

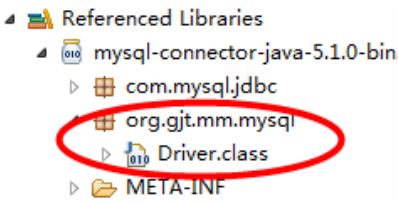
单元 10 个性化通讯录 (一)

单元教学进度设计

教学环节	教学内容	教师活动	学生活动	时间									
检查与反馈	<p>1. 有关 JDBC 的选项正确的是哪一个?</p> <p>A. JDBC 是一种被设计成通用的数据库连接技术, JDBC 技术不光可以应用在 Java 程序里面, 还可以用在 C++ 这样的程序里面。</p> <p>B. JDBC 技术是 SUN 公司设计出来专门用在连接 Oracle 数据库的技术, 连接其他的数据库只能采用微软的 ODBC 解决方案</p> <p>C. 微软的 ODBC 和 SUN 公司的 JDBC 解决方案都能实现跨平台使用, 只是 JDBC 的性能要高于 ODBC</p> <p>D. JDBC 只是个抽象的调用规范, 底层程序实际上要依赖于每种数据库的驱动文件</p> <p>答案: D</p> <p>解析:</p> <p>1. JDBC (Java Data Base Connectivity, java 数据库连接) 是一种用于执行 SQL 语句的 Java API, 可以为多种关系数据库提供统一访问, 它由一组用 Java 语言编写的类和接口组成。JDBC 提供了一种基准, 据此可以构建更高级的工具和接口。</p> <p>2. ODBC: ODBC (Open Database Connectivity, 开放数据库互连) 是微软公司开放服务结构 (WOSA, Windows Open Services Architecture) 中有关数据库的一个组成部分, 它建立了一组规范, 并提供了一组对数据库访问的标准 API (应用程序编程接口)。</p> <p>3. Java 不能直接使用 ODBC。因为 ODBC 使用 C 语言接口, 如果让 Java 来调用本机 C 代码的话, 那么会在安全、属性、健壮性、应用的可移植性等方面带来困难。</p> <p>4. JDBC 的 Java API 提供"纯 Java"的解决办法。当使用 ODBC 时, ODBC 驱动器管理程序与驱动器必须手工地装入到每架客户机上。而 JDBC 驱动器全部是用 Java 编写的, JDBC 代码则在所有 Java 平台 (从网络计算机到主机) 上都可自动安装, 并且是可移植的和安全的。</p>	提问解析	抢答	5									
	<p>检查: 小组互查是否已经建立了数据库表, 是否已经存入了初始数据表 Friend</p> <table border="1"> <thead> <tr> <th>编号</th> <th>字段名</th> <th>字段类型</th> <th>约束</th> <th>备注</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	编号	字段名	字段类型	约束	备注						检查	互查
编号	字段名	字段类型	约束	备注									

	1.	id	int	primary key auto_increment	主键					
	2.	name	varchar(20)	not null	姓名					
	3.	dept	varchar(50)		单位					
	4.	mark	varchar(20)		标识					
	5.	phone	varchar(20)		手机					
	6.	email	varchar(30)		邮箱					
	7.	tel	varchar(20)		固定电话					
	8.	workGrou p	varchar(20)		工作组					
	9.	address	varchar(50)		地址					
	10.	note	text		备注					
sql 语句如下: <pre> create table Friend(id int primary key auto_increment, name varchar(20) not null, dept varchar(50) , mark varchar(20) , phone varchar(20) , email varchar(30) , tel varchar(20) , workGroup varchar(20), address varchar(50) , note text) type="myIsam"; </pre>										
id	name	dept	mark	phone	email	tel	workGroup	address	note	
1.	李庆	15 网 3	李庆	17862341659	17862341659@163.com		1 组	文泽园 1 号楼 502		
2.	刘磊	15 网 3	刘磊	17862342386	17862342386@163.com		1 组	文泽园 1 号楼 502		
3.	陈丹妮	15 网 3	妮妮	17806214223	17806214223@163.com		2 组	文泽园 2 号楼 204		
4.	董银娟	15 网 3	小娟	17862792656	17862792656@163.com		2 组	文泽园 2 号楼 204		
教学环节	教学内容							教师活动	学生活动	时间
新课导入	在《我爱大明星》的程序中，我们在第四个单元中，将明星的信息存储到了 Excel 表中，在程序运行时，从 Excel 表中将数据读取出来（演							提出问题	讨论	

