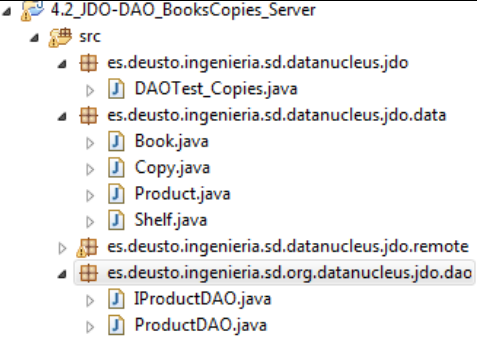



PART 1- DAO Layer testing

<p>This is an example for testing the insertion, selection, update and removal of a set of related objects.</p> <p>Book inherits from Product (Inheritance). A Book has many Copies (1:N). A Copy is stored in Shelf. A Shelf can store many Copies (M:N).</p> <p>Creating copies means adding the copy to the shelf where it will be stored.</p>	
<p>DAO Pattern and Datanucleus There is a class ProductDAO.java (implementing IProductDAO) where all the coding related to Persistence is isolated.</p> <p>Notice the code in DAOTest_Copies.java. No reference, at all, at any implementation details.</p>	<p>Edit datanucleus.properties</p> <p>Notice the annotation in the domain objects @PersistenceCapable(detachable = "true")</p> <p>Notice the annotation in the fields that need to be retrieved (because they are not fetched by default) @Persistent(defaultFetchGroup="true")</p> <p>In code, in the retrieval methods in the DAO layer = pm.getFetchPlan().setMaxFetchDepth(3);</p>
<p>Executing the example</p> <p>Start MySQL (mysqld --console --explicit_defaults_for_timestamp) Start MySQL Workbench Create the database schema using the script in db folder</p> <p>Execute ant run</p> <p>Use ant deletescema for resetting the db tables.</p>	

PART 2 - RMI + DAO Integration

<p>ProductFinderServer.java is a small RMI server that will use IProductDAO.</p> <p>Edit the remote interface IProductFinder and notice that we are transferring JDO domain objects, not special hand-made DTO.</p> <pre>public interface IProductFinder extends Remote { public List<Product> getProducts() throws RemoteException; }</pre>	
<p>Execute ant export in order to generate the library needed for the RMI client side of the application.</p> <p>This jar file must contain not only the remote interface, but also Product.class</p> <p>Sample Client source code:</p> <pre>List<Product> products = rsl.getService().getProducts();</pre> <p>Launch the rmiregistry (util folder)</p> <p>Start the server using ant runserver</p>	
<p>Go to the Client Project</p> <p>Do not forget to copy the InventoryService.jar library</p> <p>Run ant compile</p>	
<p>Compile the client and ant run</p>	<p>What happens? It seems to be a very very simple omission in our code ??????</p> <p>Edit the code Product.java. Any idea? Do not forget to revise Copy and Shelf.</p>
<p>Client-side of the Application</p> <pre>client: [java] Client looking for service : //127.0.0.1:1099/InventoryService [java] Mixture of Products and Books ... [java] Product: iPad [A standard iPad from Apple] [java] Book: Bram Stoker - Dracula [First Edition] [java] Copy details: 30, Perfect, almost new [java] In Shelf: First Shelf [java] Copy details: 45, Scratches on front cover [java] In Shelf: First Shelf BUILD SUCCESSFUL Total time: 11 seconds</pre>	<p>Server-side of the application</p> 