JDBC

1. 连接数据库

1.1. String username = "C##SCOTT";//用户名

String password = "123"; // 密码

String driver = "oracle.jdbc.driver.OracleDriver";// 声明驱动字符串

String url = "jdbc:oracle:thin:@localhost:1521:orcl";// 获取数据的URL

Class.forName(driver);

Connection connection = DriverManager.getConnection(url, username, password);//

创建连接

2. 对象持久化

2.1. 用一个JavaBean规范的类来和数据库表对应,实现数据的持久化

1.变量私有;

2.getter/setter 方法;

3.构造函数

3. Connection

3.1. 获取连接

4. PreparedStatement

4.1. PreparedStatement是Statement接口的子接口，属于预处理操作，与直接使用Statement不同，PreparedStatement在操作时，是预先在数据表中准备好了一条SQL语句，但是此SQL语句的具体内容暂时不设置，而是之后在进行设置。

5. CallableStatement

5.1. CallableStatement 对象为所有的 DBMS

提供了一种以标准形式调用已储存过程的方法。已储存过程储存在数据库中。对已储存过程的调用是 CallableStatement 对象所含的内容。

5.2. IN 和 OUT 和IN OUT参数

1.将 IN 参数传给 CallableStatement 对象是通过 setXXX 方法完成的。该方法继承自 PreparedStatement。所传入参数的类型决定了所用的 setXXX 方法

2.如果已储存过程返回 OUT 参数，则在执行 CallableStatement 对象以前必须先注册每个 OUT 参数的 JDBC 类型（这是必需的，因为某些 DBMS 要求 JDBC 类型）。注册 JDBC 类型是用 registerOutParameter 方法来完成的。语句执行完后，CallableStatement 的 getXXX 方法将取回参数值。

3.既支持输入又接受输出的参数（INOUT 参数）除了调用 registerOutParameter 方法外，还要求调用适当的 setXXX 方法（该方法是从 PreparedStatement 继承来的）。setXXX 方法将参数值设置为输入参数，而 registerOutParameter 方法将它的 JDBC 类型注册为输出参数。setXXX 方法提供一个 Java 值，而驱动程序先把这个值转换为 JDBC 值，然后将它送到数据库中。

5.3.

```
conn = DBU.get();
cs = conn.prepareCall("{call my_pro2(?,?)}");
cs.registerOutParameter(1, oracle.jdbc.OracleTypes.CURSOR);
cs.registerOutParameter(2, oracle.jdbc.OracleTypes.CURSOR);
cs.execute();
rs = (ResultSet) cs.getObject(1);
```

6. ResultSet

6.1. 当执行的语句是查询语句时, resultSet对象用于封装查询结果.

6.2. `rs = (ResultSet) cs.getObject(1);`

```
while (rs.next()) {
   Emp emp = new Emp();
   emp.setEmpno(rs.getInt(1));
   emp.setEname(rs.getString(2));
   emp.setJob(rs.getString(3));
   emp.setMgr(rs.getInt(4));
   emp.setHiredate(rs.getDate(5));
   emp.setSal(rs.getDouble(6));
   emp.setComm(rs.getDouble(7));
   emp.setDeptno(rs.getInt(8));
   list.add(emp);
}
rs = (ResultSet) cs.getObject(2);
while (rs.next()) {
   list1.add(rs.getInt(1));
}
```

6.3.

```
cs = conn.prepareCall("{call showPage(?,?,?,?,?)}");
cs.setInt(1, 1);
cs.setInt(2, 3);
cs.registerOutParameter(3, oracle.jdbc.OracleTypes.NUMBER);
cs.registerOutParameter(4, oracle.jdbc.OracleTypes.NUMBER);
cs.registerOutParameter(5, oracle.jdbc.OracleTypes.CURSOR);
cs.execute();
list1.add(cs.getInt(3));
list1.add(cs.getInt(4));
rs = (ResultSet) cs.getObject(5);
while (rs.next()) {
  Emp emp = new Emp();
```

```
        emp.setEmpno(rs.getInt(1));
        emp.setEname(rs.getString(2));
        list.add(emp);
    }
        map.put("list1", list);
        map.put("list2", list1);
```