	<p>示程序运行），装入到主程序的数组中，这样做可以随时修改明星的信息，或者更改明星的信息。将数据存放在 Excel 表中是一种常见的方式，在一些微型程序中可以应用，但是当数据量非常大，或者需要复杂的查询的时候，这种方式明显就不能满足我们的需求了。</p> <p>我们学过了数据库基础这门课程，知道数据库具有非常强大的数据存储和检索的功能，我们可以考虑，如果将数据存储数据库中，不但可以实现 Excel 表的永久存储功能，还能够弥补 Excel 在存储数据、检索等方面的缺陷。</p>	分析	思考	
任务一	<p>任务引入</p> <p>要想实现从数据库中读写数据，首先要建立数据库连接，建立连接后还要断开连接。</p>	布置任务	思考	
	<p>任务部署</p> <p>JDBC 驱动是什么？</p> <p>JDBC 驱动程序实现 JDBC API 定义的接口和数据库服务器交互。例如，使用 JDBC 驱动程序使您可以打开数据库连接，并与它进行交互，通过发送 SQL 或数据库命令，然后接收结果返回到 Java 程序。</p> <p>java.sql 包中附带的 JDBC 包含各种类定义他们的行为和他们的实际实现第三方的驱动程序。第三方供应商在他们的数据库驱动程序实现了 java.sql.Driver 接口。</p>			2 5
	<p>任务实施</p> <p>1. 导入外部 Jar 包</p> <p>mysql-connector-java-5.1.0-bin</p> <p>2. 建立连接</p> <pre> public void getConn(){ try{ //指定驱动程序 Class.forName("org.gjt.mm.mysql.Driver"); String url="jdbc:mysql://localhost:3306/xuanke"; Connection conn=DriverManager.getConnection(url,"root","8819568"); }catch(Exception ex){ ex.printStackTrace(); } } </pre> <p>与mysql数据库建立jdbc连接 与本地数据库通过3306端口建立连接</p> <p>指定驱动程序</p> <p>与xuanke数据库建立连接</p> <p>建立连接</p> <p>用户名 密码</p> <p>获得连接Connection</p>	辅导	编程	
<p>3. 测试连接是否成功</p>				

	<pre> public void getConn(){ try{ Class.forName("org.gjt.mm.mysql.Driver"); String url="jdbc:mysql://localhost:3306/java"; Connection conn=DriverManager.getConnection(url,"root","3819568"); System.out.println("success get Connection ! "); }catch(Exception ex){ ex.printStackTrace(); } } </pre> <p>↓ 建立连接后打印反馈信息。 → 如果建立失败，则输出错误信息。</p>			
	<p>学生编程处理</p>	<p>辅导</p>	<p>编程</p>	
<p>错误情境</p>	<p>错误 1:</p> <pre> java.lang.ClassNotFoundException: org.gjt.mm.mysql.Driver at java.net.URLClassLoader\$1.run(Unknown Source) at java.security.AccessController.doPrivileged(Native Method) at java.net.URLClassLoader.findClass(Unknown Source) at java.lang.ClassLoader.loadClass(Unknown Source) at sun.misc.Launcher\$AppClassLoader.loadClass(Unknown Source) at java.lang.ClassLoader.loadClass(Unknown Source) at java.lang.Class.forName0(Native Method) at java.lang.Class.forName(Unknown Source) at unit1.practice.TestConnect.getConn(TestConnect.java:8) at unit1.practice.TestConnect.main(TestConnect.java:19) </pre> <p>错误原因：无法找到驱动程序。</p> <p>处理 1：检查是否有导入 jar 包。</p>  <p>圆圈所圈出的 Drive 类，是 mysql 的 jdbc 驱动程序。</p> <p>处理 2：驱动程序指定错误。</p> <pre> org.gjt.mm.mysql.Driver </pre>			
	<pre> java.sql.SQLException: Access denied for user 'root'@'localhost' (using password: YES) at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:1056) at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:957) at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:2938) </pre> <p>错误原因：密码输入错误。</p> <p>处理：确定密码。</p>			
	<pre> com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Unknown database 'java: at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source) at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source) at java.lang.reflect.Constructor.newInstance(Unknown Source) at com.mysql.jdbc.Util.handleNewInstance(Util.java:353) at com.mvsql.idbc.Util.getInstance(Util.iava:336) </pre> <p>错误原因：找不到要连接的数据库。</p>			

		处理：新建数据库或修改要连接数据库的名称。			
	任务执行	<p>断开连接：判断连接是否存在，如果有连接再关闭连接。</p> <pre> public void closeConn(Connection conn){ if(conn!=null){ → 如果有连接 try{ conn.close(); → 关闭连接 conn=null; → 把连接设置为null }catch(Exception ex){ ex.printStackTrace(); } } } </pre>			
任务二	任务引入	插入一个联系人信息	引入	思考	
	任务部署	<p>操作数据库的步骤</p> <ol style="list-style-type: none"> (1) 建立连接 (2) 给出 sql 语句 (3) 建立数据库连接桥上的小车 (4) 执行 sql 语句 (5) 获得执行结果。 <p>注意：将所有的操作都放在 try……catch 中。</p>	讲解总结	思考	2
	任务执行	<p>1. 修改建立连接的方法，将建立的连接返回。</p> <pre> public Connection getConnection(){ try{ Class.forName("org.gjt.mm.mysql.Driver"); String url="jdbc:mysql://localhost:3306/java1"; Connection conn=DriverManager.getConnection(url,"root","3819568"); System.out.println("success get Connection !"); return conn; }catch(Exception ex){ ex.printStackTrace(); return null; } } </pre>	编程	思考 计算	0

