

11 예외처리

1

예외 처리의 개요

2

page 디렉티브 태그를 이용한 예외 처리

3

web.xml 파일을 이용한 예외처리

4

Try-catch-finally를 이용한 예외처리

5

[웹 쇼핑몰] 예외 처리 페이지 처리하기

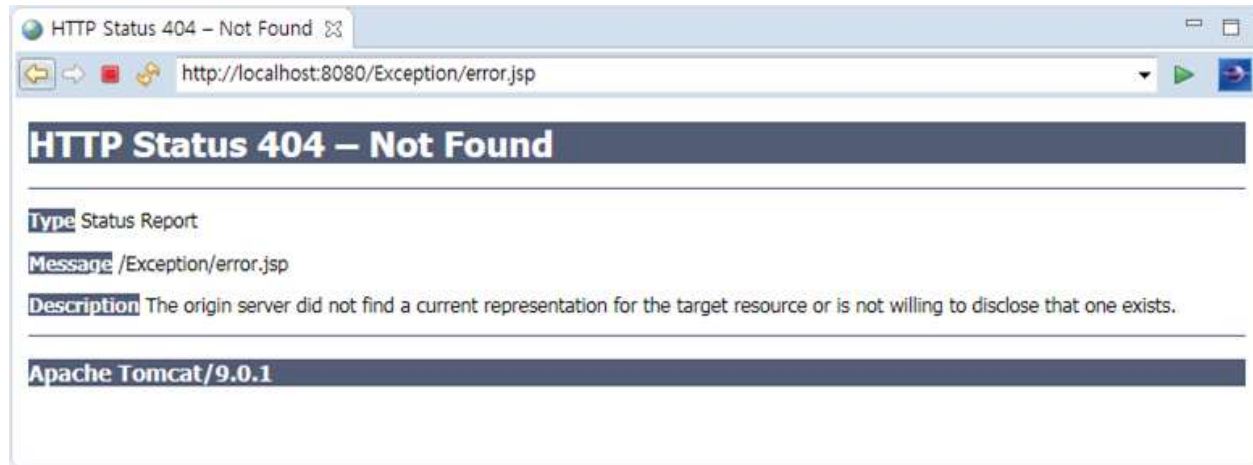
학습목표

- 예외 처리의 개념을 이해합니다.
- page 디렉티브 태그를 이용한 예외 처리 방법을 익힙니다.
- web.xml 파일을 이용한 예외 처리 방법을 익힙니다.
- try-catch-finally를 이용한 예외 처리 방법을 익힙니다.
- 웹 쇼핑몰의 예외 처리 페이지를 만듭니다.

1. 예외 처리의 개요

❖ 예외 처리

- 프로그램이 처리되는 동안 특정한 문제가 발생했을 때 처리를 중단하고 다른 처리를 하는 것으로 오류 처리라고도 함
- 웹 사이트를 이용하다가 주소를 잘못 입력하면 오류 페이지를 보게 됨
 - 웹 서버가 제공하는 오류 페이지로 해당 페이지에 발생한 오류, 디렉터리 구조, 톰캣 버전 등의 정보가 나타나 있기 때문에 웹 보안이 취약하여 쉽게 해킹 당할 수 있음



1. 예외 처리의 개요

❖ 예외 처리

- 웹 애플리케이션 실행 도중에 발생할 수 있는 오류에 대비한 예외 처리 코드를 작성하여 비정상적인 종료를 막을 수 있음

❖ 예외 처리 방법의 종류

표 11-1 예외 처리 방법의 종류

예외 처리 방법	설명
page 디렉티브 태그를 이용한 예외 처리	errorPage와 isErrorPage 속성을 이용합니다.
web.xml 파일을 이용한 예외 처리	<error-code> 또는 <exception-type> 요소를 이용합니다.
try/catch/finally를 이용한 예외 처리	자바 언어의 예외 처리 구문을 이용합니다.

2. page 디렉티브 태그를 이용한 예외 처리

❖ **errorPage** 속성으로 오류 페이지 호출하기

- **errorPage** 속성
 - 오류 페이지를 호출하는 **page** 디렉티브 태그의 속성

```
<%@ page errorPage = "오류 페이지 URL" %>
```

- JSP 페이지가 실행되는 도중에 오류가 발생하면 웹 서버의 기본 오류 페이지를 대신하여 **errorPage** 속성에 설정한 페이지가 오류 페이지로 호출

2. page 디렉티브 태그를 이용한 예외 처리

[errorPage 속성 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
```

```
<%@ page errorPage="error.jsp"%>
```

```
<html>
```

```
<head>
```

```
<title>Exception</title>
```

```
</head>
```

```
<body>
```

```
<%
```

```
String value = session.getAttribute("id").toString();
```

```
%>
```

```
</body>
```

```
</html>
```

2 오류 페이지 호출

1 오류 발생

2. page 디렉티브 태그를 이용한 예외 처리

예제 11-1 page 디렉티브 태그에 `errorPage` 속성을 이용하여 오류 페이지 호출하기

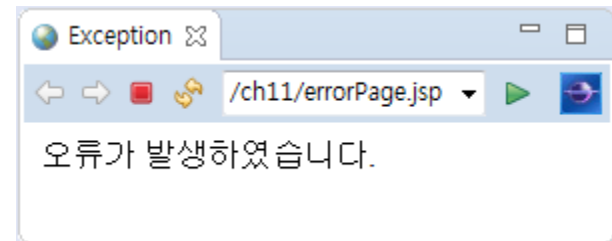
JSPBook/WebContent/ch11/errorPage.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <% page errorPage="errorPage_error.jsp"%>
03 <html>
04 <head>
05 <title>Exception</title>
06 </head>
07 <body>
08     name 파라미터 : <%=request.getParameter("name").toUpperCase()%>
09 </body>
10 </html>
```

