JDBC编程步骤：

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");

获得数据库连接：

DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/"数据库名称","admin","password");

创建Statement对象：conn.createStatement();

采用MVC三层架构

View(视图层)

Control(控制层)

Model(模型层)

DB(数据库)

视图层

流程：程序启动后，一直保持在运行状态

循环接收控制台的输入参数

调用Action（控制层）响应，并将返回结果展示在控制台中

直到输入特定的输入标记（如EXIT）后，程序退出

问题点：

1.循环接收参数

2.某个功能的保持

一、JDBC常用的API深入详解及存储过程的调用

二、JDBC的事务管理

三、数据库连接池

1、dbcp 2、c3p0

四、JDBC的替代产品

mybatis

JDBC调用无参数存储过程

JDBC调用含输入参数存储过程

JDBC调用含输出参数存储过程

JDBC调用含输入、输出参存储过程

1、调用无参存储过程

CREATE PROCEDUER imooc\_db.sp\_select\_nofilter()

BEGIN

select\*from imooc\_goddess;

END;

调用代码：

Connection conn = JDBC.getConnection();

CallableStatement c = conn.prepareCall("call sp\_select\_nofilter()");

c.setString(1,sp\_name);

c.execute();

ResultSet rs = c.getResultSet();

2、调用带输入参数的存储过程

CREATE DEFINER='imooc'@'localhost' PROCEDURE 'sp\_select\_filter'(IN sp\_name CHAR（10））

BEGIN

IF sp\_name IS UNLL OR sp\_name = '' THEN

...

END

调用代码：

String sp\_name = "参数";

List<Goddess> result = null;

result = dao.select\_filter(sp\_name);

调用带输出参数的存储过程

CREATE DEFINER='imooc'@'localhost' PROCEDURE 'sp\_select\_filter'(OUT count INT（10））

BEGIN

SELECT count(\*) INTO count FROM imooc\_goddess;

END

调用代码：

Integer count = 0;

//1.获得连接

Connection conn = JDBC.getConnection();

//2.获得callableStatement

CallableStatement c = conn.prepareCall("call sp\_select\_nofilter(?)");

c.registerOutParameter(1,Types.INTEGER);

//3.执行存储过程

c.execute();

//处理返回的结果：结果集，出参

count = c.getInt(1);

return count;

JDBC事务的概念和特点

事务的概念：

事务是作为单个逻辑工作单元执行的一系列操作。

这些操作作为一个整体一起向系统提交，要么都执行、要么都不执行。

事务的特点：

原子性：事务是一个完整的操作

一致性：当事务完成时，数据必须处于一直状态。

隔离性：对数据进行修改的所有并发事务是彼此隔离的。

永久性：事务完成后，它对数据库的修改被永久保持。

JDBC对事务管理的支持

1. 我们通过提交commit（）或是回退rollback（）来管理事务的操作
2. 事务操作默认是自动提交
3. 可以通过调用setAutoCommit（false）来禁止自动提交

JDBC升级之连接池

常用的开源数据库连接池

1. dbcp 2、c3p0



通过DBCP连接池方式操纵数据库比一般方式快

C3p0

C3p0是一个开源的JDBC连接池，它实现了数据源和JNDI绑定，支持JDBC3和JDBC2的标准扩展。









