**今天完成的事情：**

**由于在任务完成过程中，发现自己的基础知识的缺乏严重影响到对于任务的理解和完成，所以今后的日报分为两个部分，第一部分为任务的完成情况，第二部分为基础知识的学习情况，请师兄检查指导。**

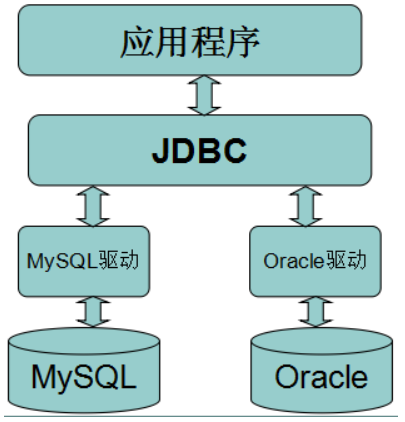
1. **JDBC（JAVA Database Connectivity）**

JAVA数据链接库是Java语言中用来规范客户端程序如何来访问数据库的应用程序接口，提供了诸如查询和更新数据库中数据的方法。JDBC也是Sun Microsystems的商标。它JDBC是面向关系型数据库的。

JDBC就是用于执行SQL语句的一类JAVA API，通过JDBC我们可以直接使用JAVA编程来对关系数据库进行操作。

通过封装，可以使开发人员使用纯JAVA API完成SQL的执行。

我们安装好数据库之后，我们的应用程序也是不能直接使用数据库的，必须要通过相应的数据库驱动程序，通过驱动程序去和数据库打交道。其实也就是数据库厂商的JDBC接口实现，即对Connection等接口的实现类的jar文件。



1. **常用接口**

**1# Driver接口**

Driver接口由数据库厂家提供，JAVA开发人员只需要使用Driver接口

在编程中要连接数据库，必须先装载特定厂商的数据库驱动程序

**不同的数据库有不同的装载方法，例如：**

装载MySql驱动：Class.forName("com.mysql.jdbc.Driver");

装载Oracle驱动：Class.forName("oracle.jdbc.driver.OracleDriver");

**2# Connection接口**

Connection与特定数据库的连接（会话）

作用是在连接的上下构件中执行sql语句并返回结果

**DriverManager.getConnection（url, user, password）是建立在JDBC URL定义的数据库Connection连接上的方法：**

连接MySql数据库：Connection conn = DriverManager.getConnection（”jdbc:mysql://host:port/database”, “user”, “password”）;

连接Oracle数据库：Connection conn = DriverManager.getConnection（”jdbc:oracle:thin:@host:port:database”, “user”, “password”）;

连接SqlServer数据库：Connection conn = DriverManager.getConnection（”jdbc:Microsoft:sqlserver://host:port;DatabaseName=database”,“user”, “password”）;

**常用方法：**

* createStatement()：创建向数据库发送sql的statement对象。
* prepareStatement(sql) ：创建向数据库发送预编译sql的PrepareSatement对象。
* prepareCall(sql)：创建执行存储过程的callableStatement对象。
* setAutoCommit(boolean autoCommit)：设置事务是否自动提交。
* commit() ：在链接上提交事务。
* rollback() ：在此链接上回滚事务。

**3# Statement接口**

Statement接口用于执行静态SQL语句并返回它所生成结果的对象。

**三种Statement接口：**

* Statement：由createStatement创建，用于发送简单的SQL语句（不带参数）
* PreparedStatement：继承自Statement接口，由preparedStatement创建，用于发送含有一个或多个参数的SQL语句。

**// PreparedStatement对象比Statement对象的效率更高，并且可以防止SQL注入，所以我们一般都使用PreparedStatement //**

* CallableStatement：继承自PreparedStatement接口，由方法prepareCall创建，用于调用存储过程。

**常用Statement接口：**

* execute(String sql):运行语句，返回是否有结果集
* executeQuery（String sql）：运行select语句，返回ResultSet结果集
* executeUpdate（String sql)：运行insert/update/delete操作，返回更新的行数
* addBatch（String sql） ：把多条sql语句放到一个批处理中
* executeBatch（）：向数据库发送一批sql语句执行