2. page 디렉티브 태그를 이용한 예외 처리

JSPBook/WebContent/ch11/errorPage_error.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     오류가 발생하였습니다.
08 </body>
09 </html>
```



2. page 디렉티브 태그를 이용한 예외 처리

❖ isErrorPage 속성으로 오류 페이지 만들기

■ isErrorPage 속성

- 현재 JSP 페이지를 오류 페이지로 호출하는 page 디렉티브 태그의 속성
- 이때 오류 페이지에서 exception 내장 객체를 사용할 수 있음

```
<%@ page isErrorPage = "true" %>
```

표 11-2 exception 내장 객체의 메소드

메소드	형식	설명
getMessage()	String	오류 이벤트와 함께 들어오는 메시지를 출력합니다.
toString()	String	오류 이벤트의 toString()을 호출하여 간단한 오류 메시지를 확인합니다.
printStackTrace()	String	오류 메시지의 발생 근원지를 찾아 단계별로 오류를 출력합니다.

2. page 디렉티브 태그를 이용한 예외 처리

[isErrorPage 속성 사용 예]

```
//error.jsp
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page isErrorPage="true"%>
<html>
<head>
<title>Exception</title>
</head>
<body>
    <p> 오류가 발생하였습니다.
    <p> 예외 유형 : <%=exception.toString()%>
</body>
</html>
```

오류가 발생하였습니다.

예외 유형 : java.lang.NullPointerException

2. page 디렉티브 태그를 이용한 예외 처리

예제 11-2 page 디렉티브 태그에 isErrorPage 속성을 이용하여 오류 페이지 만들기

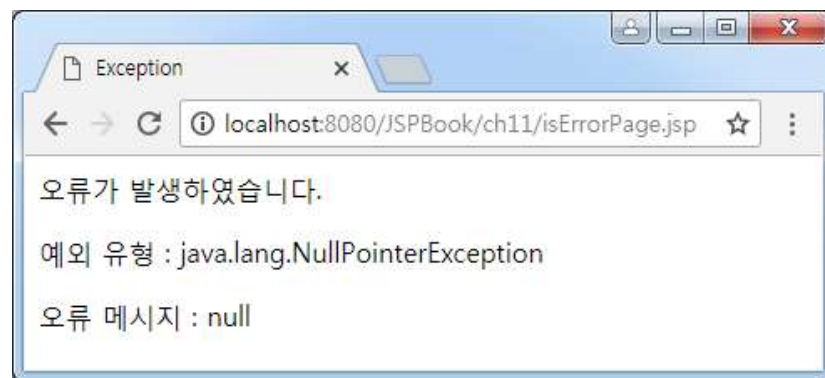
JSPBook/WebContent/ch11/isErrorPage.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page errorPage="isErrorPage_error.jsp"%>
03 <html>
04 <head>
05 <title>Exception</title>
06 </head>
07 <body>
08     name 파라미터 : <%=request.getParameter("name").toUpperCase()%>
09 </body>
10 </html>
```

2. page 디렉티브 태그를 이용한 예외 처리

JSPBook/WebContent/ch11/isErrorPage_error.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page isErrorPage="true"%>
03 <html>
04 <head>
05 <title>Exception</title>
06 </head>
07 <body>
08     <p>오류가 발생하였습니다.
09     <p> 예외 유형 : <%=exception.
        getClass().getName()%>
10     <p> 오류 메시지 : <%=exception.getMessage()%>
11 </body>
12 </html>
```

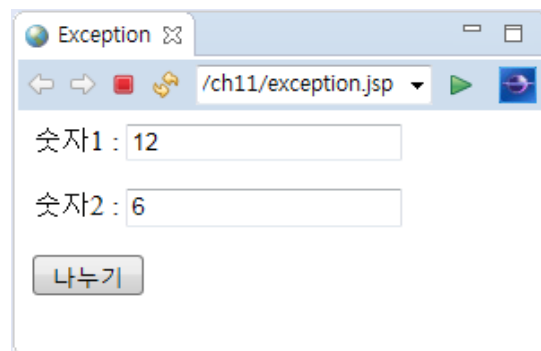


2. page 디렉티브 태그를 이용한 예외 처리

예제 11-3 page 디렉티브 태그에 `errorPage`와 `isErrorPage` 속성을 이용하여 예외 처리하기

JSPBook/WebContent/ch11/exception.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     <form action="exception_process.jsp" method="post">
08         <p> 숫자1 : <input type="text" name="num1">
09         <p> 숫자2 : <input type="text" name="num2">
10         <p> <input type="submit" value="나누기">
11     </form>
12 </body>
13 </html>
```



