**实验报告**

姓名： 林炬乙 学号： 3180103721

课程名称： Java应用技术 任课老师： 楼学庆

实验名称： 数据库操作 实验日期： 2020/12/19

# 1 实验目的和要求

（分点简要说明本次实验需要进行的工作和最终的目的）

Java的数据库操作有三重境界：

1、直接数据库操作，数据库连接、SQL语句等。

2、对应数据库中的表建一个类，以类的方式隔离数据库操作。

3、对于任意指定的表自动生成相应的类的Java程序。实现基本的数据库查询、显示与增、删、改等常规操作。

本作业理论上应该有四~五个Java源程序：

1)对数据库的直接操作；建立连接、SQL语句等。12.15

2)对指定表建一个类，以类的方式实现对数据库操作的隔离。12.16

3)对于任意指定表可自动产生相应的类的程序[程序2)]。12.18 -12.19

4)1到2个对于不同的表，由3)产生的自动类实例。为其它程序员提供数据库编程的工具的程序。

# 2 实验原理

### 1. 用java类表示一个表

 Apache Derby是一个完全用 java 编写的数据库，Derby是一个Open source的产品。 Apache Derby非常小巧，核心部分derby.jar只有2M，既可以做为单独的数据库服务器使用，也可以内嵌在应用程序中使用。

ORM：对象关系映射

表--->类

字段--->属性

行记录--->对象

一般建package的时候，**常用entity、pojo、vo、dto或者pojo来代表实体类文件夹**

### 2. 让一个类表示一个表

类是一个iterator, vector存储着所有数据.

方法executeQuery  
       用于产生单个结果集的语句，例如 SELECT 语句。 被使用最多的执行 SQL 语句的方法是 executeQuery。这个方法被用来执行 SELECT 语句，它几乎是使用最多的 SQL 语句。

 方法executeUpdate  
       用于执行 INSERT、UPDATE 或 DELETE 语句以及 SQL DDL（数据定义语言）语句，例如 CREATE TABLE 和 DROP TABLE。INSERT、UPDATE 或 DELETE 语句的效果是修改表中零行或多行中的一列或多列。executeUpdate 的返回值是一个整数，指示受影响的行数（即更新计数）。对于 CREATE TABLE 或 DROP TABLE 等不操作行的语句，executeUpdate 的返回值总为零。

      使用executeUpdate方法是因为在 createTableCoffees 中的 SQL 语句是 DDL （数据定义语言）语句。创建表，改变表，删除表都是 DDL 语句的例子，要用 executeUpdate 方法来执行。你也可以从它的名字里看出，方法 executeUpdate 也被用于执行更新表 SQL 语句。实际上，相对于创建表来说，executeUpdate 用于更新表的时间更多，因为表只需要创建一次，但经常被更新。

PreparedStatement 可以创建参数化的SQL语句.

如果使用PreparedStatement，可以解决两个问题

\* 1、SQL注入，使用PreparedStatement时，是可以在SQL语句中，对值所在的位置使用?这种占位符的,使用占位符之后，实际的值，可以通过另外一份放在数组中的参数来代表。此时PreparedStatement会对值做特殊的处理，往往特殊处理后，就会导致不法分子的恶意注入的SQL代码失效

\* 2、提升性能，使用PreparedStatement之后，结构类似的SQL语句，都变成一样，因为值的地方都会变成?，那么一条SQL语句，在MySQL中只会编译一次，后面的SQL语句过来，就直接拿编译后的执行计划加上不同的参数直接执行，可以大大提升性能

CallableStatement 可以执行sql 存储过程.

DatabaseMetaData可以获取元数据. 如url, 用户名, 驱动程序名字

ResulteMetaData可以获取元数据. 如表的列数和行数

### 3.对于任意指定表可自动产生相应的类的程序[程序2)]。

数据库是固定的, 就先建立链接,sql 选择不同的表, 然后提取表的metadata元数据即可建立对应的类.

可以搜索关键词: ResultSetMetaData document 来查询返回值. 如下面这个网址<http://www.openoffice.org/api/docs/common/ref/com/sun/star/sdbc/DataType.html>

code += "private void Init() {\n"; 看起来非常冗长,但是直接 输入code += " ";, 然后复制自己写的类的代码, eclipse可以直接加上\n换行符,比较方便.

可滚动和可更新的结果集.Sql的update语句比建立一个查询, 然后一边遍历一边修改数据高效的多, 不过对于用户能够任意修改数据的交互式程序来说,使用可更新的结果集是非常有意义的.

难点: String sql 怎么提取出每一个列的名字?可以用metaname.

# 3 实验内容

首先完成对数据库的链接,

12月14日 完成对数据库的直接操作；建立连接、SQL语句等。

12月15日, 创建新project, 如上图完成相对路径, 单独文件夹打包即可, 不再依赖绝对路径.

