします。

ログイン

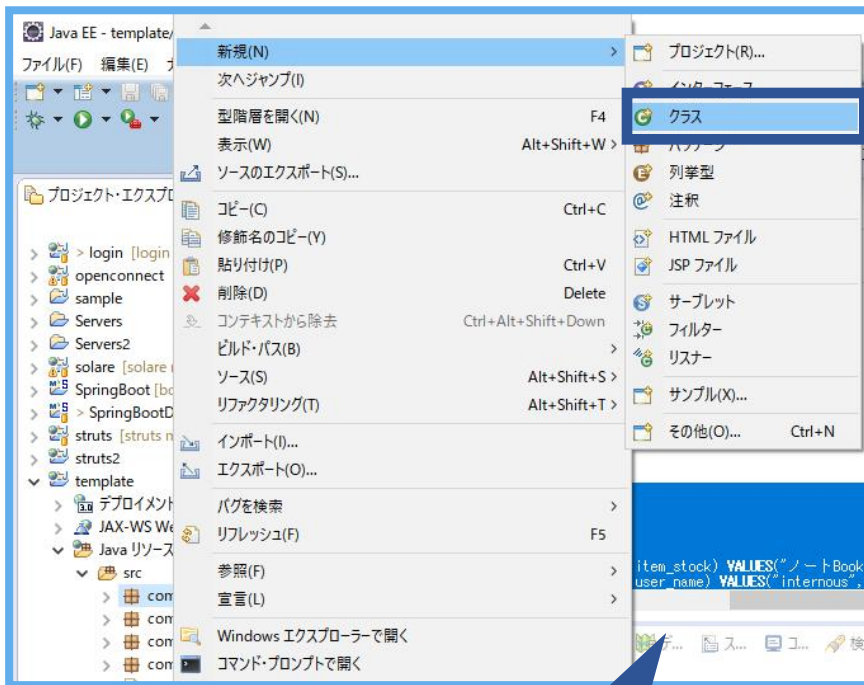
新規ユーザー登録は[こちら](#)

Homeへ戻る場合は[こちら](#)

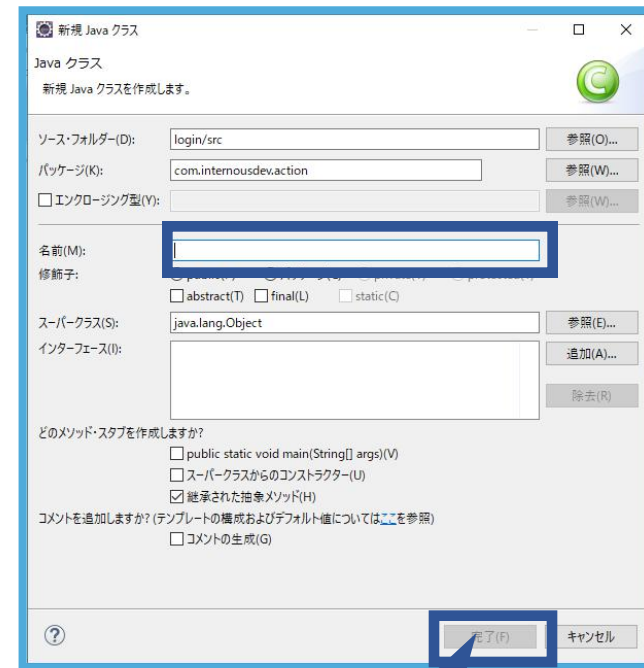
上記の画面が表示できれば成功です。

# LoginAction.javaの作成

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

# LoginAction.javaの作成

LoginAction (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.internousdev.ecsite.dao.BuyItemDAO;
import com.internousdev.ecsite.dao.LoginDAO;
import com.internousdev.ecsite.dto.BuyItemDTO;
import com.internousdev.ecsite.dto.LoginDTO;
import com.opensymphony.xwork2.ActionSupport;
public class LoginAction extends ActionSupport implements SessionAware {

    private String loginUserId;
    private String loginPassword;
    private Map<String, Object> session;
    private LoginDAO loginDAO = new LoginDAO();
    private LoginDTO loginDTO = new LoginDTO();
    private BuyItemDAO buyItemDAO = new BuyItemDAO();
```

次へ続きます。

# LoginAction.javaの作成

前の続きです。