Exception

/ch11/exception.jsp

숫자1 : 12

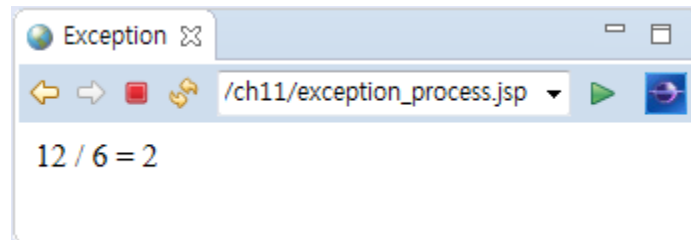
숫자2 : 6

나누기

2. page 디렉티브 태그를 이용한 예외 처리

JSPBook/WebContent/ch11/exception_process.jsp

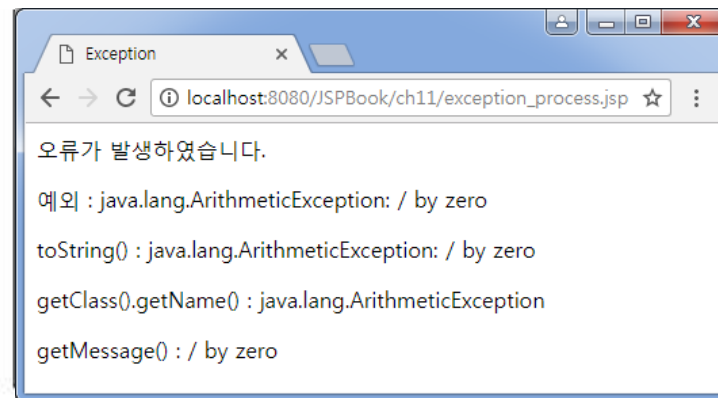
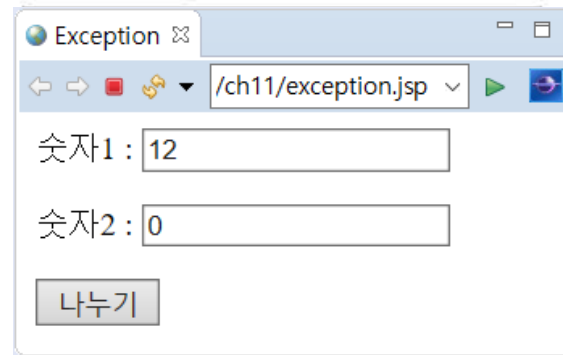
```
01 <% page contentType="text/html; charset=utf-8"%>
02 <% page errorPage="exception_error.jsp"%>
03 <html>
04 <head>
05 <title>Exception</title>
06 </head>
07 <body>
08     <%
09         String num1 = request.getParameter("num1");
10         String num2 = request.getParameter("num2");
11         int a = Integer.parseInt(num1);
12         int b = Integer.parseInt(num2);
13         int c = a / b;
14         out.print(num1 + " / " + num2 + " = " + c);
15     %>
16 </body>
17 </html>
```



2. page 디렉티브 태그를 이용한 예외 처리

JSPBook/WebContent/ch11/exception_error.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page isErrorPage="true"%>
03 <html>
04 <head>
05 <title>Exception</title>
06 </head>
07 <body>
08     <p>오류가 발생하였습니다.
09     <p> 예외 : <%=exception%>
10     <p> toString() : <%=exception.
        toString()%>
11     <p> getClass().getName() : <%=exception.
        getClass().getName()%>
12     <p> getMessage() : <%=exception.getMessage()%>
13 </body>
14 </html>
```



3. web.xml 파일을 이용한 예외 처리

❖ web.xml 파일을 이용한 예외 처리

- web.xml 파일을 통해 오류 상태와 오류 페이지를 보여주는 방법
- <error-page>...</error-page> 요소 내에 처리할 오류 코드나 오류 유형 및 오류 페이지를 호출

```
<error-page>
  <error-code>...</error-code>|<exception-type>...</error-exception-type>
  <location>...</location>
</error-page>
```

표 11-3 <error-page>를 구성하는 하위 요소

요소	설명
<error-code>	오류 코드를 설정하는 데 사용합니다.
<exception-type>	자바 예외 유형의 정규화된 클래스 이름을 설정하는 데 사용합니다.
<location>	오류 페이지의 URL을 설정하는 데 사용합니다.

- web.xml 파일은 웹 애플리케이션의 /WEB-INF/폴더에 있어야 함

3. web.xml 파일을 이용한 예외 처리

❖ 오류 코드로 오류 페이지 호출하기

- 오류 코드는 웹 서버가 제공하는 기본 오류 페이지에 나타나는 404, 500과 같이 사용자의 요청이 올바르지 않을 때 출력되는 코드로 응답 상태 코드라고도 함
- JSP 페이지에서 발생하는 오류가 web.xml 파일에 설정된 오류 코드와 일치하는 경우 오류 코드와 오류 페이지를 보여줌

❖ web.xml 파일에 오류 코드와 오류 페이지를 설정하는 형식

```
<error-page>  
  <error-code>오류 코드</error-code>  
  <location>오류 페이지의 URI</location>  
</error-page>
```

3. web.xml 파일을 이용한 예외 처리

표 11-4 주요 오류 코드의 종류

코드	설명
200	요청이 정상적으로 처리됩니다.
307	임시로 페이지가 리다이렉트됩니다.
400	클라이언트의 요청이 잘못된 구문으로 구성됩니다.
401	접근이 허용되지 않습니다.
404	지정된 URL을 처리하기 위한 자원이 존재하지 않습니다(페이지가 없음).
405	요청된 메소드가 허용되지 않습니다.
500	서버 내부의 에러입니다(JSP에서 예외가 발생하는 경우).
503	서버가 일시적으로 서비스를 제공할 수 없습니다(서버 과부하나 보수 중인 경우).

3. web.xml 파일을 이용한 예외 처리

[<error-code> 요소 사용 예]

