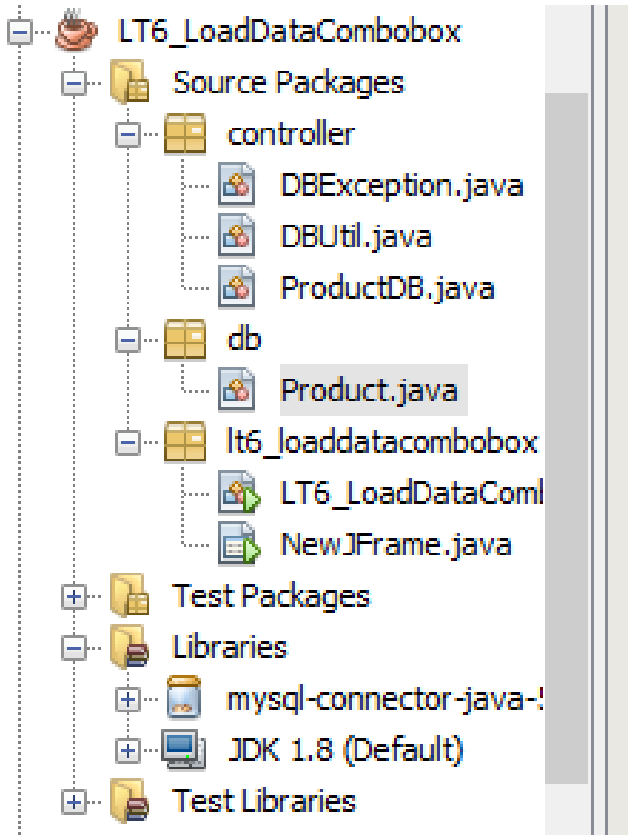
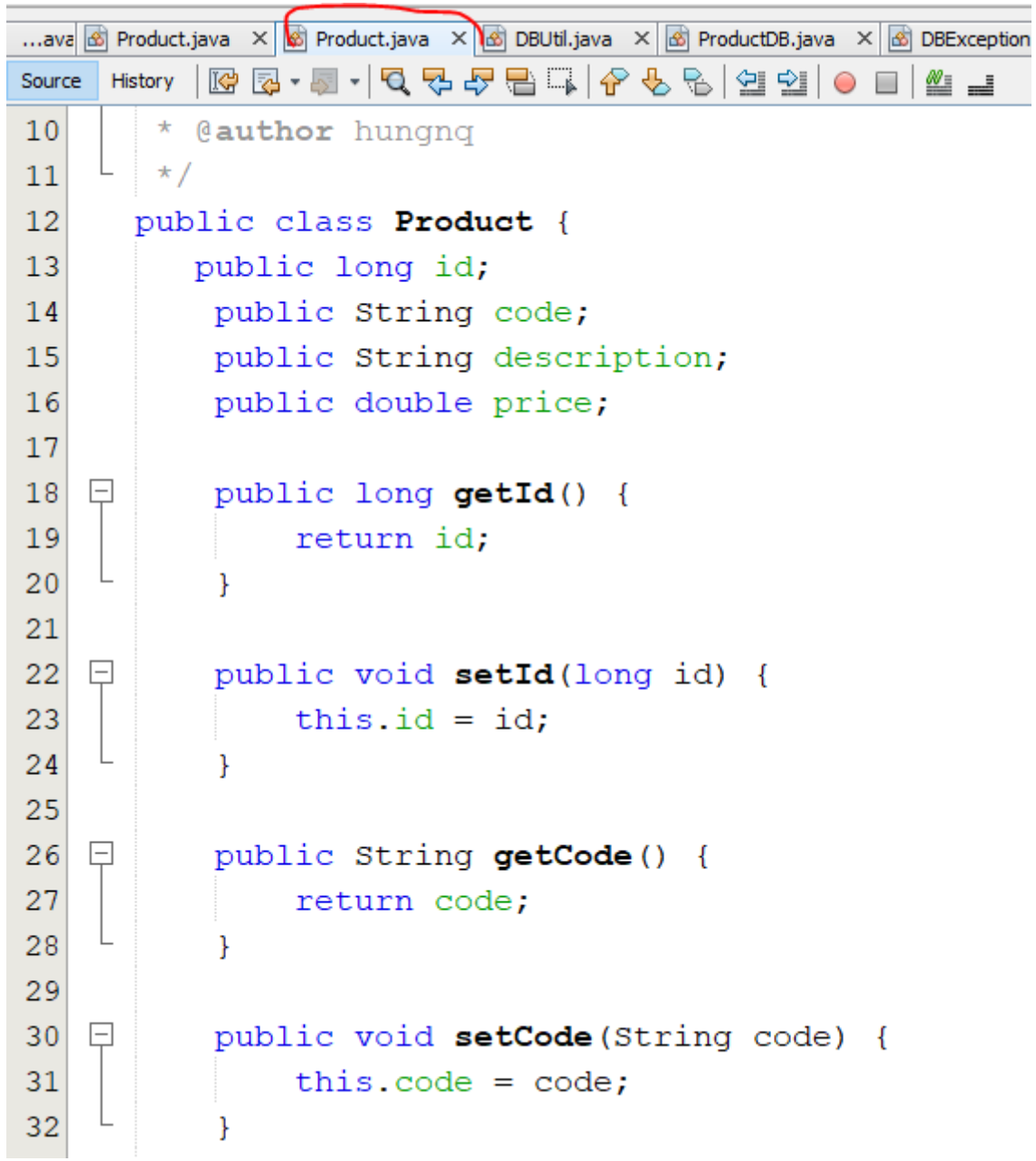
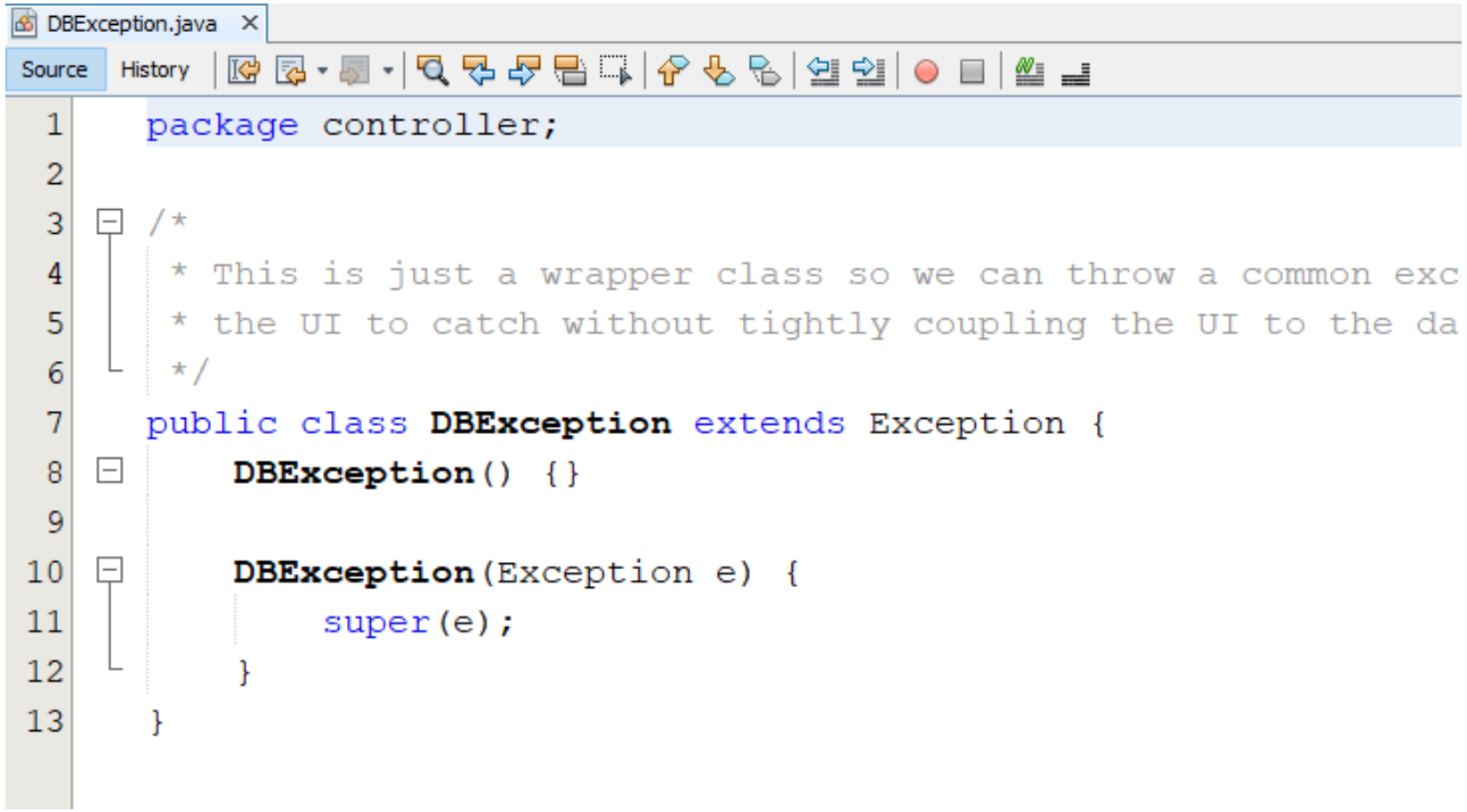
LoadDataToCombobox

```java
public static void LoadDataToCombobox(JComboBox combo) throws DBException
{
    try {
        //khai bao chuoi ket noi
        String sql = "SELECT * from product";
        //mo ket noi csdl
        Connection connection = DBUtil.getConnection();
        //chuan bi lenh doc du lieu
        PreparedStatement ps = connection.prepareStatement(sql);
        //thuc thi va luu du lieu
        ResultSet rs = ps.executeQuery(); //muon lay du lieu, k muon lay: execute
        //doc du lieu tu noi luu tru (rs)
        while (rs.next()) {
            //doc
            String code = rs.getString("Code");
            combo.addItem(code);
        }
    } catch (SQLException ex) {
        Logger.getLogger(ProductDB.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