```
public String execute() {  
    String result = ERROR;  
    loginDTO = loginDAO.getLoginUserInfo(loginUserId, loginPassword);  
    session.put("loginUser", loginDTO);  
  
    if(((LoginDTO) session.get("loginUser")).getLoginFlg()) {  
        result = SUCCESS;  
        BuyItemDTO buyItemDTO = buyItemDAO.getBuyItemInfo();  
  
        session.put("login_user_id", loginDTO.getLoginId());  
        session.put("id", buyItemDTO.getId());  
        session.put("buyItem_name", buyItemDTO.getItemName());  
        session.put("buyItem_price", buyItemDTO.getItemPrice());  
  
        return result;  
    }  
    return result;  
}
```

入力値からユーザー情報の検索を行います。  
ログイン認証が成功した場合、次の画面で  
「商品情報」が必要なため商品情報を取得します。

次へ続きます。

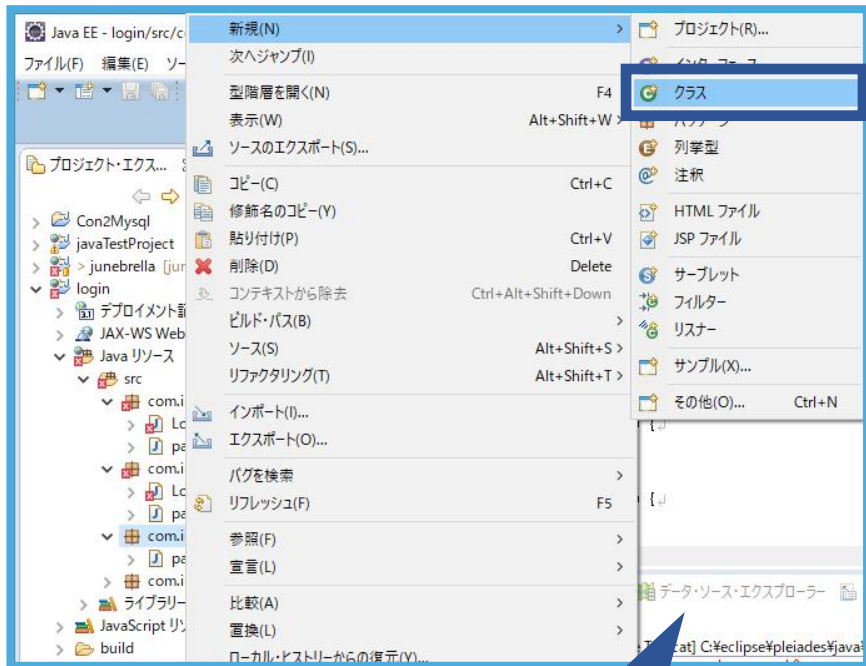
# LoginAction.javaの作成

前の続きです。

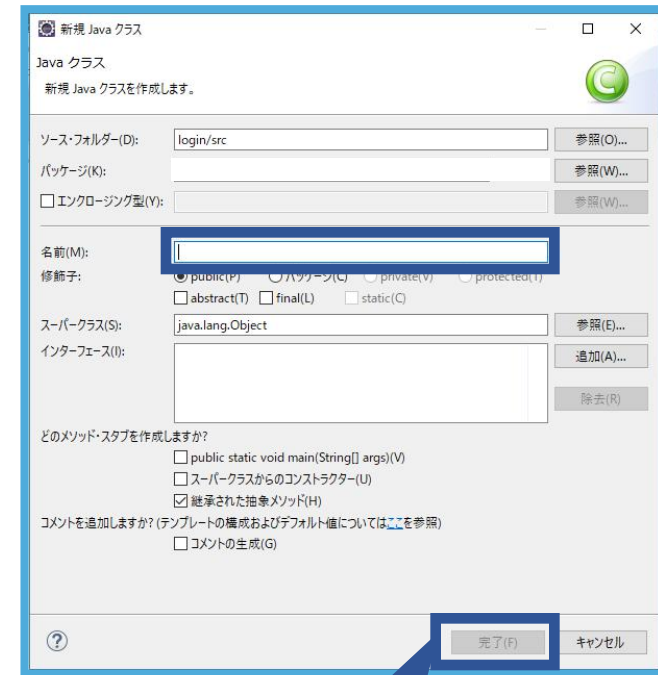
```
public String getLoginUserId() {  
    return loginUserId;  
}  
public void setLoginUserId(String loginUserId) {  
    this.loginUserId = loginUserId;  
}  
public String getLoginPassword() {  
    return loginPassword;  
}  
public void setLoginPassword(String loginPassword) {  
    this.loginPassword = loginPassword;  
}  
public Map<String, Object> getSession() {  
    return this.session;  
}  
@Override  
public void setSession(Map<String, Object> session) {  
    this.session = session;  
}  
}
```

# LoginDTO.javaの作成

## 4 LoginDTO.java



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「LoginDTO」を入力し、完了ボタンをクリックします。

# LoginDTO.javaの作成

③ 以下の内容を写経します。

LoginDTO(javaファイル)

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

public class LoginDTO {

    private String loginId;
    private String loginPassword;
    private String userName;
    private boolean loginFlg = false;

    public String getLoginId() {
        return loginId;
    }

    public void setLoginId(String loginId) {
        this.loginId = loginId;
    }
}
```

