

XML

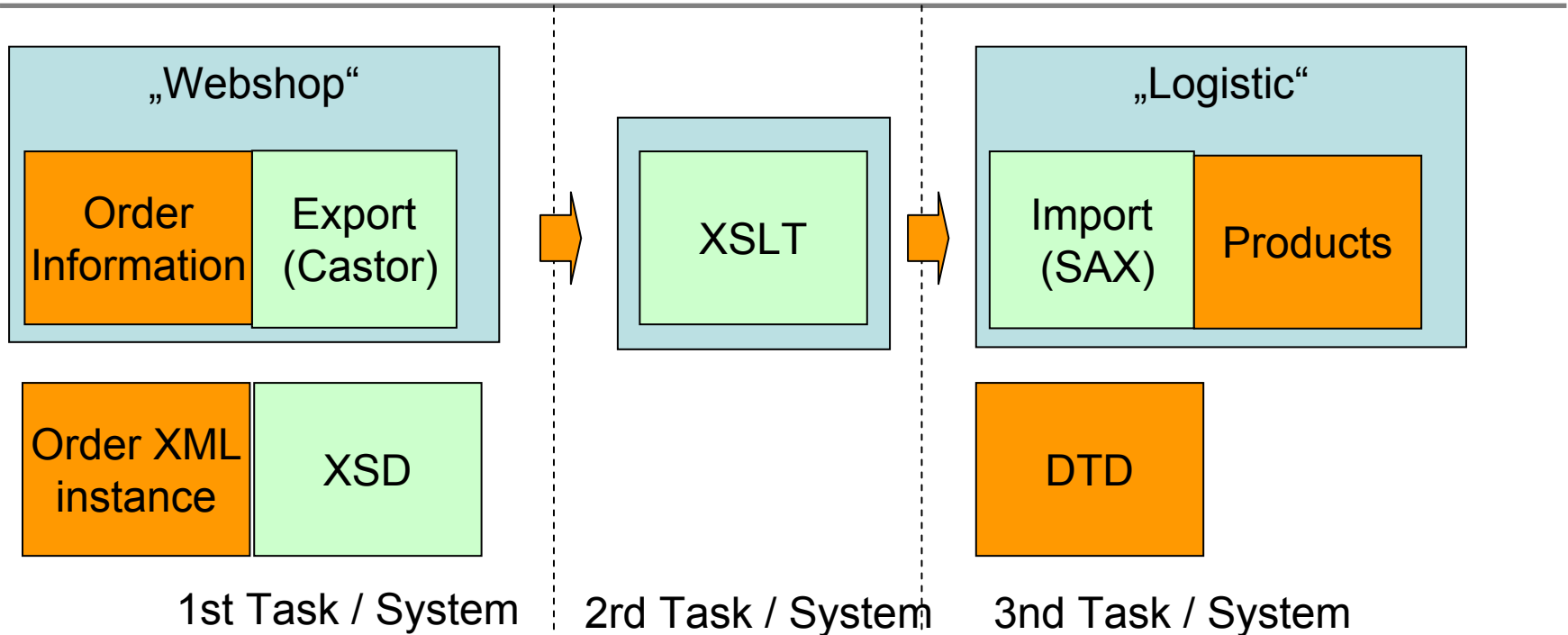
<subtitle>Practical Assignment for
XML Lecture</subtitle>

<author>Prof. Dr. Christian Pape</author>

Goal of Assignment

- Continuous example for electronic data exchange with XML
 - An order is created in a “Webshop”
 - An XML representation is created from this order and send to the target system (“Logistic”)
 - Before further processing by the target system the XML must be transformed into another XML order representation
 - Target systems reads in order and performs some business logic on its content
- Goals: learn how to
 - use Data Binding (with Castor)
 - use a SAX Parser
 - write an XML Schema (a little bit)
 - transform an XML instances with XSLT

