

---

# ECサイト初級

購入履歴機能の作成

---

## 6 時間目

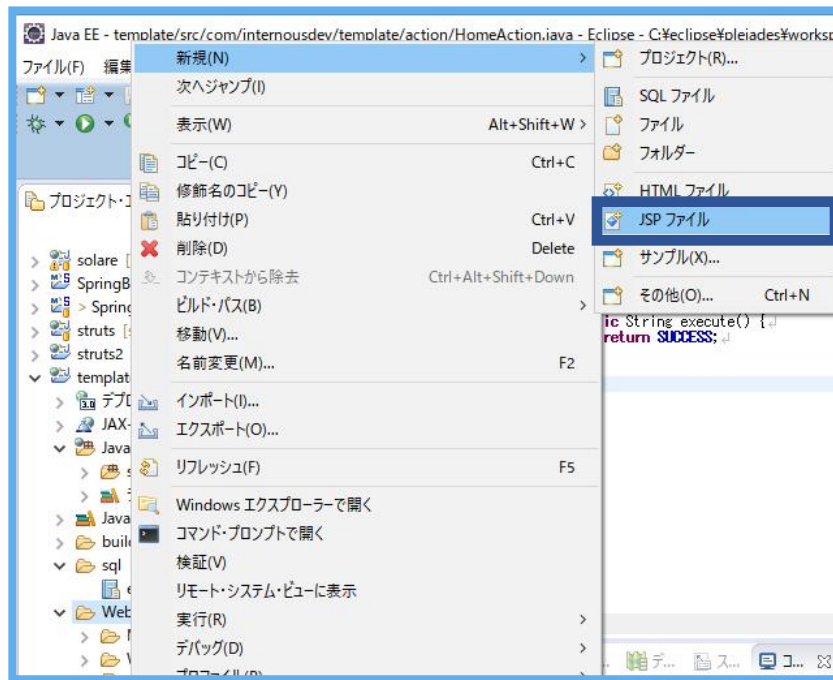
# 作業目次

## 作成機能一覧

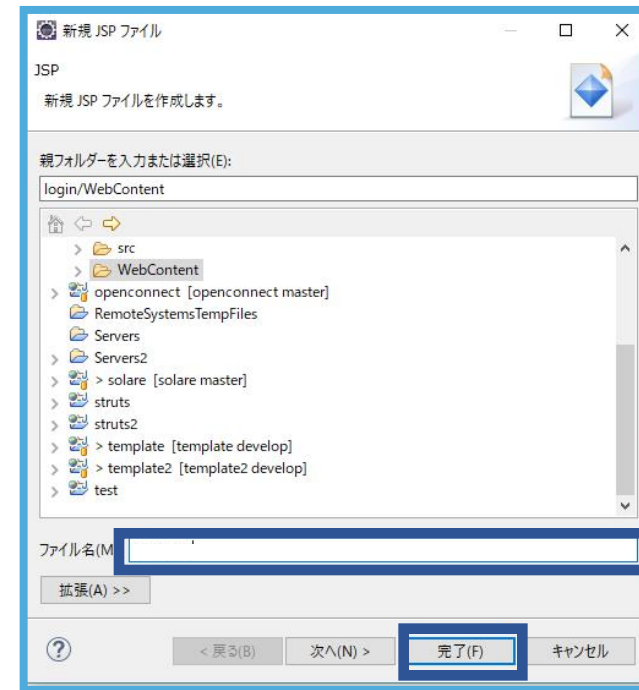
1. ログイン認証機能
2. ユーザー登録機能
3. 商品購入機能
- 4. 購入履歴機能**

# myPage.jspの作成

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

# myPage.jspの作成

③ 以下の内容を写経します。

myPage.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 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===== */  
#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。

前の続きです。

```
</style>
</head>
<body>
  <div id="header">
    <div id="pr">
    </div>
  </div>
  <div id="main">
    <div id="top">
      <p>MyPage</p>
    </div>
    <div>
      <s:if test="session.message == ''">
        <h3>ご購入情報は以下になります。</h3>
        <table>
          <tr>
            <td>商品名</td>
            <td><s:property value="session.buyItem_name" /></td>
          </tr>
```

次へ続きます。

前の続きです。

```
</tr>
<tr>
  <td>値段</td>
  <td>
    <s:property value="session.total_price" />
    <span>円</span>
  </td>
</tr>
<tr>
  <td>購入回数</td>
  <td>
    <s:property value="session.total_count" />
    <span>個</span>
  </td>
</tr>
<tr>
  <td>支払い方法</td>
  <td><s:property value="session.total_payment" /></td>
</tr>
</table>
```