次へ続きます。

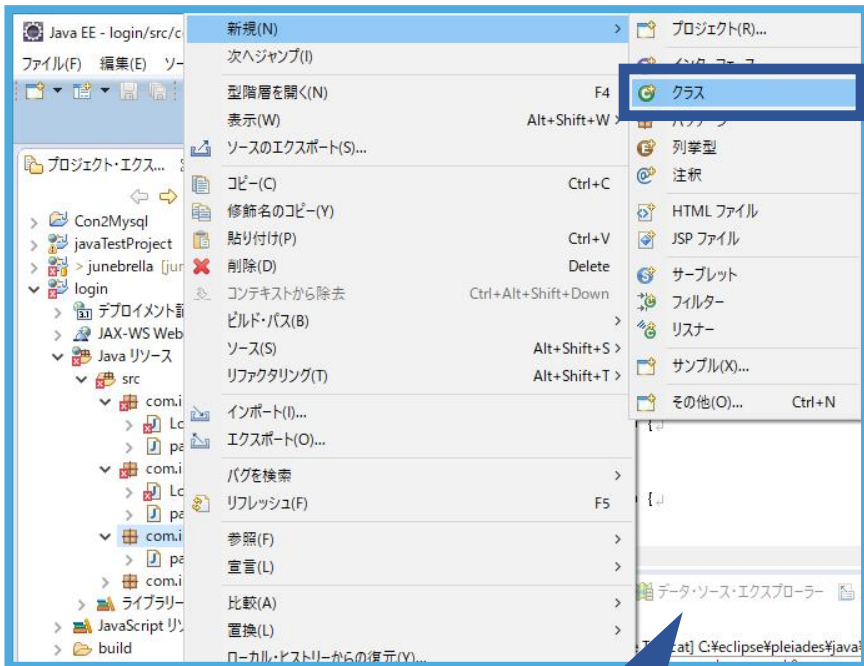


前の続きです。

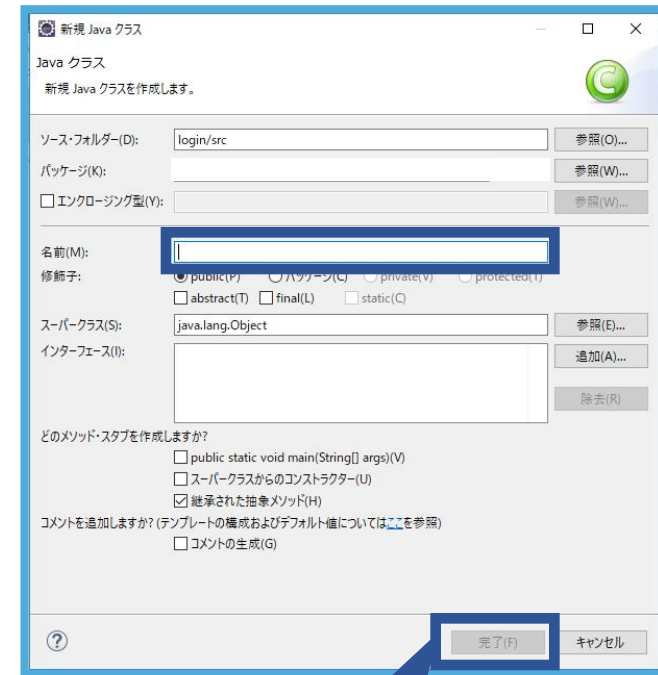
```
public String getLoginPassword() {  
    return loginPassword;  
}  
public void setLoginPassword(String loginPassword) {  
    this.loginPassword = loginPassword;  
}  
public String getUserName() {  
    return userName;  
}  
public void setUserName(String userName) {  
    this.userName = userName;  
}  
public boolean getLoginFlg() {  
    return loginFlg;  
}  
public void setLoginFlg(boolean loginFlg) {  
    this.loginFlg = loginFlg;  
}  
}
```

# LoginDAO.javaの作成

## 5 LoginDAO.java



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「LoginDAO」を入力し、完了ボタンをクリックします。

# LoginDAO.javaの作成

③ 以下の内容を写経します。

LoginDAO(javaファイル)

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

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

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

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

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

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

```
public class LoginDAO {
```

```
    public LoginDTO getLoginUserInfo(String loginUserId, String loginPassword) {
```

```
        DBConnector db = new DBConnector();
```

```
        Connection con = db.getConnection();
```

```
        LoginDTO loginDTO = new LoginDTO();
```

メソッド名は処理内容を分かりやすくするために「getLoginUserInfo」にします。

次へ続きます。

# LoginDAO.javaの作成

前の続きです。

```
String sql = "SELECT * FROM login_user_transaction where login_id = ? AND  
login_pass = ?";  
  
try {  
    PreparedStatement ps = con.prepareStatement(sql);  
  
    ps.setString(1, loginUserId);  
    ps.setString(2, loginPassword);  
  
    ResultSet rs = ps.executeQuery();  
  
    if(rs.next()) {  
        loginDTO.setLoginId(rs.getString("login_id"));  
        loginDTO.setLoginPassword(rs.getString("login_pass"));  
        loginDTO.setUserName(rs.getString("user_name"));  
  
        if(rs.getString("login_id") != null) {  
            loginDTO.setLoginFlg(true);  
        }  
    }  
}
```

