**Important Instructions:**

* Please read the document thoroughly before you code.
* Import the given skeleton code into your Eclipse.
* Do not change the Skeleton code or the package structure, method names, variable names, return types, exception clauses, access specifiers etc.
* You can create any number of private methods inside the given class.
* Use **Spring** annotations to configure, inject **beans**.
* You are provided with a **Main class** with the main method to check the correctness of the DAO methods written.
* Having completed writing the DAO methods, execute the main method and identify the result.

**Assessment Coverage:**

* Spring **JdbcTemplate** methods – update(), query() & **BeanPropertyRowMapper**
* **PlatformTransactionManager** methods – getTransaction(), commit(), rollback()

1. **Business Scenario:**

Electricity consumption is recorded in terms of kWh or Units from the electricity meter installed in the premise. A person from the electricity provider records the current reading and subtracts it from previous reading to calculate the current period consumption.

You are required to develop a Spring JDBC application to view such electricity consumption records and delete some. Detailed descriptions given below

1. **Skeleton Details:**

Following classes/interfaces fully implemented as part of the skeleton,

* com.cts.handson.AppConfiguration
* com.cts.handson.model.EBill
* com.cts.handson.util.DateUtil
* SQL file to create database & table



1. **Functional Requirements:**

The application has the below classes and methods to be implemented by you.

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Method(s)** | **Responsibilities** | **Exception** |
| EBillDAO | void deleteBill (long…billNumbers) | * Accepts one or more bill numbers as varArgs. * In try block, it iterates bill numbers and delete records from the EBILL table **if the bill number >=100** * After successful transaction, the transaction manager commits entire transaction. | If any bill number is less than 100, throw **new Exception()** and **rollback()** entire transaction in catch block. |
| EBillDAO | List<EBill> getAllBill() | * Uses BeanPropertyRowMapper and returns EBILL table records as list |  |
| ElectricityBillApplication | main() | * Get EBillDAO bean * Using dao, **delete records** (eg. 100,99) * Using dao, **retrieve** and display all EBills |  |