Scenario

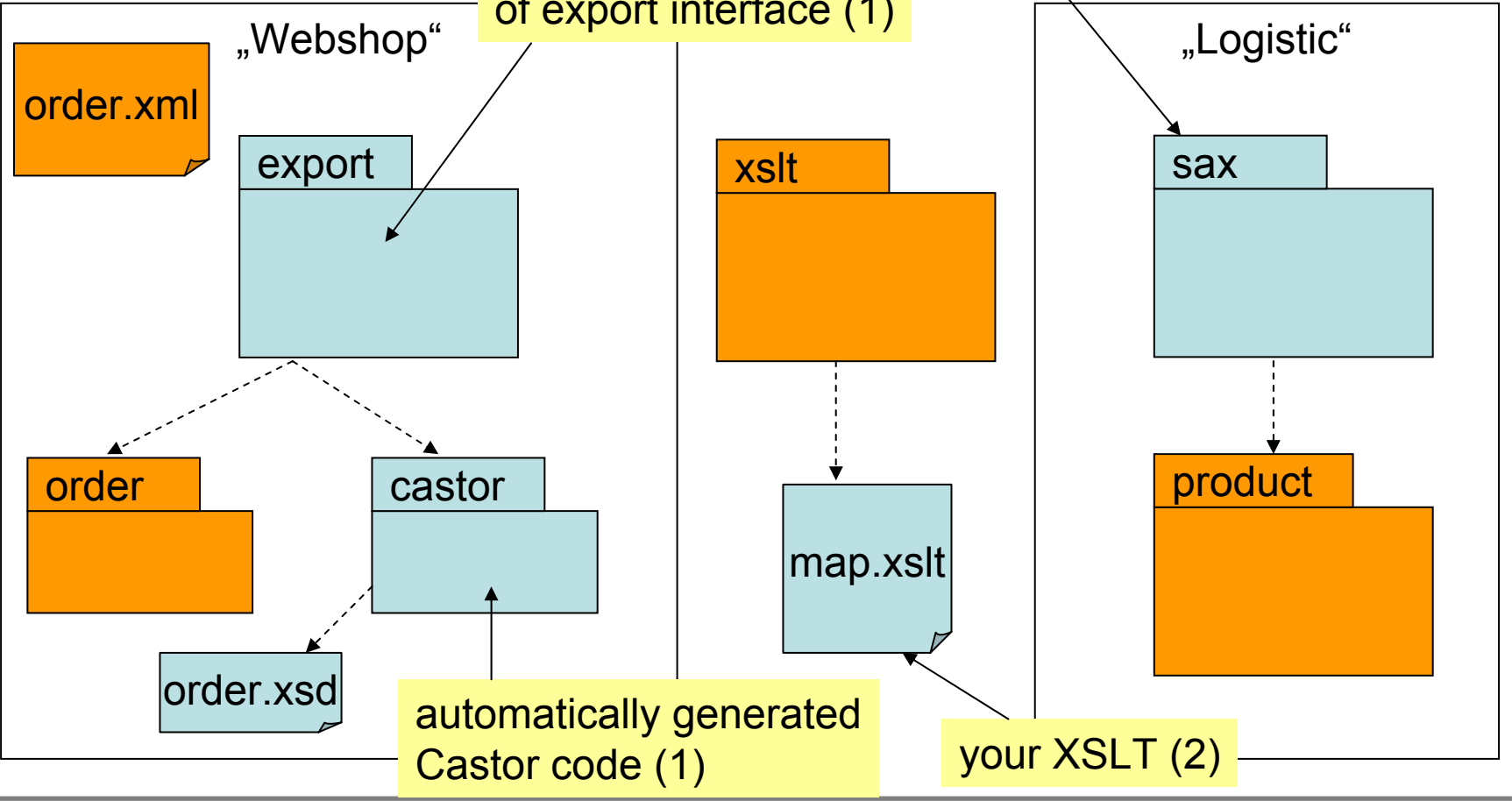


1. Design XML Schema (XSD) for order documents, use Castor and existing Order Information to create and send XML instances
2. Write XSLT that transform exported XML instance into imported XML instance
3. Write Input for Order (DTD given) with SAX ContentHandler
 - Check whether ordered products are available and check total of order
 - Check inventory: create lists of available and unavailable products

Given
Todo

Application Design

• de.hska.xml



Application Implementation: Task 1

- de.hska.xml
 - **AllInOneTest.java**: Executes all three tasks in sequence
 - Implement **de/hska/xml/order.xsd** such that your XML Schema matches the given de/hska/xml/order.xml
 - You can use the Visual Studio XML Editor for editing the XML Schema
- de.hska.xml.order
 - Contains plain Java objects for information
 - Use **CreateOrder.createOrder()** to create a fixed programmed order: you have to convert the given data structure Order to objects of the generated castor classes
- de.hska.xml.export
 - Use castor plugin to generate Castor classes into **de/hska/xml** from your XML Schema
 - Your code goes into: **OrderToCastorOrder.java**
 - Use **ExportTest.java** to test your XML export only

Application Implementation: Task 2

- `de.hska.xml.xslt`
 - Implement **`de/hska/xml/map.xslt`**
 - Use **`Xslt.java`** to test you transformation on a fixed XML order file (`de/hska/xml/order.xml`)