次へ続きます。

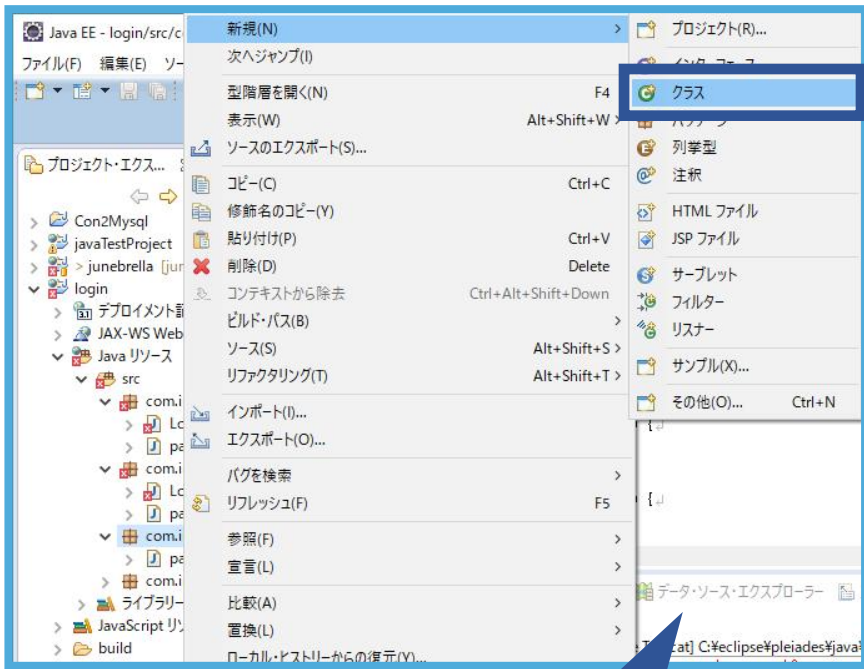
# LoginDAO.javaの作成

前の続きです。

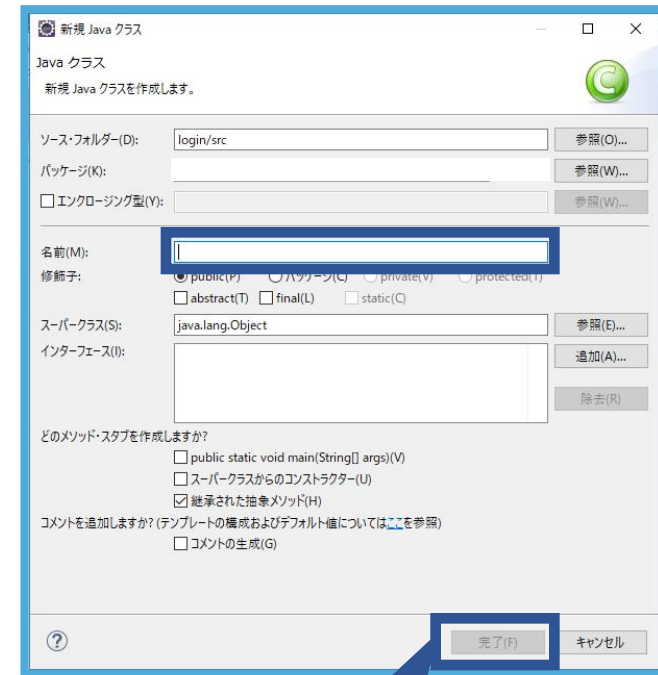
```
    }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return loginDTO;  
}  
  
}
```

# BuyItemDAO.javaの作成

## 6 BuyItemDAO.java



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.javaの作成

③ 以下の内容を写経します。

BuyItemDAO(javaファイル)

```
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 {
    public BuyItemDTO getBuyItemInfo() {
        DBConnector db = new DBConnector();
        Connection con = db.getConnection();
        BuyItemDTO buyItemDTO = new BuyItemDTO();
```

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

```
        try {
```

```
            PreparedStatement ps = con.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
```

SQL文を実行します。

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.javaの作成

前の続きです。

```
        if(rs.next()) {  
            buyItemDTO.setId(rs.getInt("id"));  
            buyItemDTO.setItemName(rs.getString("item_name"));  
            buyItemDTO.setItemPrice(rs.getString("item_price"));  
        }
```

DBから取得した情報を  
DTOクラスに格納します。

```
    } catch(Exception e) {  
        e.printStackTrace();  
    }
```

