# PART 1- DAO Layer testing

|  |  |
| --- | --- |
| This is an example for testing the insertion, selection, update and removal of a set of related objects.  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).  Creating copies means adding the copy to the shelf where it will be stored. |  |
| DAO Pattern and Datanucleus  There is a class ProductDAO.java (implementing IProductDAO) where all the coding related to Persistence is isolated.  Notice the code in DAOTest\_Copies.java. No reference, at all, at any implementation details. | Edit datanucleus.properties  Notice the annotation in the domain objects  @PersistenceCapable(detachable = "true")  Notice the annotation in the fields that need to be retrieved (because they are not fetched by default)  @Persistent(defaultFetchGroup="true")  In code, in the retrieval methods in the DAO layer = pm.getFetchPlan().setMaxFetchDepth(3); |
| Executing the example Start MySQL (mysqld --console --explicit\_defaults\_for\_timestamp)  Start MySQL Workbench  Create the database schema using the script in db folder  Execute **ant run**  Use ant deleteschema for reseting the db tables. |  |

# PART 2 - RMI + DAO Integration

|  |  |
| --- | --- |
| ProductFinderServer.java is a small RMI server that will use IProductDAO.  Edit the remote interface IProductFinder and notice that we are tranferring **JDO domain objects**, **not special hand-made DTO**.  **public** **interface** IProductFinder **extends** Remote {  **public** List<Product> getProducts() **throws** RemoteException;  } |  |
| Execute **ant export** in order to generate the library needed for the RMI client side of the application.  This jar file must contain not only the remote interface, but also Product.class  Sample Client source code:  List<Product> products = rsl.getService().getProducts();  Launch the **rmiregistry** (util folder)  Start the server using **ant runserver** |  |
| Go to the Client Project  Do not forget to copy the InventoryService.jar library  Run **ant compile** |  |
| Compile the client and **ant run** | **What happens?**  **It seems to be a very very simple omission in our code**  **??????**  Edit the code Product.java. Any idea?  Do not forget to revise Copy and Shelf. |
| Client-side of the Application | Server-side of the application |