```
<web-app...>
  <error-page>
    <error-code>404</error-code>
    <location>/errorCode_404.jsp</location>
  </error-page>

  <error-page>
    <error-code>500</error-code>
    <location>/errorCode_500.jsp</location>
  </error-page>
</web-app>
```

3. web.xml 파일을 이용한 예외 처리

예제 11-4 web.xml 파일에 오류 코드로 오류 페이지 호출하기

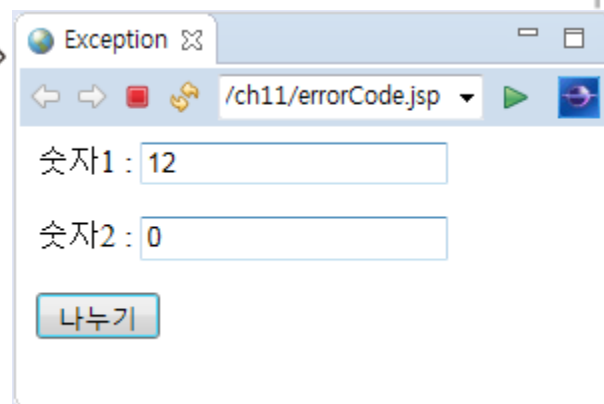
JSPBook/WebContent/WEB-INF/web.xml

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <web-app>
03     ... (생략) ...
04     <error-page>
05         <error-code>500</error-code>
06         <location>/ch11/errorCode_error.jsp</location>
07     </error-page>
08 </web-app>
```

3. web.xml 파일을 이용한 예외 처리

JSPBook/WebContent/ch11/errorCode.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     <form action="errorCode_process.jsp" method="post">
08         <p> 숫자1 : <input type="text" name="num1">
09         <p> 숫자2 : <input type="text" name="num2">
10         <p> <input type="submit" value="나누기">
11     </form>
12 </body>
13 </html>
```



Exception

/ch11/errorCode.jsp

숫자1 : 12

숫자2 : 0

나누기

3. web.xml 파일을 이용한 예외 처리

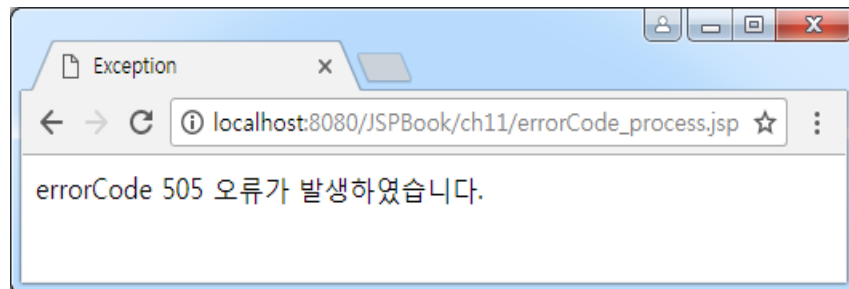
JSPBook/WebContent/ch11/errorCode_process.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     <%
08         String num1 = request.getParameter("num1");
09         String num2 = request.getParameter("num2");
10         int a = Integer.parseInt(num1);
11         int b = Integer.parseInt(num2);
12         int c = a / b;
13         out.print(num1 + " / " + num2 + " = " + c);
14     %>
15 </body>
16 </html>
```

3. web.xml 파일을 이용한 예외 처리

JSPBook/WebContent/ch11/errorCode_error.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     errorCode 505 오류가 발생하였습니다.
08 </body>
09 </html>
```



3. web.xml 파일을 이용한 예외 처리

❖ 예외 유형으로 오류 페이지 호출하기

- 예외 유형에 따른 오류 페이지 호출 방법은 JSP 페이지가 발생시키는 오류가 web.xml 파일에 설정된 예외 유형과 일치하는 경우 예외 유형과 오류 페이지를 보여줌

❖ web.xml 파일에 예외 유형과 오류 페이지를 설정하는 형식

```
<error-page>
    <exception-type>예외 유형</exception-type>
    <location>오류 페이지의 URI</location>
</error-page>
```


3. web.xml 파일을 이용한 예외 처리

표 11-5 주요 예외 유형의 종류

예외 유형	설명
ClassNotFoundException	클래스를 찾지 못했을 때 발생합니다.
NullPointerException	null 오브젝트로 접근했을 때 발생합니다.
ClassCastException	변환할 수 있는 유형으로 객체를 변환할 때 발생합니다.
OutOfMemoryException	메모리 부족으로 메모리를 확보하지 못했을 때 발생합니다.
StackOverflowError	스택 오버플로일 때 발생합니다.
ArrayIndexOutOfBoundsException	범위 밖의 배열 첨자를 설정했을 때 발생합니다.
NegativeArraySizeException	음수로 배열 크기를 설정했을 때 발생합니다.
IllegalArgumentException	부적절한 문자열을 수치로 변환하려 할 때 발생합니다.
IOException	요청된 메소드가 허용되지 않을 때 발생합니다.
NumberFormatException	부적절한 문자열을 수치로 변환하려 할 때 발생합니다.
ArithmeticException	어떤 값을 0으로 나누었을 때 발생합니다.

3. web.xml 파일을 이용한 예외 처리

[<exception-type> 요소 사용 예]

