CST8288 OOP with Design Patterns

Introduction to JDBC

Talking Points (You will need to see the readings and/or do research online if you miss lecture)

- What is JDBC?
- What are JDBC driver types?
- What are the typical components used?
  - DriverManager (and older Class.forName)
  - Connection
  - Statement or Prepared Statement
  - ResultSet
    - Concepts: read-once, forward-only, cursor, before, after
- Mapping Database data types to Java data types.
- SQLException
- finally for resource clean up (or try-with-resources)

JDBC (Simple Example, to test database and JDBC driver installation)

- **See the MySQL installation and configuration handout first.**
- Create a new project in Eclipse: "WeekXyzSimpleJDBCDemo) where Xyz is the semester's week number.
- From Chris Thayer. (2014):
  - In the Package Explorer create a new folder named "External"
  - Using the operating system Folder Explorer copy this file into the External folder:
  - C:\Program Files (x86)\\MySQL\\Connector.J 5.1\mysql-connector-java-5.1.37-bin.jar
  - Right-click on the **mysql-connector-java-5.1.37-bin.jar (or newer version number)**
    - Build Path → Add to Build Path

```java
/* File: BooksDatabaseAuthorReader.java
 * Author: Stanley Pieda
 * Date: 2015
 * Sample Simple JDBC Demo
 */
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class BooksDatabaseAuthorReader {

    public static void main(String[] args) {
        String connectionString =
                "jdbc:mysql://localhost/books";
        String username = "scott";
        String password = "tiger";
        Connection connection = null;
        PreparedStatement statement = null;
        ResultSet resultset = null;
        try{
            connection =
                DriverManager.getConnection(connectionString, username, password);
            statement = connection.prepareStatement(
                "SELECT AuthorID, FirstName, LastName FROM Authors ORDER BY AuthorID");
            resultset = statement.executeQuery();
            while(resultset.next()){
                System.out.printf("%d %s %s%n", // %n is platform appropriate new-line
                        resultset.getInt("AuthorID"),
                        resultset.getString("FirstName"),
                        resultset.getString("LastName"));
            }
        }
        catch(SQLException e){
            System.err.println("Problem accessing database");
            System.err.println(e.getMessage());
        }
        finally{ // most important lines in this entire program!!!
            try{if(resultset != null){resultset.close();}}
            catch(SQLException e){System.err.println(e.getMessage());}

            try{if(statement != null){statement.close();}}
            catch(SQLException e){System.err.println(e.getMessage());}

            try{if(connection != null){connection.close();}}
            catch(SQLException e){System.err.println(e.getMessage());}
        } // close try-catch
    } // close main
} // close class
```

Additional Example to Show ResultSet MetaData for Column Names

```
/* File: BooksDatabaseReader.java
 * Author: Stanley Pieda
 * Date: 2015
 * Description: Simple JDBC Example to test MySQL installation and sample
 *              database.
 * Note: The code in this file is based on Figure 28.23 of Deitel and Deitel (2012)
 * Deitel, Paul, Harvey Deitel. (2012). CST8110 Java How to Program (early objects) 9e,
 * 9th Edition. Pearson Learning Solutions. Chapter 28 pp 1189
 */
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;

public class BooksDatabaseReader {

    public static void main(String[] args) {
        String connectionString = "jdbc:mysql://localhost/books";
        Connection connection = null;
        Statement statement = null;
        ResultSet resultSet = null;

        try{
            connection =
                    DriverManager.getConnection(connectionString, "scott", "tiger");
            statement =
                    connection.createStatement();
            resultSet =
                    statement.executeQuery(
                    "SELECT AuthorID, FirstName, LastName FROM Authors ORDER BY AuthorID");

            System.out.println("Authors:");
            printResultSet(resultSet);
            resultSet =
                    statement.executeQuery(
                    "SELECT AuthorID, ISBN FROM AuthorISBN ORDER BY AuthorID");

            System.out.println("AuthorISBN:");
            printResultSet(resultSet);

            resultSet =
                    statement.executeQuery(
                    "SELECT ISBN, Title, EditionNumber, Copyright FROM Titles ORDER BY Title");

            System.out.println("Titles:");
            printResultSet(resultSet);
        }
        catch(SQLException ex){
            System.out.println("Problem accessing database");
            System.out.println(ex.getMessage());

        }
        finally{
            try{
                try{if(resultSet != null){ resultSet.close();}}
                 catch(SQLException e){System.err.println(e.getMessage());}
                 try{if(statement != null){ statement.close();}}
                 catch(SQLException e){System.err.println(e.getMessage());}
                 try{if(connection != null){ connection.close();}}
                 catch(SQLException e){System.err.println(e.getMessage());}
            }
            catch(Exception ex){
                System.out.println(ex.getMessage());
                ex.printStackTrace();
            } // end catch
        } // end finally
    } // end main

    private static void printResultSet(ResultSet resultSet) throws SQLException{
        ResultSetMetaData metaData = resultSet.getMetaData();
        for(int columnNumber = 1; columnNumber <= metaData.getColumnCount(); columnNumber++){
            System.out.printf("%-8s\t", metaData.getColumnName(columnNumber));
        }   // %-8s means left-justified with minimum length 8 characters (Alvin Alexander 2015)
        System.out.println();

        while(resultSet.next()){
            for(int columnNumber = 1; columnNumber <= metaData.getColumnCount(); columnNumber++){
                System.out.printf("%-8s\t", resultSet.getObject(columnNumber));
            }
            System.out.println();
        }
    }
}
```

Required Readings

Textbook Chapter 24 (Deitel and Deitel 10th Ed.)

Chris Thayer. (2014). Connecting to MySql with Connector/J. [video] Retrieved from
https://www.youtube.com/watch?v=_ySyOq-Hnpw

tutorialspoint. (2014). JDBC - Driver Types. [web page] Retrieved from
http://www.tutorialspoint.com/jdbc/jdbc-driver-types.htm
(Note: There are many other tutorial topics available on the left hand side of the page)

Oracle. (2015). 5.3 Java, JDBC and MySQL Types. [web page] Retrieved from
https://dev.mysql.com/doc/connector-j/en/connector-j-reference-type-conversions.html

Recommended Readings:

Oracle. (2015). The Java™ Tutorials. Lesson: JDBC Introduction. [website]. Retrieved from
http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html

Oracle. (2015). The Java™ Tutorials. Lesson: JDBC Basics. [website]. Retrieved from
http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html
(Note: Use the navigation strip on the left side to access all the parts of the tutorial)

JournalDev. (2014). JDBC Statement vs PreparedStatement – SQL Injection Example. [webpage]. Retrieved from
http://www.journaldev.com/2489/jdbc-statement-vs-preparedstatement-sql-injection-example

Sources Cited:

Chris Thayer. (2014). Connecting to MySql with Connector/J. [video] Retrieved from
https://www.youtube.com/watch?v=_ySyOq-Hnpw

Alvin Alexander. (2015). A printf format reference page (cheat sheet). [web page] Retrieved from
http://alvinalexander.com/programming/printf-format-cheat-sheet