次へ続きます。



前の続きです。

```
<s:form action="MyPageAction">
    <input type="hidden" name="deleteFlg" value="1">
    <s:submit value="削除" method="delete" />
</s:form>
</s:if>
<s:if test="session.message != null">
    <h3><s:property value="session.message" /></h3>
</s:if>
<div>
    <br>
    <span>前画面に戻る場合は</span>
    <a href='<s:url action="HomeAction" />'>ログアウト</a>
    <span>をお願いします</span>
</div>
</div>
</div>
```

次へ続きます。

前の続きです。

```
<div id="footer">  
    <div id="pr">  
    </div>  
</div>  
</body>  
</html>
```

# myPage.jspの作成

## myPage.jspの画面

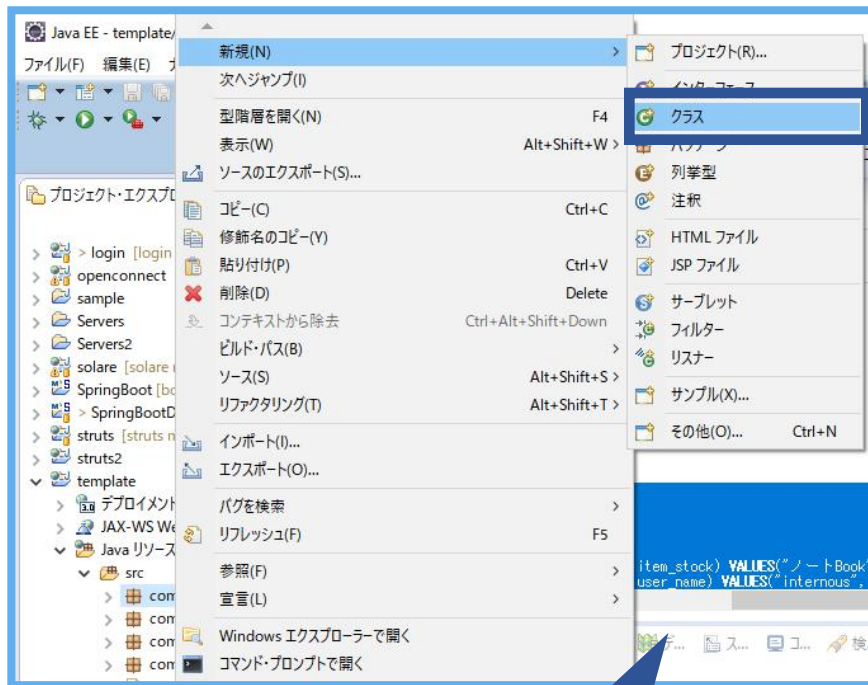
The screenshot shows a web form titled "BuyItem" within a browser window. The form contains the following elements:

- A header bar with the text "BuyItem".
- Form fields for:
  - 商品名 (Product Name): NoteBook
  - 値段 (Price): 100円
  - 在庫 (Inventory): A dropdown menu showing "1".
- Payment method options: 支払い方法 (Payment Method) with radio buttons for 現金払い (Cash Payment) and クレジットカード (Credit Card). The 現金払い option is selected.
- A "購入" (Purchase) button.
- A link at the bottom: 前画面に戻る場合は[こちら](#) (If you want to return to the previous screen, click [here](#)).

上記の画面が表示できれば成功です。

# MyPageAction.javaの作成

## 2 MyPageAction.java



① 「com.internousdev.tempalte.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.javaの作成

③ 以下の内容を写経します。

MyPageAction .java  
(javaファイル)

```
package com.internousdev.template.action;
import java.sql.SQLException;
import java.util.Map;
import org.apache.struts2.interceptor.SessionAware;
import com.internousdev.template.dao.MyPageDAO;
import com.internousdev.template.dto.MyPageDTO;
import com.opensymphony.xwork2.ActionSupport;

public class MyPageAction extends ActionSupport implements SessionAware{
    public Map<String, Object> session;
    public MyPageDAO myPageDAO = new MyPageDAO();
    public MyPageDTO myPageDTO = new MyPageDTO();
    public String deleteFlg;
    private String result;
    public String execute() throws SQLException {

        // 商品履歴を削除しない場合
        if(deleteFlg == null) {
```