```
 */
public class NewJFrame extends javax.swing.JFrame {

    /**
     * Creates new form NewJFrame
     */
    public NewJFrame() throws DBException {
        initComponents();

        ProductDB.LoadDataToCombobox(jComboBox1);
    }

    /**
```


LT6_LoadDataCombobox
Source Packages
controller
DBException.java
DBUtil.java
ProductDB.java
db
Product.java
lt6_loaddatacombobox
LT6_LoadDataComl
NewJFrame.java
Test Packages
Libraries
mysql-connector-java-
JDK 1.8 (Default)
Test Libraries

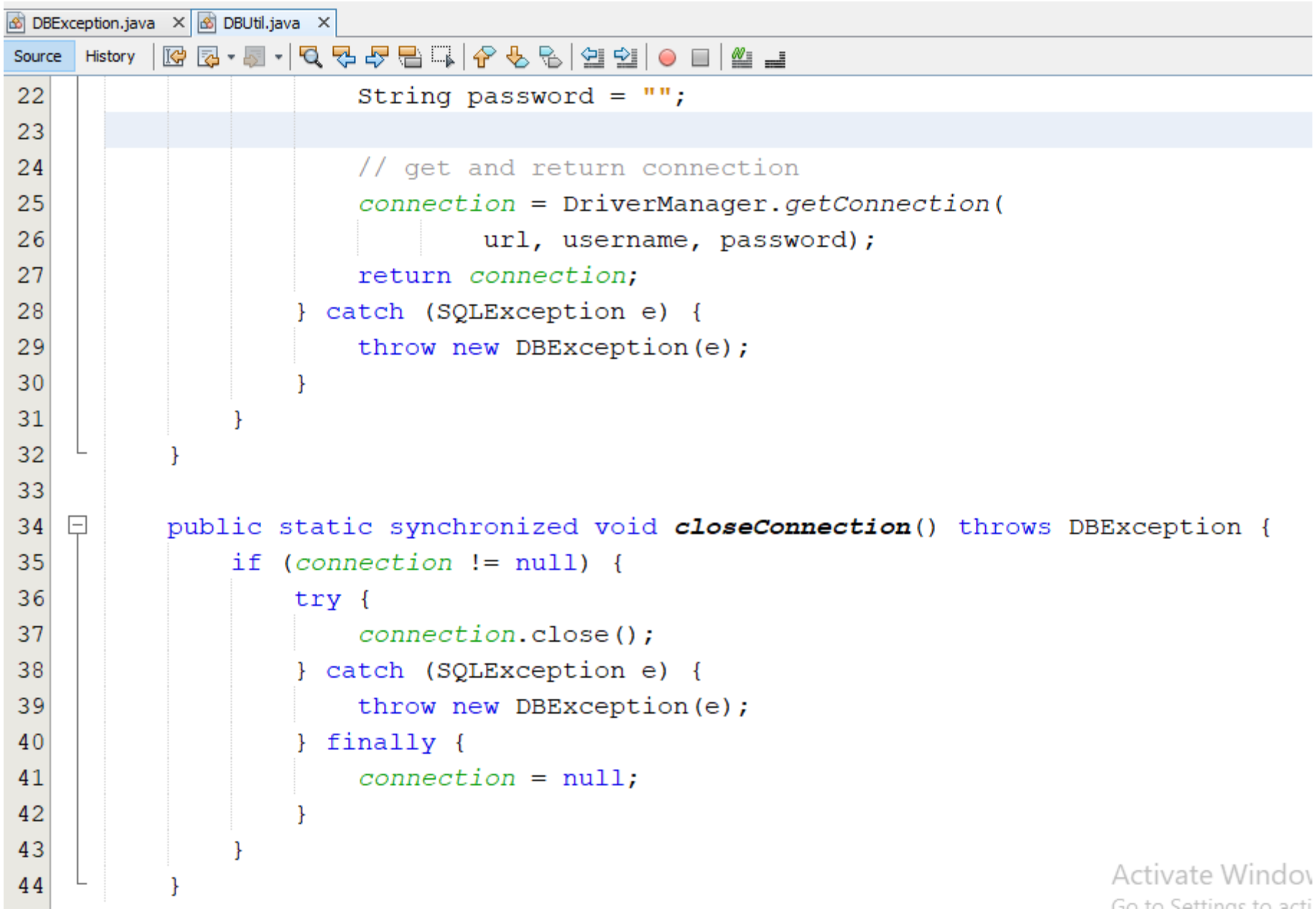

...ava
Product.java
Product.java
DBUtil.java
ProductDB.java
DBException
Source
History
10 * @author hungnq
11 */
12 public class Product {
13 public long id;
14 public String code;
15 public String description;
16 public double price;
17
18 public long getId() {
19 return id;
20 }
21
22 public void setId(long id) {
23 this.id = id;
24 }
25
26 public String getCode() {
27 return code;
28 }
29
30 public void setCode(String code) {
31 this.code = code;
32 }


DBException.java
Source
History
1 package controller;
2
3 /*
4 * This is just a wrapper class so we can throw a common exc
5 * the UI to catch without tightly coupling the UI to the da
6 */
7 public class DBException extends Exception {
8 DBException() {}
9
10 DBException(Exception e) {
11 super(e);
12 }
13 }

DBException.java | DBUtil.java

Source | History

```java
package controller;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBUtil {

    private static Connection connection;

    private DBUtil() {}

    public static synchronized Connection getConnection() throws DBException {
        if (connection != null) {
            return connection;
        }
        else {
            try {
                // set the db url, username, and password
                String url = "jdbc:mysql://localhost:3306/mma";
                String username = "root";
                String password = "";
```


DBException.java
DBUtil.java
Source
History
22 String password = "";
23
24 // get and return connection
25 connection = DriverManager.getConnection(
26 url, username, password);
27 return connection;
28 } catch (SQLException e) {
29 throw new DBException(e);
30 }
31 }
32 }
33
34 public static synchronized void closeConnection() throws DBException {
35 if (connection != null) {
36 try {
37 connection.close();
38 } catch (SQLException e) {
39 throw new DBException(e);
40 } finally {
41 connection = null;
42 }
43 }
44 }