```
<web-app...>
  <error-page>
    <exception-type>java.lang.NullPointerException</exception-type>
    <location>/errorNullPointer.jsp</location>
  </error-page>
</web-app>
```

3. web.xml 파일을 이용한 예외 처리

예제 11-5 web.xml 파일에 예외 유형으로 오류 페이지 호출하기

JSPBook/WebContent/WEB-INF/web.xml

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <web-app>
03     ... (생략) ...
04     <error-page>
05         <exception-type>java.lang.Exception</exception-type>
06         <location>/ch11/exceptionType_error.jsp</location>
07     </error-page>
08 </web-app>
```

3. web.xml 파일을 이용한 예외 처리

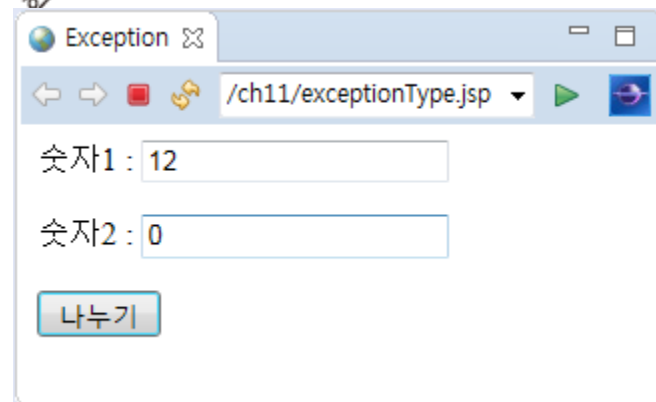
JSPBook/WebContent/ch11/exceptionType.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     <form action="exceptionType_process.jsp" method="post">
08         <p> 숫자1 : <input type="text" name="num1">
09         <p> 숫자2 : <input type="text" name="num2">
10         <p> <input type="submit" value="나누기">
11     </form>
12 </body>
13 </html>
```

3. web.xml 파일을 이용한 예외 처리

JSPBook/WebContent/ch11/exceptionType_process.jsp

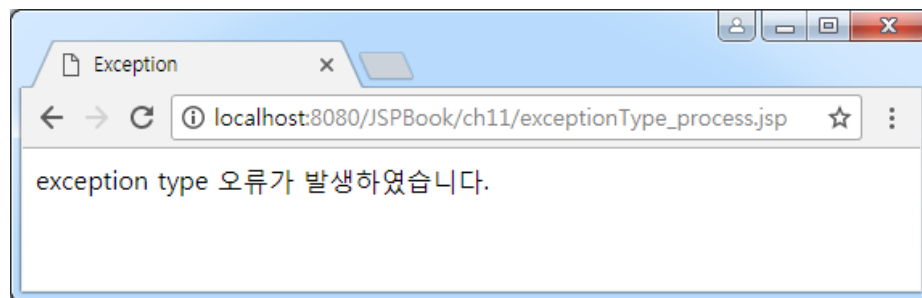
```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     <%
08         String num1 = request.getParameter("num1");
09         String num2 = request.getParameter("num2");
10         int a = Integer.parseInt(num1);
11         int b = Integer.parseInt(num2);
12         int c = a / b;
13         out.print(num1 + " / " + num2 + " = " + c);
14     %>
15 </body>
16 </html>
```



3. web.xml 파일을 이용한 예외 처리

JSPBook/WebContent/ch11/exceptionType_error.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     exception type 오류가 발생하였습니다.
08 </body>
09 </html>
```



4. try-catch-finally를 이용한 예외 처리

❖ try-catch-finally

- 자바의 예외 처리 구문으로 스크립틀릿 태그에 작성

```
try{  
    //예외가 발생할 수 있는 실행문  
}  
catch(처리할 예외 유형 설정{  
    //예외 처리문  
}  
[finally{  
    //예외와 상관없이 무조건 실행되는 문장(생략 가능)  
}])
```

- try 구문에는 예외가 발생할 수 있는 코드를 작성하고, catch 구문에는 오류가 발생할 수 있는 예외 사항을 예측하여 오류를 처리하는 코드를 작성
- finally 구문에는 try 구문이 실행된 후 실행할 코드를 작성하는데 이는 생략 가능

4. try-catch-finally를 이용한 예외 처리

[try-catch-finally 사용 예]

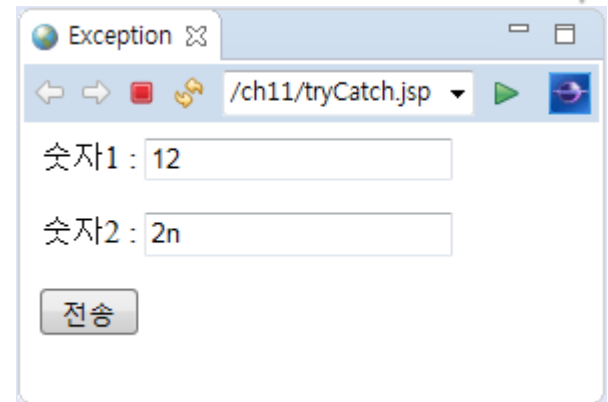
```
<%@ page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Exception</title>
</head>
<body>
    <%
        try {
            int num = 10 / 0;
        } catch (NumberFormatException e) {
            RequestDispatcher dispatcher = request.getRequestDispatcher("error.
            jsp");
            dispatcher.forward(request, response);
        }
    %>
</body>
</html>
```