Application Implementation: Task 3

- `de.hska.xml.sax`
 - a) Implement **CalculateTotalContentHandler.java** (fill out missing method bodies)
 - b) Implement **AvailabilityCheckContentHandler.java**
 - Use **ImportTest.java** for testing on fixed input XML file (`de/hska/xml/order2.xml`)
- `de.hska.xml.product`
 - Contains plain Java objects for product information and availability check of products

Application Implementation

- Do not change existing source code, except those code mentioned in the previous slides
- Do not put new classes into the following packages (assume them to be applications bought by your company with only the binaries at hand)
 - de.hska.xml.order
 - de.hska.xml.product

Required Software

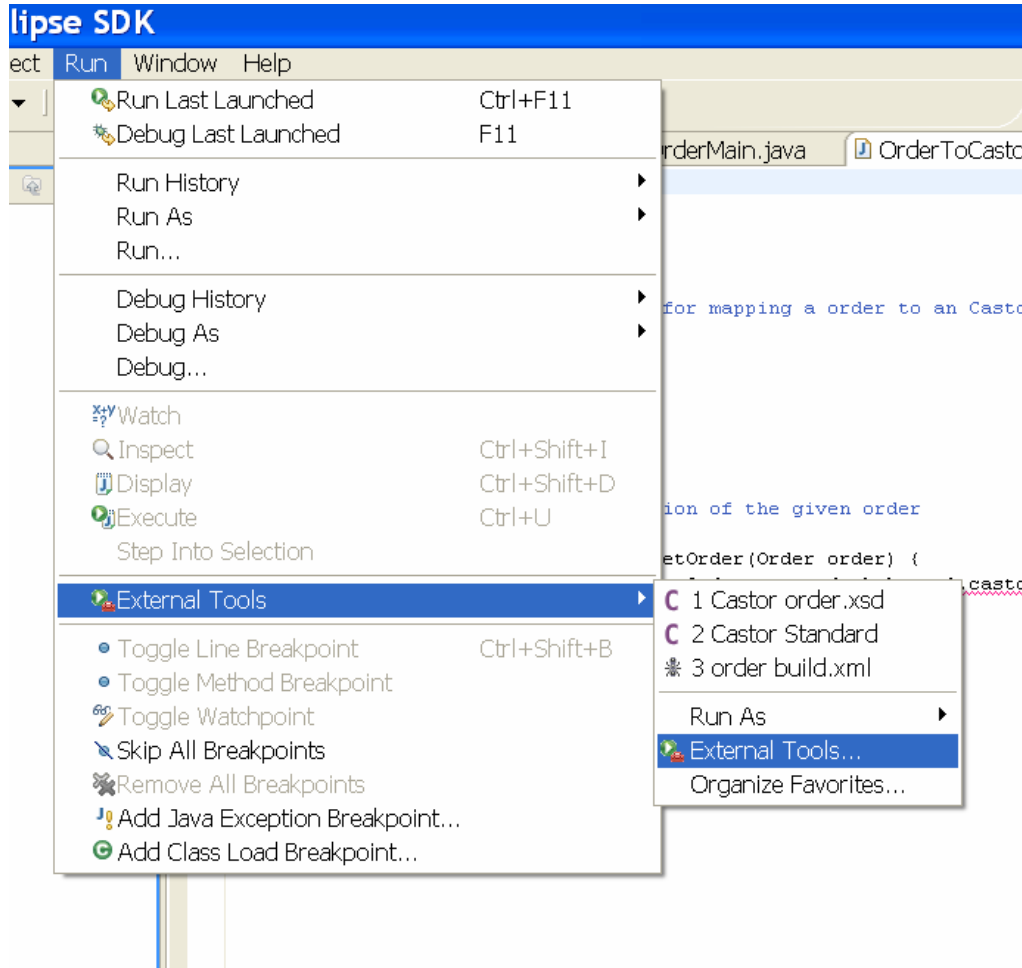
- Eclipse 3.1.x, JDK 1.5
- On your own computer
 - XML Buddy (XML, DTD Editor) Plug-in
 - <http://www.xmlbuddy.com/>
 - Free version but **without** XSLT + XML Schema support
 - Castor Plug-in (includes Castor library and Xerces)
 - <http://xdoclipse.sourceforge.net/presence/projects/castor/index.html>
 - Castor available from www.castor.org
- XSLT Implementation (part of JDK 1.5)
- SAX Parser (part of JDK 1.5)
 - Xerces already part Castor-Plug-in distribution
- ZIP file with Java classes and XML examples
 - xmlAssignment.zip from Webpage