	<pre> public void testInsert(){ try{ 1. Connection conn=this.getConnection(); 2. String sql="insert into AddressBook(name,dept,mark,phone) " + "values('张三','15级软件1班','张三','13568459876)"; 3. Statement stmt=conn.createStatement(); 4. int i=stmt.executeUpdate(sql); 5. System.out.println("成果插入了"+i+"条记录"); 6. this.closeConn(conn); }catch(Exception ex){ ex.printStackTrace(); } } </pre>	<p>演示 讲解</p>	<p>学习 修改</p>	
	<p>2. 执行 sql 语句</p> <p>SQL 共分为 2 类，一类是 select 语句，一类是非 select 语句。只要不是 select 语句，执行结果都是整数。</p> <pre> public void testInsert(){ try{ 1.建立连接 Connection conn=this.getConnection(); 2.给出sql语句 String sql="insert into AddressBook(name,dept,mark,phone) " + "values('张三','15级软件1班','张三','13568459876)"; Statement stmt=conn.createStatement(); 3.在连接上有一辆叫Statement的小车，用来传递信息 int i=stmt.executeUpdate(sql); System.out.println("成果插入了"+i+"条记录"); 4.执行sql语句 this.closeConn(conn); 5.处理执行结果 }catch(Exception ex){ 6.关闭连接 ex.printStackTrace(); 注意：对数据库的操作放在try catch中。 } } </pre>			
	<pre> mysql> select * from addressBook; +----+-----+-----+-----+-----+-----+-----+-----+-----+ id name dept mark phone email tel workGroup address +----+-----+-----+-----+-----+-----+-----+-----+-----+ 1 张三 15级软件1班 张三 13568459876 NULL NULL NULL NULL 2 张三 15级软件1班 张三 13568459876 NULL NULL NULL NULL </pre>	<p>演示 讲解</p>	<p>学习 记录</p>	
	<p>学生练习。</p>	<p>辅导</p>	<p>编程</p>	
<p>任务 引入</p>	<p>读取某个联系人的信息。</p>	<p>引入</p>	<p>思考</p>	
<p>任务 三</p>	<p>重复操作数据库的步骤</p> <p>(1) 建立连接</p> <p>(2) 给出 sql 语句</p>	<p>引导</p>	<p>思考 讨论</p>	<p>2 0</p>

	<p>(3) 建立数据库连接桥上的小车</p> <p>(4) 执行 sql 语句</p> <p>(5) 获得执行结果。</p> <p>注意：将所有的操作都放在 try……catch 中。</p> <p>当执行 sql 语句时，返回值放在 ResultSet 中。</p>			
<p>任务实施</p>	<pre> public void getItem(){ try{ Connection conn=this.getConnection(); String sql="select * from AddressBook where id=1"; Statement stmt=conn.createStatement(); ResultSet rs=stmt.executeQuery(sql); if(rs.next()){ System.out.println(rs.getString("name")+"\t"+rs.getString("dept")); } this.closeConn(conn); }catch(Exception ex){ ex.printStackTrace(); } } </pre> <p>注意： 对操作系统的操作，严格按照五步骤。</p>	<p>演示讲解</p>	<p>学习记录</p>	
	<p>根据讲解，学生做实验。</p>			
<p>错误情境</p>	<p>1. 如果检索一条不存在的记录，则没有执行结果反馈。程序也不报错。</p> <pre> <terminated> TestConnect [Java Application] C:\Proc success get Connection ! </pre> <pre> success get Connection ! java.sql.SQLException: Column 'names' not found. at com.mysql.jdbc.SQLException.createSQLException(SQLError.java:1056) at com.mysql.jdbc.SQLException.createSQLException(SQLError.java:957) at com.mysql.jdbc.SQLException.createSQLException(SQLError.java:927) at com.mysql.jdbc.ResultSet.findColumn(ResultSet.java:1064) at com.mysql.jdbc.ResultSet.getString(ResultSet.java:5685) at unit1.practice.TestConnect.getItem(TestConnect.java:62) at unit1.practice.TestConnect.main(TestConnect.java:71) </pre> <p>错误原因：表格中不存在名为 names 的字段。</p>	<p>演示讲解</p>	<p>学习记录</p>	