4. try-catch-finally를 이용한 예외 처리

예제 11-6 try-catch-finally를 이용하여 예외 처리하기

JSPBook/WebContent/ch11/tryCatch.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     <form action="tryCatch_process.jsp" method="post">
08         <p> 숫자1 : <input type="text" name="num1">
09         <p> 숫자2 : <input type="text" name="num2">
10         <p> <input type="submit" value="전송">
11     </form>
12 </body>
13 </html>
```



Exception

/ch11/tryCatch.jsp

숫자1 : 12

숫자2 : 2n

전송

4. try-catch-finally를 이용한 예외 처리

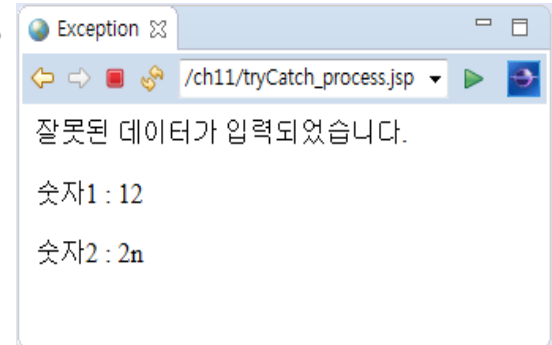
JSPBook/WebContent/ch11/tryCatch_process.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07 <%
08     try {
09         String num1 = request.getParameter("num1");
10         String num2 = request.getParameter("num2");
11         int a = Integer.parseInt(num1);
12         int b = Integer.parseInt(num2);
13         int c = a / b;
14     } catch (NumberFormatException e) {
15         RequestDispatcher dispatcher = request.getRequestDispatcher
16             ("tryCatch_error.jsp");
17         dispatcher.forward(request, response);
18     }
19 %>
20 </body>
21 </html>
```