処理中に例外が発生した際に  
実行する処理です。

```
    return buyItemDTO;  
}
```

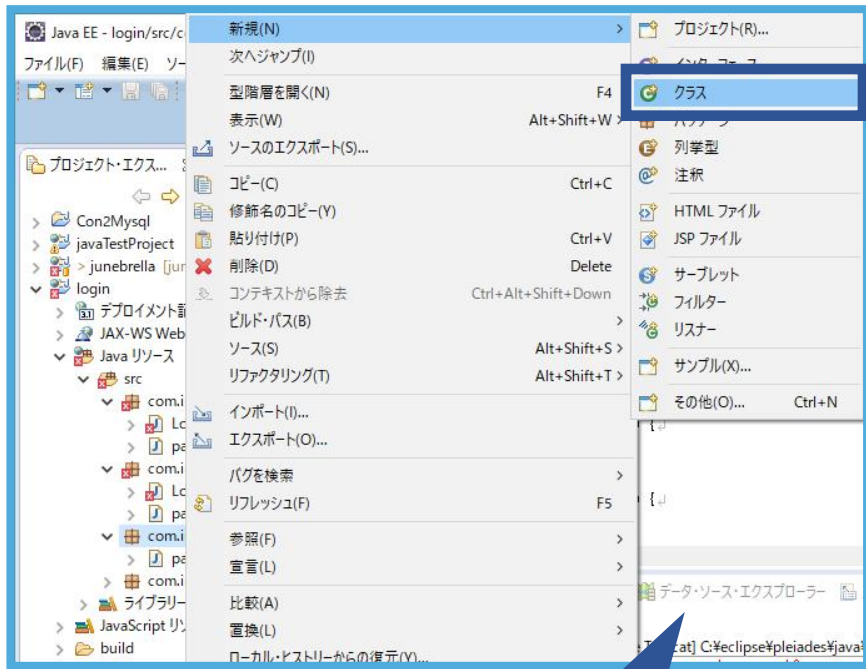
ActionクラスにDTOクラスを返します。

```
}
```

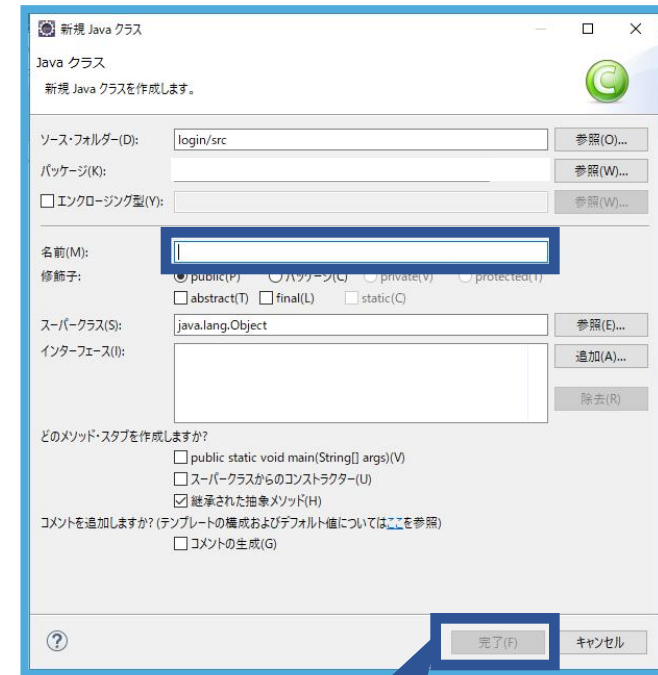


# BuyItemDTO.javaの作成

## 7 BuyItemDTO.java



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.javaの作成

③ 以下の内容を写経します。

BuyItemDTO(javaファイル)

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

```
public class BuyItemDTO {
```

```
    private int id;  
    private String itemName;  
    private String itemPrice;
```