	<pre> public void getOneItem2(){ try{ Connection conn=this.getConnection(); String sql="select * from AddressBook where id=1"; Statement stmt=conn.createStatement(); ResultSet rs=stmt.executeQuery(sql); if(rs.next()){ System.out.println(rs.getString(2)+"\t"+rs.getString(3)); } this.closeConn(conn); }catch(Exception ex){ ex.printStackTrace(); } } </pre> <p>从结果集中，按照序号取出数据。 思考：在检索时，那种方式更好？</p>	<p>演示 讲解</p>	<p>学习 记录</p>																																				
<p>任务 导入</p>	<p>读取若干条联系人信息</p> <pre> public void getOneItems(){ try{ Connection conn=this.getConnection(); String sql="select * from AddressBook"; Statement stmt=conn.createStatement(); ResultSet rs=stmt.executeQuery(sql); while(rs.next()){ System.out.println(rs.getString(2)+"\t"+rs.getString(3)); } this.closeConn(conn); }catch(Exception ex){ ex.printStackTrace(); } } </pre> <p>怎样理解结果集？</p> <table border="1" data-bbox="311 1467 933 1792"> <thead> <tr> <th>Sno</th> <th>Sname</th> <th>Ssex</th> <th>Sage</th> <th>Sdept</th> </tr> </thead> <tbody> <tr> <td>S01</td> <td>王建平</td> <td>男</td> <td>21</td> <td>自动化</td> </tr> <tr> <td>S02</td> <td>刘华</td> <td>女</td> <td>19</td> <td>自动化</td> </tr> <tr style="border: 2px solid red;"> <td>S03</td> <td>范林军</td> <td>女</td> <td>18</td> <td>计算机</td> </tr> <tr> <td>S04</td> <td>李伟</td> <td>男</td> <td>19</td> <td>数学</td> </tr> <tr> <td>S05</td> <td>黄河</td> <td>男</td> <td>18</td> <td>数学</td> </tr> <tr> <td>S06</td> <td>长江</td> <td>男</td> <td>20</td> <td>数学</td> </tr> </tbody> </table> <p>1. rs.next () 做了两件事： (1) 判断是否有下一条记录 (2) 如果有下一条记录，则游标向下移动一行，指向当前行。 2. rs.getString () 干什么？ 当鼠标指向了当前行，则依次读取各字段的值，在读取时，可以按照字</p>	Sno	Sname	Ssex	Sage	Sdept	S01	王建平	男	21	自动化	S02	刘华	女	19	自动化	S03	范林军	女	18	计算机	S04	李伟	男	19	数学	S05	黄河	男	18	数学	S06	长江	男	20	数学	<p>演示 讲解</p>	<p>学习 记录</p>	<p>1 5</p>
Sno	Sname	Ssex	Sage	Sdept																																			
S01	王建平	男	21	自动化																																			
S02	刘华	女	19	自动化																																			
S03	范林军	女	18	计算机																																			
S04	李伟	男	19	数学																																			
S05	黄河	男	18	数学																																			
S06	长江	男	20	数学																																			

	<p>段名读取，也可以按照字段的编号读取。按照字段编号读取时，注意字段编号从 1 开始数。</p>			
	学生练习	辅导	编程	
错误情境	<p>success get Connection ! com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'selecte * from AddressBook' at line 1 at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source) at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source) at java.lang.reflect.Constructor.newInstance(Unknown Source) at com.mysql.jdbc.Util.handleNewInstance(Util.java:353) at com.mysql.jdbc.Util.getInstance(Util.java:336) at com.mysql.jdbc.SQLError.createSQLException(SQL_Error.java:1031) at com.mysql.jdbc.SQLError.createSQLException(SQL_Error.java:957) at com.mysql.jdbc.MySQLIO.checkErrorPacket(MySQLIO.java:2938) at com.mysql.jdbc.MySQLIO.sendCommand(MySQLIO.java:1601)</p> <p>com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'selecte * from AddressBook' at line 1</p> <p>MySQLSyntaxErrorException:sql 语句书写有误</p> <p>遇到这种错误时，回过头来，检查 sql 语句。</p>			
教学小结	<ol style="list-style-type: none"> 1. 如何建立数据库的连接 2. 如何断开与数据的连接 3. 数据库操作五步走 <ol style="list-style-type: none"> (1) 建立连接 (2) 给出 sql 语句 (3) 建立数据库连接桥上的小车 (4) 执行 sql 语句 (5) 获得执行结果。 	总结	听讲记录	5
布置作业	<p>课后任务</p> <p>同步课外项目：图书管理系统。</p> <p>本节复习：课堂录像、项目说明文档</p> <p>作业要求：程序上传至云盘，放在 *组-姓名 文件夹下。</p>			
课后学习资源	<p>课堂录像</p> <p>SQL 语句复习资料</p>			