**Code smell** 1- Generic exceptions should never be thrown

Code snippet:

*/\*\*  
 \** ***@return*** *Java Class path for establishing JDBC connection.  
 \*/*public String getDriverClassPath() throws Error {  
 return this.driverPath**;**}

Location of the code: src/main/java/jabref/logic/remote/shared/security/DBMSType.java

In this specific piece of code, a generic exception is thrown. Using such generic exceptions as Error, RuntimeException, Throwable, and Exception prevents calling methods from handling true, system-generated exceptions differently than application-generated errors.

This code smell can easily be fixed by throwing a more specific exception.

Refactoring proposal:

*/\*\*  
 \** ***@return*** *Java Class path for establishing JDBC connection.  
 \*/*public String getDriverClassPath() throws SpecificException {  
 return this.driverPath**;** //this method throws SpecificException

}

**Code smell** 2- String literals should not be duplicated

Code snippet:

try (Statement statement = oracleConnection.createStatement()) {  
 ((OracleStatement) statement).setDatabaseChangeRegistration(databaseChangeRegistration)**;** StringBuilder selectQuery = new StringBuilder()  
 .append("SELECT 1 FROM ")  
 .append(escape("ENTRY"))  
 .append(", ")  
 .append(escape("METADATA"))**;** // this execution registers all tables mentioned in selectQuery  
 statement.executeQuery(selectQuery.toString())**;** }  
 } catch (SQLException e) {  
 *LOGGER*.error("SQL Error: "**,** e)**;** }  
}

Location of the code: src/main/java/jabref/logic/remote/shared/security/OracleProcessor.java

In this specific code the literal “SQL Error: “is repeated 4 times. It shouldn’t happen since it makes the code repetitive therefore difficult to read.

This code smell can easily be fixed by creating a constant to replace all these literals.

Refactoring proposal:

private static final SQL="SQL Error:"**;**

}  
 } catch (SQLException e) {  
 *LOGGER*.error(SQL**,** e)**;** }  
}

**Code smell** 3-

Code snippet:

@Override  
public void startNotificationListener(DBMSSynchronizer dbmsSynchronizer) {  
 // Disable cleanup output of ThreadedHousekeeper  
 // Logger.getLogger(ThreadedHousekeeper.class.getName()).setLevel(Level.SEVERE);  
 try {  
 connection.createStatement().execute("LISTEN jabrefLiveUpdate")**;** // Do not use `new PostgresSQLNotificationListener(...)` as the object has to exist continuously!  
 // Otherwise the listener is going to be deleted by GC.  
 PGConnection pgConnection = connection.unwrap(PGConnection.class)**;** listener = new PostgresSQLNotificationListener(dbmsSynchronizer**,** pgConnection)**;** JabRefExecutorService.*INSTANCE*.execute(listener)**;** } catch (SQLException e) {  
 *LOGGER*.error("SQL Error: "**,** e)**;** }  
}

Location of the code: src/main/java/jabref/logic/remote/shared/security/PostgreSQLProcessor.java

Programmers should not comment out code as it bloats programs and reduces readability.

Unused code should be deleted and can be retrieved from source control history if required.

Refactoring proposal:

The proposal would be to just delete the commented code.