Before starting

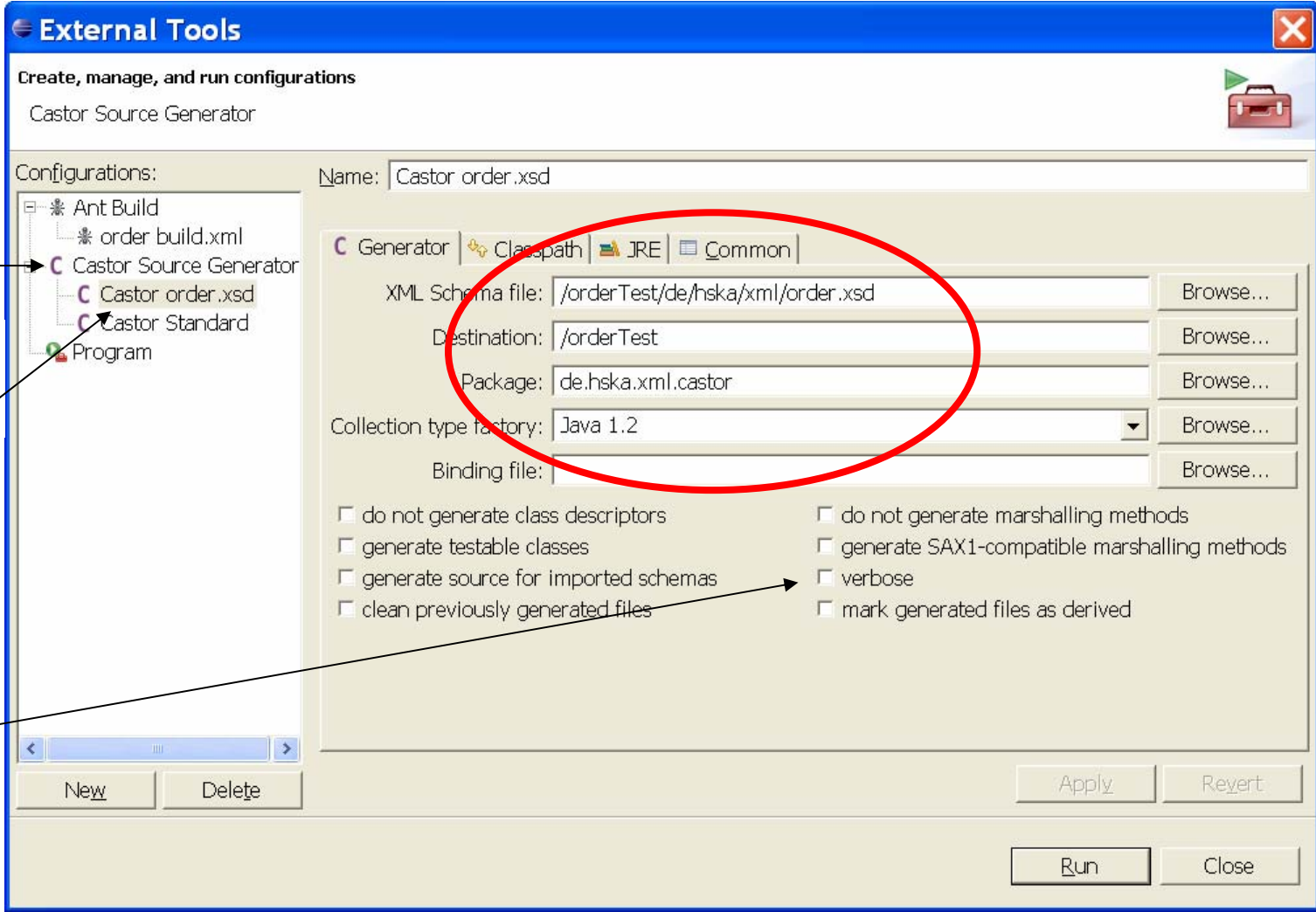
- If you are using your own computer: add plugins to Eclipse
 - For Castor: Start downloaded jar-file with Java and follow instructions
 - For other plugins: Unpack into eclipse folder, (re)start eclipse
- Create a new Java project „order“
 - Import from archive file xmlAssignment.zip to this Java project. Make sure to import the files into the projects src Folder (only newer Eclipse versions).
 - Configure Castor eclipse for this project (next slides)
 - Add JUnit classes to project (if it is missing due to import). For instance, by left click in one of the JUnit test class on the left hand side of the public class ... line and selecting „add JUnit library“