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

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

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。

フィールド変数に対応したGetter  
Setterを定義します。

次へ続きます。

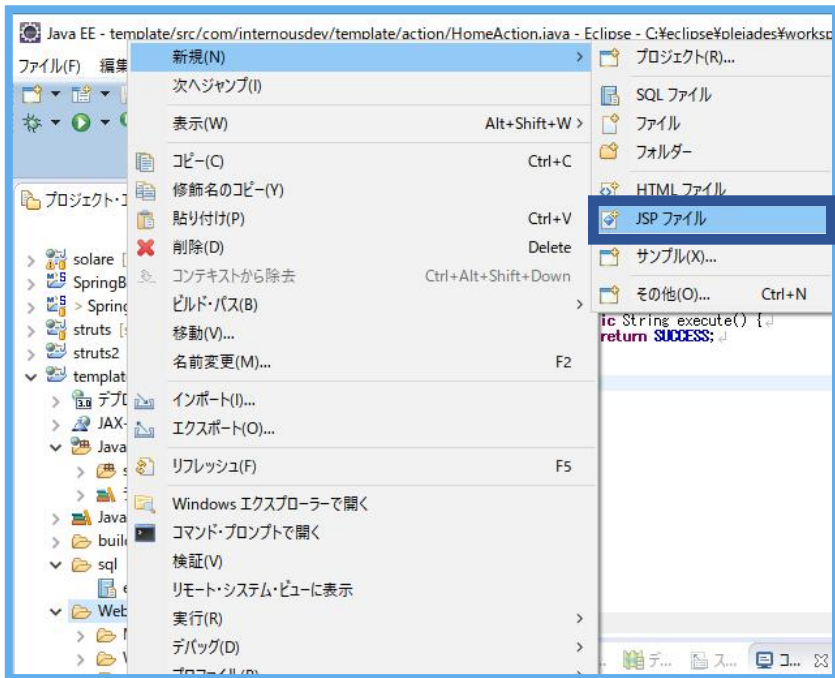
# BuyItemDTO.javaの作成

前の続きです。

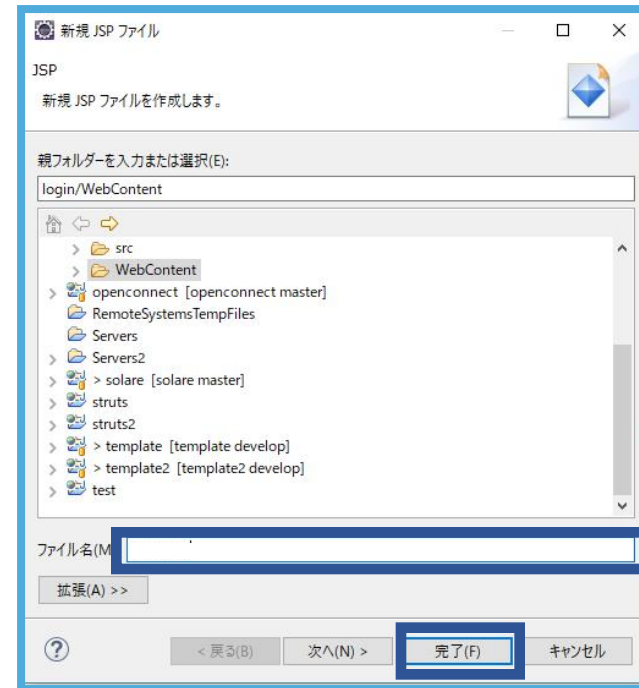
```
public String getItemName() {  
    return itemName;  
}  
  
public void setItemName(String itemName) {  
    this.itemName = itemName;  
}  
  
public String getItemPrice() {  
    return itemPrice;  
}  
  
public void setItemPrice(String itemPrice) {  
    this.itemPrice = itemPrice;  
}  
}
```

# buyItem.jspの作成

## 8 buyItem.jsp



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 charset="UTF-8">
<link rel="stylesheet" type="text/css" href="./css/style.css">

<title>BuyItem画面</title>

</head>
```

次へ続きます。

# buyItem.jspの作成

前の続きです。

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

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>
<tr>
  <td>
    <span>購入回数</span>
  </td>
  <td>
```

次へ続きます。

# buyItem.jspの作成

前の続きです。

金払い

```
</tr>
<tr>
  <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>
  <td>
    <span>支払い方法</span>
    <input type="radio" name="pay" value="1" checked="checked">現
    <input type="radio" name="pay" value="2">クレジットカード
  </td>
</tr>
```

次へ続きます。



# buyItem.jspの作成

前の続きです。

```

        <tr>
            <td>
                <s:submit value="購入" />
            </td>
        </tr>
    </table>
</s:form>
<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>
</body>
</html>
```