4. try-catch-finally를 이용한 예외 처리

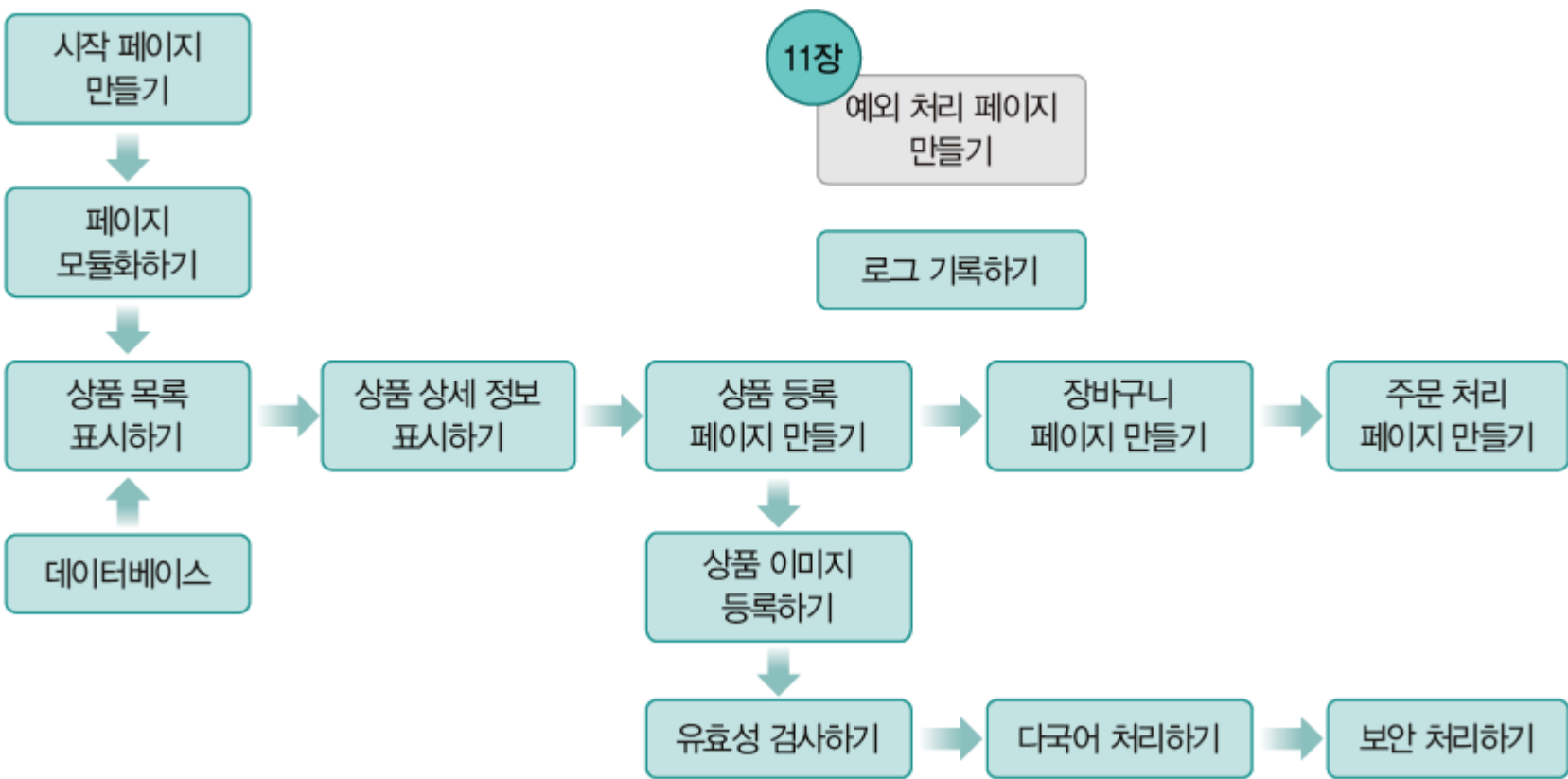
JSPBook/WebContent/ch11/tryCatch_error.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Exception</title>
05 </head>
06 <body>
07     <p>잘못된 데이터가 입력되었습니다.
08     <p> <%= " 숫자1 : " + request.getParameter("num1")%>
09     <p> <%= " 숫자2 : " + request.getParameter("num2")%>
10 </body>
11 </html>
```

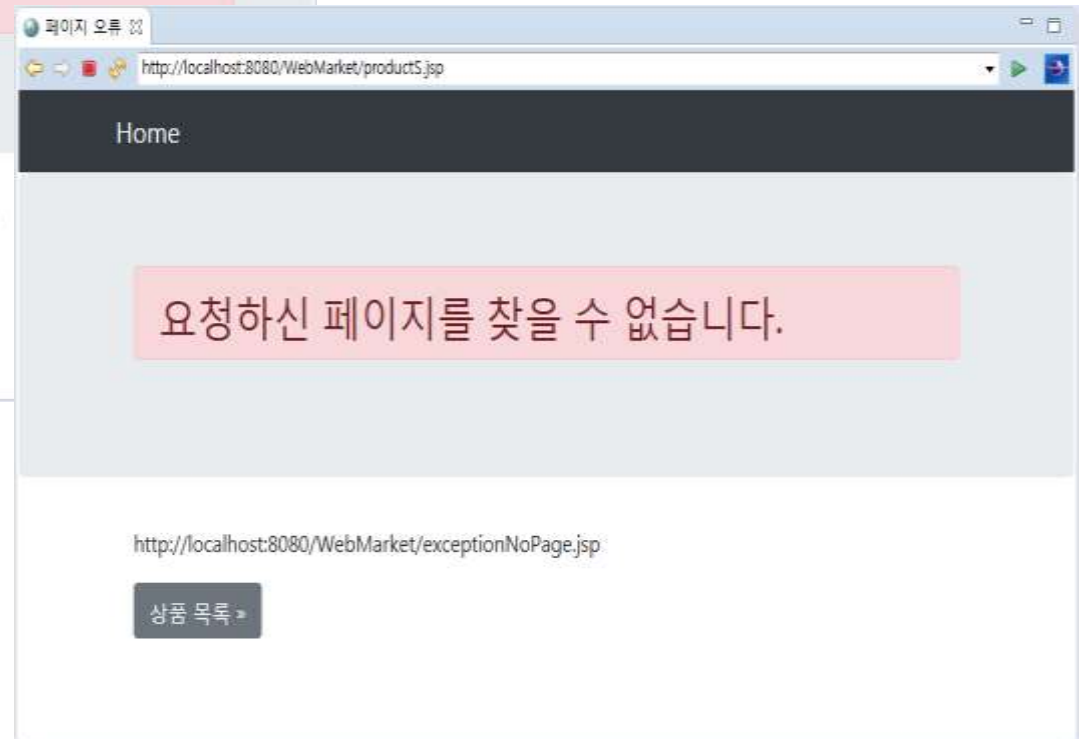
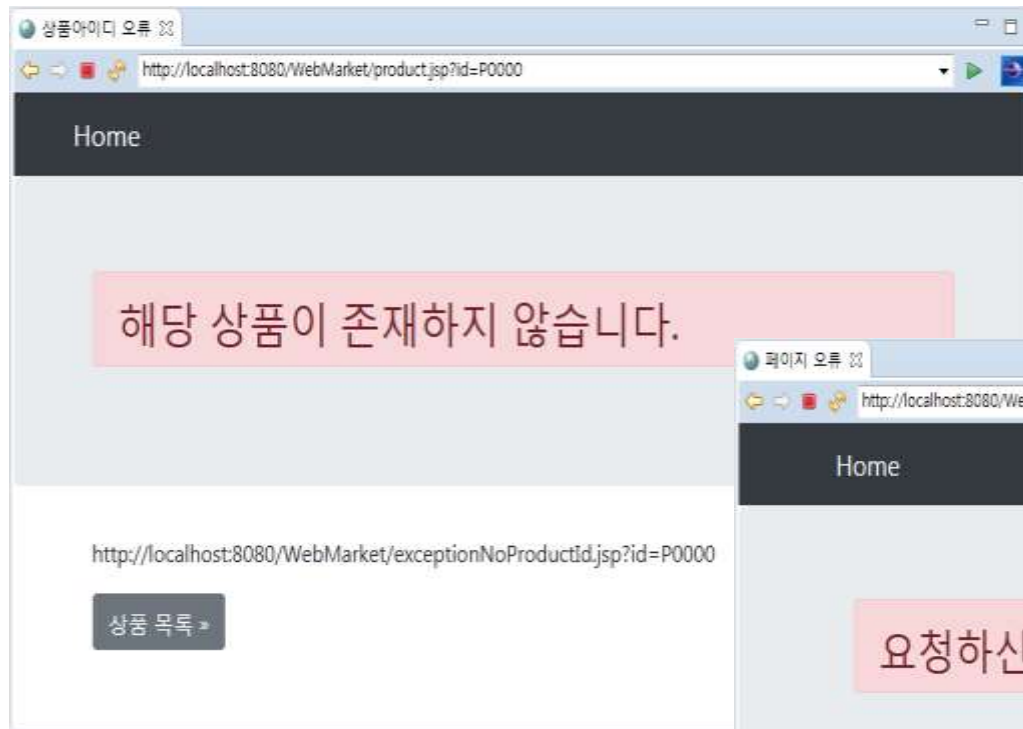


5. [웹 쇼핑몰] 예외 처리 페이지 만들기

그림 11-2 예외 처리 페이지 만들기



5. [웹 쇼핑몰] 예외 처리 페이지 만들기



5. [웹 쇼핑몰] 예외 처리 페이지 만들기

예제 11-7 page 디렉티브 태그에 errorPage 속성을 이용하여 오류 페이지 호출하기

■ 오류 페이지 작성하기

WebMarket/WebContent/exceptionNoProductId.jsp

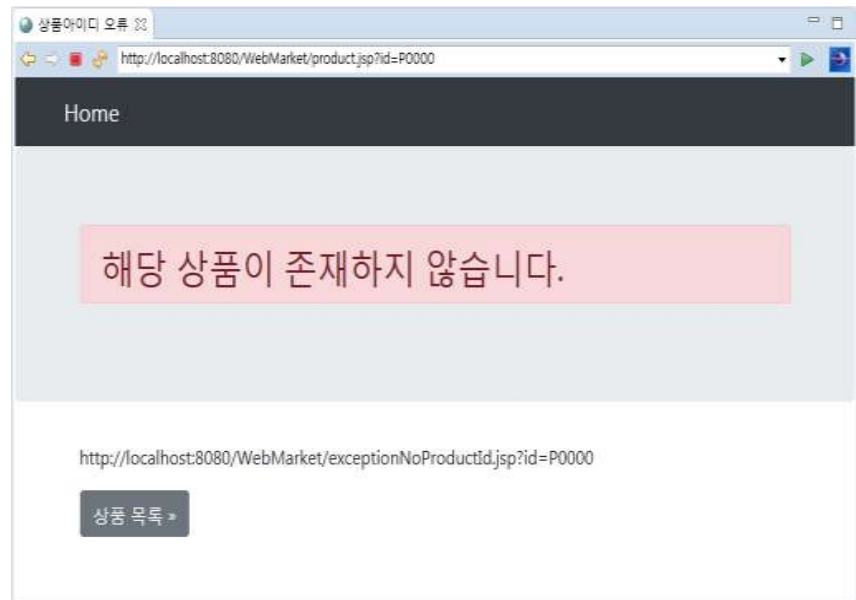
```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <link rel="stylesheet" href="./resources/css/bootstrap.min.css" />
05 <title>상품 아이디 오류</title>
06 </head>
07 <body>
08     <jsp:include page="menu.jsp" />
09     <div class="jumbotron">
10         <div class="container">
11             <h2 class="alert alert-danger">해당 상품이 존재하지 않습니다.</h2>
12         </div>
13     </div>
14     <div class="container">
15         <p>%=request.getRequestURL()%?%=request.getQueryString()%>
16         <p> <a href="products.jsp" class="btn btn-secondary"> 상품 목록
            &raquo;</a>
17     </div>