**4# ResultSet接口**

**ResultSet提供检索不同类型字段的方法，常用的有：**

* getString(int index)、getString(String columnName)：

获得在数据库里是varchar、char等类型的数据对象

* getFloat(int index)、getFloat(String columnName)：

获得在数据库里是Float类型的数据对象

* getDate(int index)、getDate(String columnName)：

获得在数据库里是Date类型的数据

* getBoolean(int index)、getBoolean(String columnName)：

获得在数据库里是Boolean类型的数据

* getObject(int index)、getObject(String columnName)：

获取在数据库里任意类型的数据

**ResultSet还提供了对结果集进行滚动的方法：**

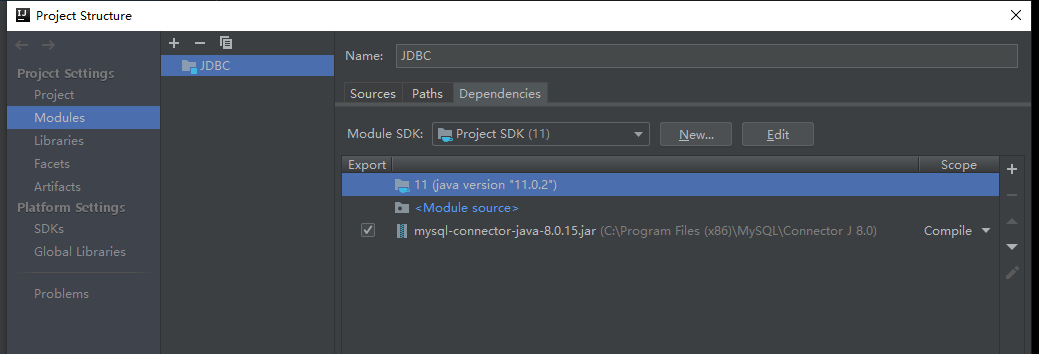
* next()：移动到下一行
* Previous()：移动到前一行
* absolute(int row)：移动到指定行
* beforeFirst()：移动resultSet的最前面
* afterLast() ：移动到resultSet的最后面