次へ続きます。

# MyPageAction.javaの作成

前の続きです。

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

myPageDTO = myPageDAO.getMyPageUserInfo(item_transaction_id,
user_master_id);

session.put("buyItem_name", myPageDTO.getItemName());
session.put("total_price", myPageDTO.getTotalPrice());
session.put("total_count", myPageDTO.getTotalCount());
session.put("total_payment", myPageDTO.getPayment());
session.put("message", "");
// 商品履歴を削除する場合
} else if(deleteFlg.equals("1")) {
    delete();
}
result = SUCCESS;
return result;
}
```

次へ続きます。

# MyPageAction.javaの作成

前の続きです。

```
public void delete() throws SQLException {  
  
    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) {  
        session.put("message", "商品情報を正しく削除しました。");  
    } else if(res == 0) {  
        session.put("message", "商品情報の削除に失敗しました。");  
    }  
}
```

次へ続きます。

# MyPageAction.javaの作成

前の続きです。

```
public String getDeleteFlg() {  
    return deleteFlg;  
}
```

```
public void setDeleteFlg(String deleteFlg) {  
    this.deleteFlg = deleteFlg;  
}
```

```
@Override
```

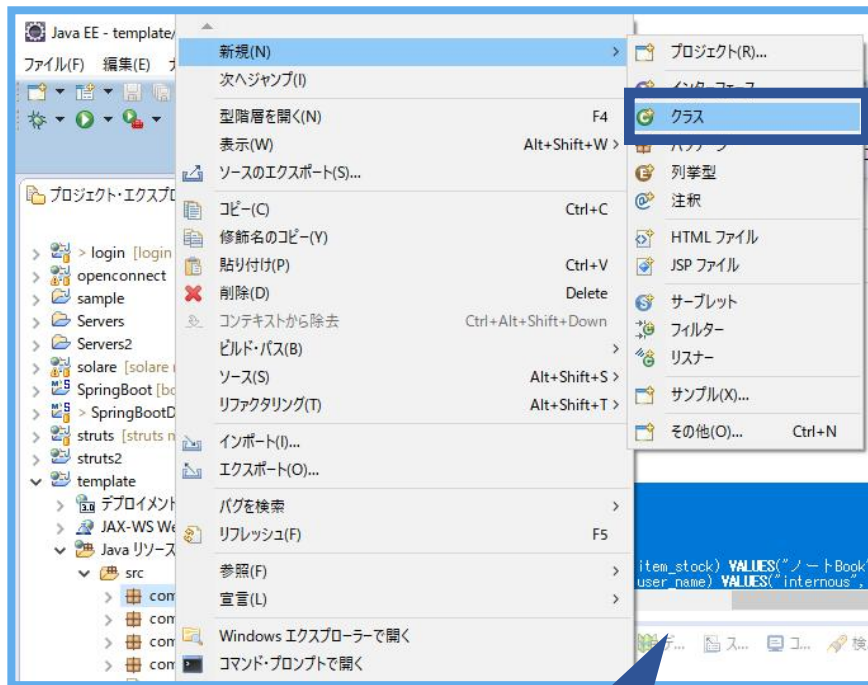
```
public void setSession(Map<String, Object> loginSessionMap) {  
    this.session = loginSessionMap;  
}
```

```
}
```

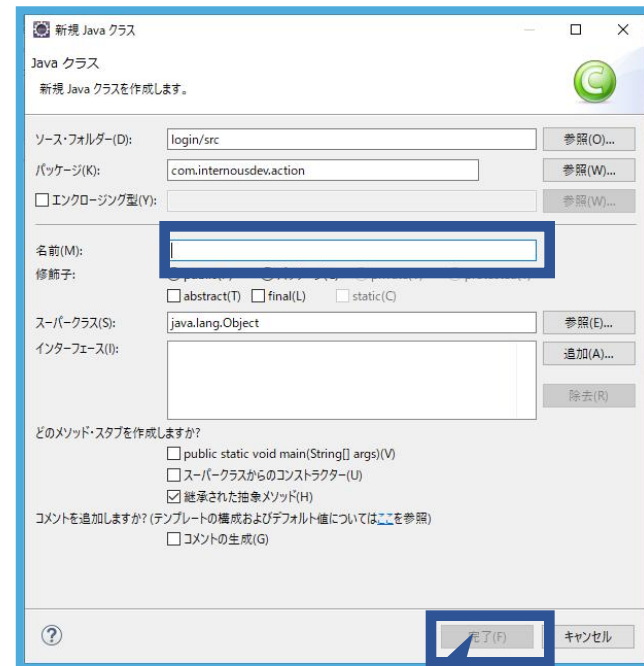


# MyPageDAO.javaの作成

## 3 MyPageDAO.java



- ① 「com.internousdev.tempalte.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.javaの作成

③ 以下の内容を写経します。

MyPageDAO .java  
(javaファイル)