18 </body>
19 </html>
```

5. [웹 쇼핑몰] 예외 처리 페이지 만들기

- 상품 상세 보기 페이지 수정하기

WebMarket/WebContent/product.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="dto.Product"%>
03 <%@ page import="dao.ProductRepository"%>
04 <%@ page errorPage="exceptionNoProductId.jsp"%>
05 ... (생략)...
```



5. [웹 쇼핑몰] 예외 처리 페이지 만들기

예제 11-8 web.xml 파일에 오류 코드로 오류 페이지 호출하기

- web.xml 파일에 추가 작성하기

JSPBook/WebContent/WEB-INF/web.xml

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <web-app>
03     ...(생략)...
04     <error-page>
05         <error-code>404</error-code>
06         <location>/exceptionNoPage.jsp</location>
07     </error-page>
08 </web-app>
```


5. [웹 쇼핑몰] 예외 처리 페이지 만들기

■ 오류 페이지 작성하기

WebMarket/WebContent/exceptionNoPage.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <link rel="stylesheet" href="./resources/css/bootstrap.min.css" />
05 <title>페이지 오류</title>
06 </head>
07 <body>
08     <jsp:include page="menu.jsp" />
09     <div class="jumbotron">
10         <div class="container">
11             <h2 class="alert alert-danger">요청하신 페이지를 찾을 수 없습니다.</h2>
12         </div>
13     </div>
14     <div class="container">
15         <p>%=request.getRequestURL()%</p>
16         <p> <a href="products.jsp" class="btn btn-secondary">
            &raquo;</a>
17     </div>
18 </body>
19 </html>
```