**使用后依次关闭对象及连接：ResultSet → Statement → Connection**

1. **准备工作**

配置JDBC数据库对应的jar包

使用IDEA->File->Project Structure->Modules->Dependencies->点+添加jar包



JDBC数据库jar包：mysql-connector-java-8.0.15.jar

在安装配置MySQL时选择安装

1. **JDBC代码编写**

**1# 加载驱动程序：Class.forName(driverclass)**

**加载mysql驱动：Class.forName(“com.mysql.jdbc.Driver”);**

**加载oracle驱动：Class.forName(“oracle.jdbc.driver.OracleDriver”);**

**2# 获得数据库连接**

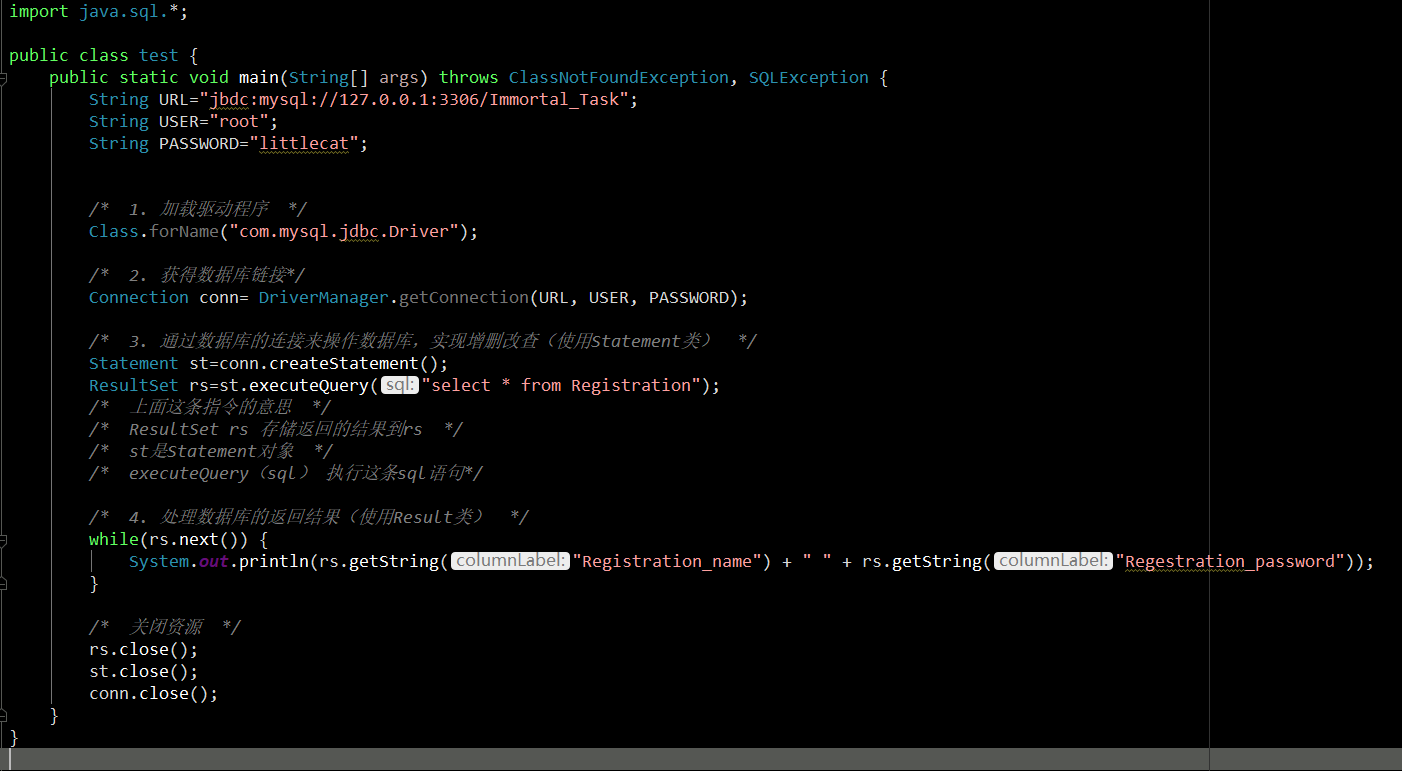
**DriverManager.getConnection(“jdbc:mysql://127.0.0.1:3306/immortal\_task”, user, password);**

**DriverManager.getConnection(URL, user, password);**

**3# 创建Statement类对象: conn.createStatement();**

**4# 向数据库发送SQL指令**

**5# 处理数据库的返回结果（ResultSet类）**



1. **问题**

编译通过但是产生了3个警告

1# Warning:java: 源值1.5已过时, 将在未来所有发行版中删除

2# Warning:java: 目标值值1.5已过时, 将在未来所有发行版中删除

3# Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.

问题原因是驱动版本的不同导致的错误

4# Exception in thread "main" java.sql.SQLException: No suitable driver found for jbdc:mysql://127.0.0.1:3306/immortal\_task

问题原因是程序没有加载驱动程序jar包，尽管在lib上已经包含了这个jar包。

1. **解决方案**

1# & 2#

网上搜索代码跟新Maven的settings和pom的文件即可

3# 将`com.mysql.jdbc.Driver' 改成 `com.mysql.cj.jdbc.Driver' 即可

是由于新驱动的语法不一样导致

4# =。= 由于拼写错误导致的问题。。。。但是收获是看到师兄解决错误的方法，学到了第一视角的操作，深受教诲

1. **JDBC类库功能调用**

在尝试建立类库来实现调用类的方式进行JDBC的功能实现

但是发现建立类的JAVA基本知识很缺乏

所以把JAVA基础知识提到第一位

通过1中对JDBC功能的简单实现，初步了解了JDBC的功能

建立类库这一任务，待明天学习JAVA知识后完成

1. **JAVA基础知识学习**

时间有限，就不将学习内容贴到今天的日报中

但是实际编写代码后，发现自己对JAVA的基本语法和规则知识很缺乏

既决定在完成任务之余，认真学习JAVA语法知识

**明天计划完成的事情：**

1. JAVA基本语法知识的学习
2. JDBC类库功能的建立与调用

完成数据库的增删改查功能

**遇到的问题：**

第一次编写JAVA代码，编译时报了4个错，最后一个粗心拼写错误花了最多时间排查

辛苦师兄！顺便跟着你排查BUG也学习了一些知识，谢谢指导！

**收获：**

感谢师兄耐心地帮我排查错误，跟师兄交流中也学会一些技巧，还有学习的方向。