Configure Castor

- Select following from the Run menu



Setting up Castor



Double click here
to get an instance

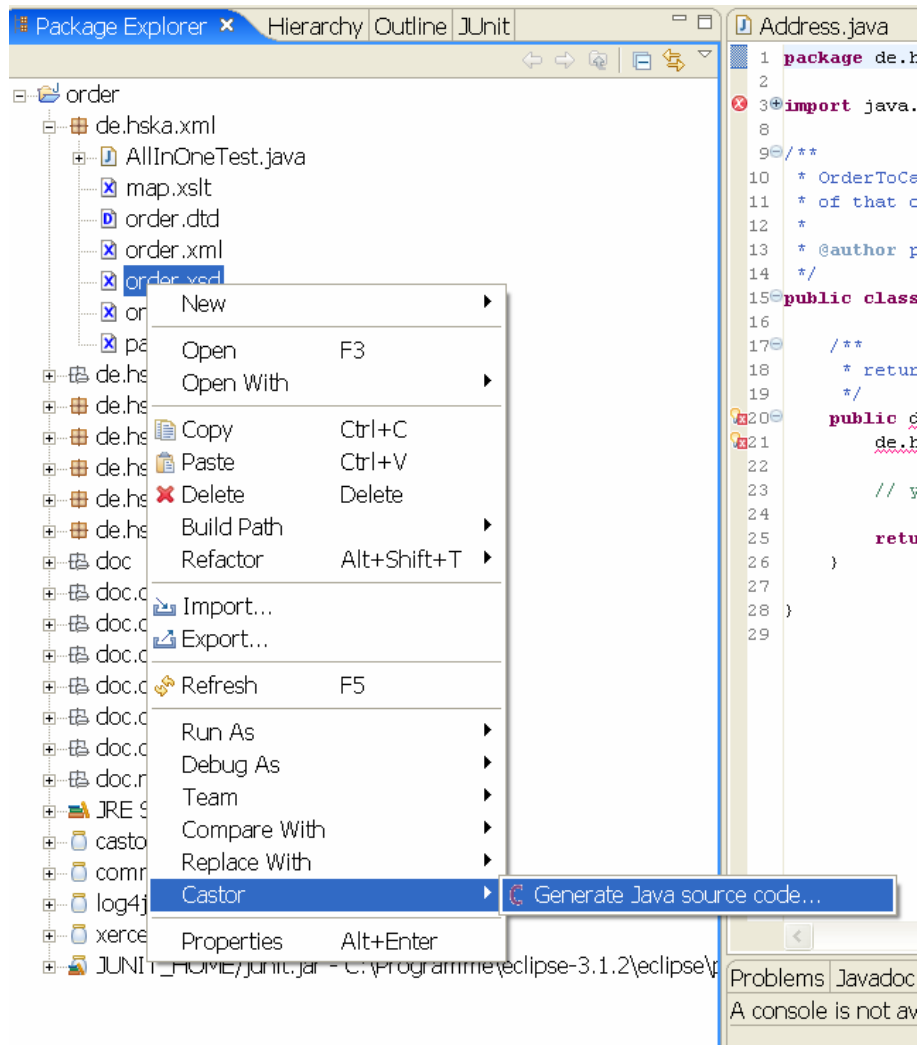
better turn Verbose option on

Adding castor libraries

- You may have to restart eclipse
- Select project and right click on it
- Select “Castor -> Add Castor libraries...”
- Add all four libraries to your project

Setting up Castor

- You may have to restart eclipse
- Selecting order.xsd with right mouse now gives a Castor entry to start source code generation



Add JUnit lib after import

The screenshot shows an IDE window with several tabs: Address.java, Item.java, CreateOrderMain.java, ordertest.xml, and Ad... The main editor displays the following code:

```
5 // **
6 * Junit test class for Address.
7 *
8 * @author pape
9 *
10 public class AddressTest extends TestCase {
11
12
13
14
15
16
17
18
19
20
21
22
23
24 public void testAddress() {
25     assertTrue(address != null);
26 }
27
```

A red circle highlights the line `public class AddressTest extends TestCase`. A yellow tooltip box is overlaid on the left, containing the text: "Add the JUnit library to the project's build class path". A blue tooltip box is overlaid on the right, containing the following actions:

- Create class 'TestCase'
- Rename in file (Ctrl+2, R direct access)
- Add JUnit libraries

Common Problems and Solutions

- **Syntax error in generated Castor classes**
 - Make sure that you added the castor classes to the projekt
 - The element order-number occurs twice in the document. If you used different data type, this results in a naming clash. Either use the same data type (recommended) or use a binding file to resolve this problem
- **Importing Project**
 - Newer Eclipse versions store the source files in a folder src. Make sure all imported files are located in this src folder (right click on src and import the archive or move the files into the src folder).