Notes on JDBC Code

Similarities between the classes

In all 3 classes (DisplayProduct, InsertProductStmt, and InsertProductPstmt) we wrote the same two lines of code which was:

Line 1: //1.Load the Driver for our specific database  
Class.*forName*("org.postgresql.Driver");

Line 2: //2.Create a connection to the database  
conn = DriverManager.*getConnection*("jdbc:postgresql://localhost:5432/postgres", "postgres", "root");

Not only is this repetitive and cumbersome but what happens if we have 500 classes? Are we going to write those same two lines of code in all 500 classes? What happens if we want to use a different database other than postgresql? Or even worse, what happens if the password changes from “root” to something else? Passwords for databases change very often. Are we going to go in all 500 classes and manually change the password? No. What if instead we were able to store all that information inside of one class and whenever a change needs to be made we just have to change that one class. That brings us to a specific class that negates not only the need for rewriting those two lines of code in every class but also allowing us to change the password whenever we want without having to change all the classes that use it. This class will allow us to just make the change in it and it will then carry the change to all the necessary classes. In the JDBC code we wrote we implemented just that and we called it DBConnection.

DBConnection

getConnection() - We imported the Connection class. We created a static method because we didn’t want other classes to be able to create an object of our class. This method returns a Connection object. Inside this method we write the two lines for loading the driver and creating the connection. Put it in a try/catch instead of throws because if you use throws then every class has to use throws also. Implementation of method can be seen in code