```
package com.internousdev.template.action;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.internousdev.template.dto.MyPageDTO;
import com.internousdev.template.util.DBConnector;

public class MyPageDAO {
    private DBConnector dbConnector = new DBConnector();
    private Connection connection = dbConnector.getConnection();
    private MyPageDTO myPageDTO = new MyPageDTO();

    public MyPageDTO getMyPageUserInfo(String item_ttion_id, String user_master_id) throws
    SQLException {
```

次へ続きます。

前の続きです。

```
String sql = "SELECT iit.item_name, ubit.total_price, ubit.total_count,
ubit.pay 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 ubit.insert_date DESC";

try {
    PreparedStatement preparedStatement =
connection.prepareStatement(sql);
    preparedStatement.setString(1, item_transaction_id);
    preparedStatement.setString(2, user_master_id);
    ResultSet resultSet = preparedStatement.executeQuery();

    if(resultSet.next()) {
        myPageDTO.setItemName(resultSet.getString("item_name"));
        myPageDTO.setTotalPrice(resultSet.getString("totalprice"));
        myPageDTO.setTotalCount(resultSet.getString("totalcount"));
        myPageDTO.setPayment(resultSet.getString("pay"));
    }
}
```

次へ続きます。

# MyPageDAO.javaの作成

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 {  
    String sql = "DELETE FROM user_buy_item_transaction WHERE  
item_transaction_id = ? AND user_master_id = ?";  
    PreparedStatement preparedStatement;  
    int result = 0;
```

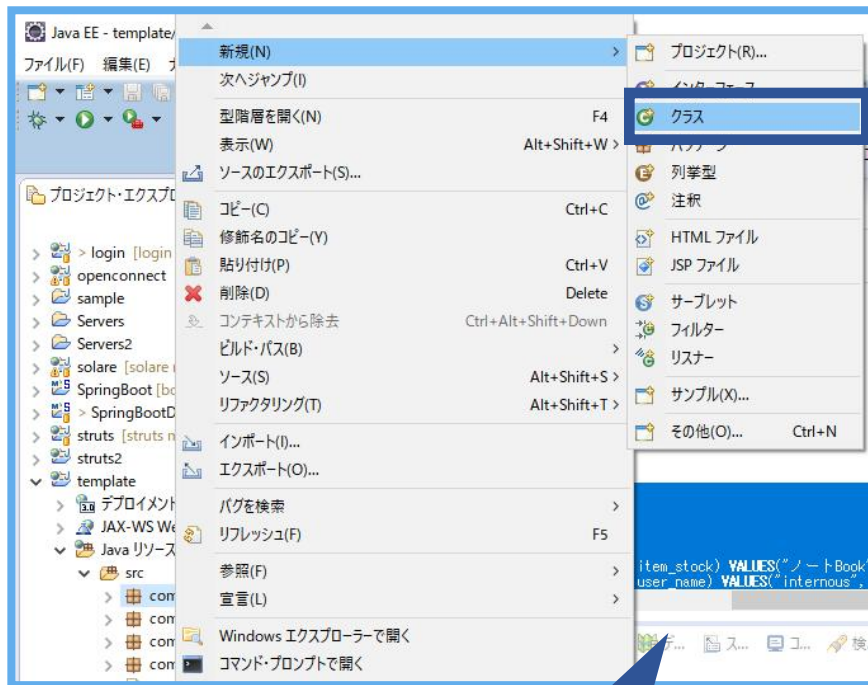
次へ続きます。

前の続きです。

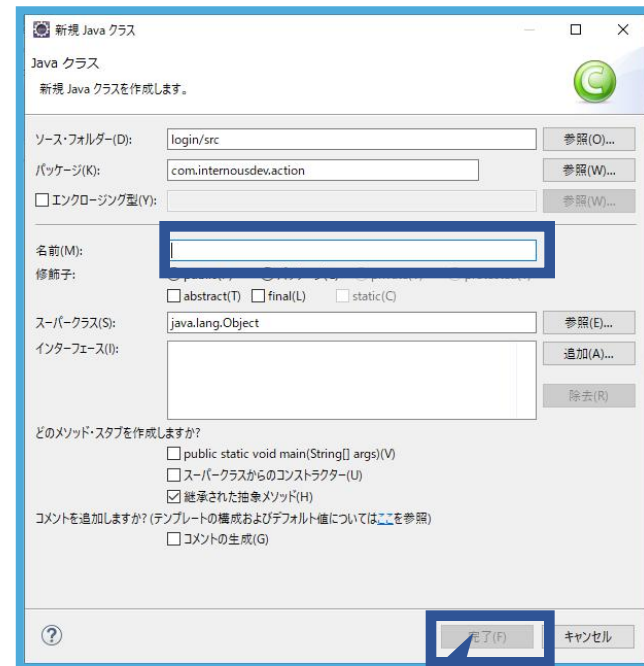
```
try {  
    preparedStatement = connection.prepareStatement(sql);  
    preparedStatement.setString(1, item_transaction_id);  
    preparedStatement.setString(2, user_, master_id);  
  
    result = preparedStatement.executeUpdate();  
  
} catch (SQLException e) {  
    e.printStackTrace();  
} finally {  
    connection.close();  
}  
return result;  
}  
}
```