次へ続きます。

# buyItem.jspの作成

## buyItem.jspの画面



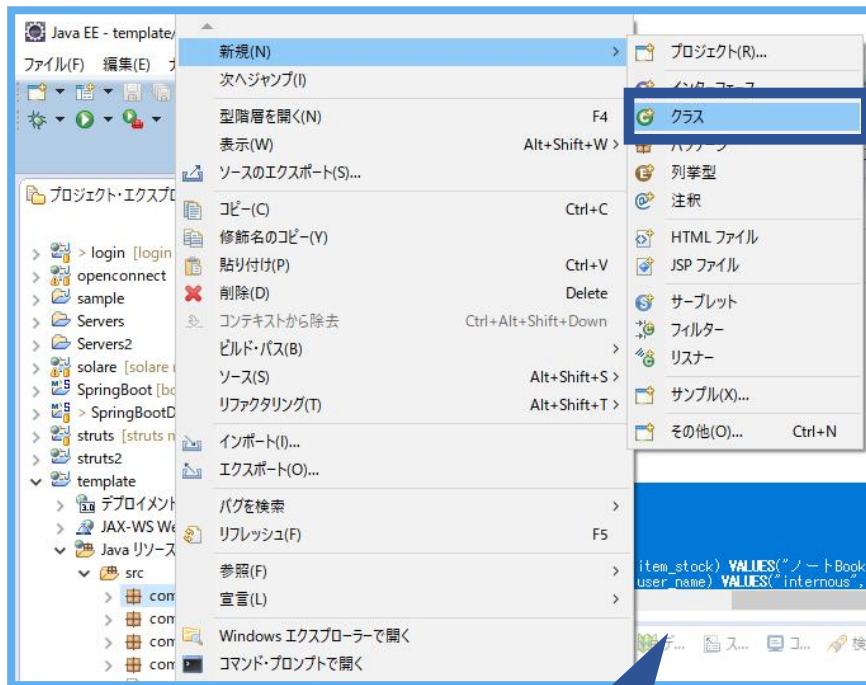
The screenshot displays a web form titled "BuyItem" within a white rectangular box. The form contains the following elements:

- 商品名** (Item Name): NoteBook
- 値段** (Price): 100円
- 購入個数** (Purchase Quantity): A dropdown menu showing "1".
- 支払い方法** (Payment Method): Two radio buttons, "現金払い" (Cash Payment) which is selected, 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こちら** (Click here to return to the previous screen): A text link.
- マイページはこちら** (Click here for my page): A text link.

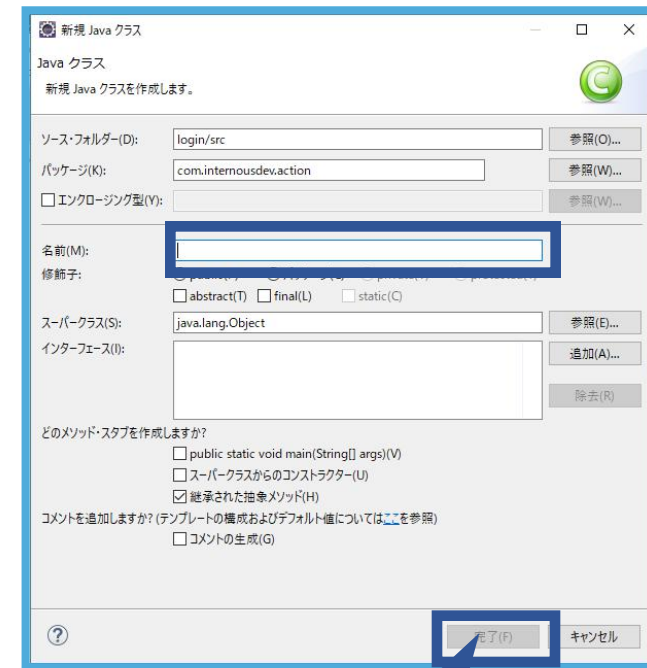
上記の画面が表示できれば成功です。

# GoHomeAction.javaの作成

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

# GoHomeAction.javaの作成

GoHomeAction(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 java.util.Map;
import org.apache.struts2.interceptor.SessionAware;

public class GoHomeAction extends ActionSupport implements SessionAware {
    private Map<String, Object> session;

    public String execute() {
        return SUCCESS;
    }

    public Map<String, Object> getSession() {
        return this.session;
    }

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