# MyPageDTO.javaの作成

## 4 MyPageDTO.java



- ① 「com.internousdev.template.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.javaの作成

③ 以下の内容を写経します。

MyPageDTO .java  
(javaファイル)

```
package com.internousdev.template.dto;

public class MyPageDTO {
    private String itemName;
    private String totalPrice;
    private String totalCount;
    private String payment;

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

次へ続きます。

# MyPageDTO.javaの作成

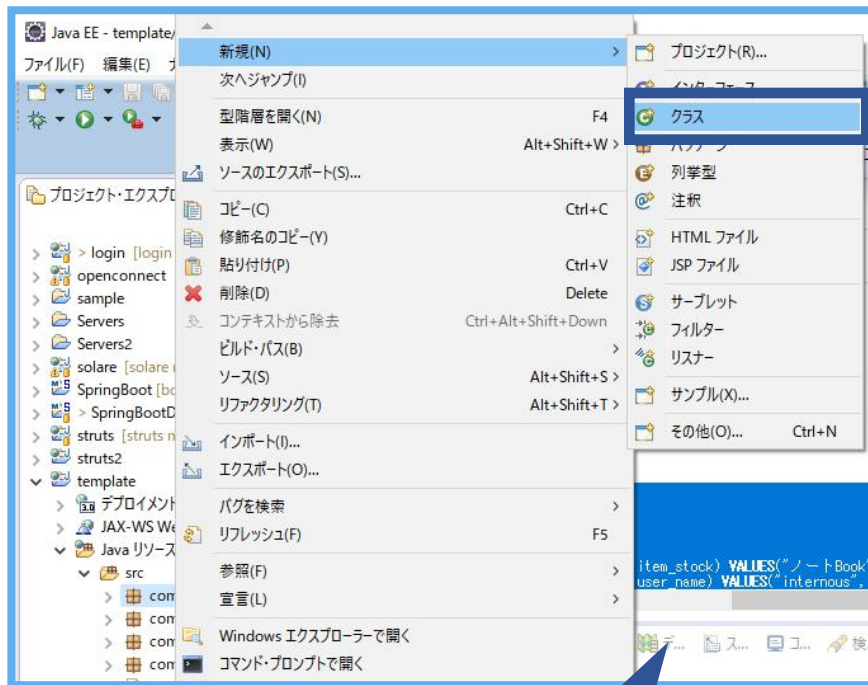
前の続きです。

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

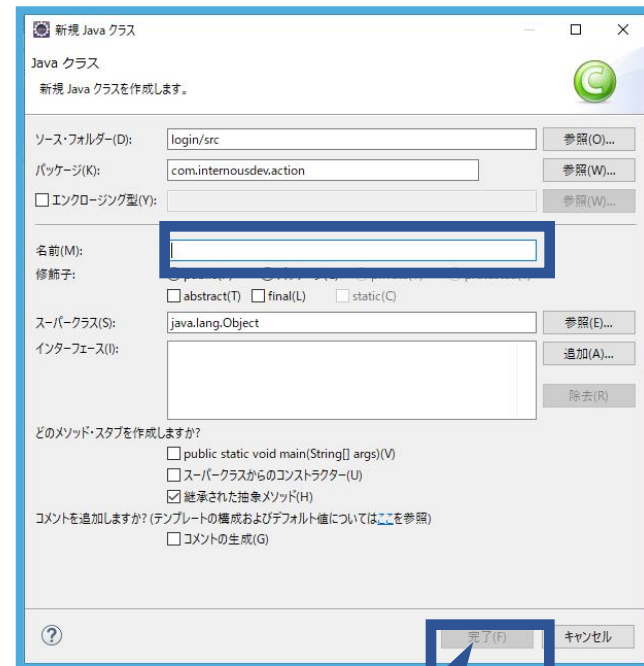


# LogoutAction.javaの作成

## 5 LogoutAction.java



① 「com.internousdev.tempalte.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.template.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;
    }
}
```