让一个类 表示一个表, 本次实验用people类表示 people表.

### 新建实体类对应数据库指定表来操作

1. 新建package dto, class People

2. 新建表people

3. vector 存储people类, 每个类存储一行的数据.

4. 在12.16日完成通过类的方法进行插入操作

### 自动编译类

完成对应表产生java文件并自动编译成类. 产生2个自动类实例.

对于任意指定的表自动生成相应的类的Java程序。实现基本的数据库查询、显示与增add()、删delete()、改setxx()等常规操作。

自动生成类,可以删除, 插入,显示. 可以自动产生较复杂的sql语句.

完成多输入的类生成.

### 遇到的错误:

问题0. 输入文件的位置作为字符串的时候,遇到Invalid escape sequence (valid ones are  \b  \t  \n  \f  \r  \"  \'  \\ ) 报错, 一个正则表达式错误

原因：Java源代码的字符串中的反斜线被解释为Unicode 转义或其他字符转义

解决办法： 在“\”后加（\b   \t   \n   \f   \r   \） 其中的一个就可以了。当然也可以直接把语句中的“\”去掉。

问题1 Unhandled exception type IOException

这是ioe异常,在read.line方法前加try,catch语句。或者在方法上抛出异常类

问题2忽然br.close();出错了,不知道为何.java.lang.NullPointerException

解决方法:把close()放在try语句里面,

问题3 解决了之后, java.lang.NullPointerException还是存在

应该是因为每个try语句里不通?

BufferedReader br = null;放在try里面

问题4

Unreachable catch block for IOException. This exception is never thrown from the try statement body不可到达的 catch 块异常。try 中的语句，永远不会引发此异常

问题5

解决方法:

问题6 Exception processing async thread queue

解决方法: 只要关掉'expressions'视图就可以了

问题7: org.apache.derby.jdbc.EmbeddedDriver 找不到

Classpath 和modulepath有什么区别?

原因: 是由于我们的项目没有加入此jar包导致的，我们右键项目点击属性

首先，导入jar包时，必须导入到 classpath下，若导入到moduelpath下，在进行 import 导入接口时，会出错（找不到该接口）

classpath与moduelepath的区别：  
ModulePath 的概念和ClassPath 类似，不过 ModulePath 中的 Jar 包或 Jmod 文件被当作 Module 来处理，而 ClassPath 中的的 Jar 包，无论是否模块化都会被当作传统 Jar 包处理。

问题8:

解决方法:

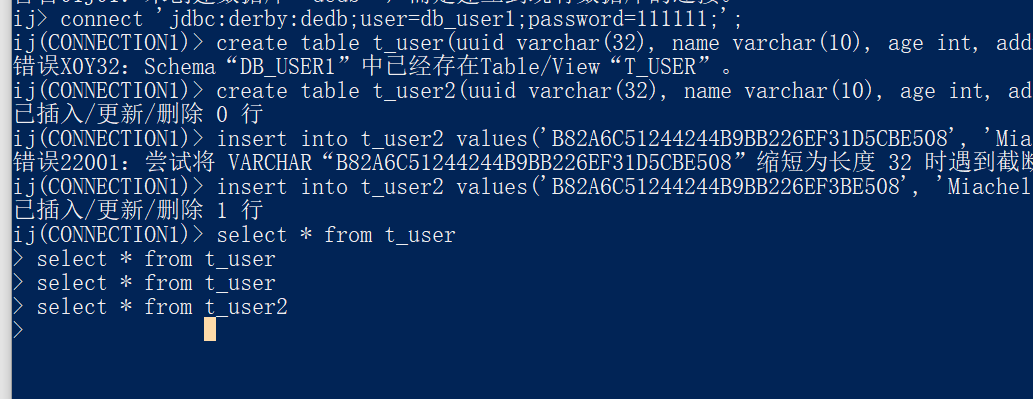
问题9: errors exist in required project

解决方法: 因为可能是同一个工程中的其他代码有错误，而执行的代码与这个有错误的代码没有关系，因为在同一个project，Eclipse依然会报错，但没有影响到执行的那段代码

问题10: java引用同一个包下面的类

如果是静态方法，直接 类名.方法名即可，如果是非静态方法，则需new一个对象出来，然后用对象.方法名调用

问题11: java.sql.SQLSyntaxErrorException: 表/视图 'T\_USER' 不存在。



命令行中要有分号, 不然无限换行.

要用对应的用户名和密码登录, 不然就会显示不存在. 用户(username)登录的和建表的用户不是一个用户。故用户保持一致即可。

但是一样了还是有'FT\_USER' 不存在。 而且T\_USER可以查看, 不知道为啥

真的很离谱, 差别就是ft\_user有主键, 另一个没有. 我又试了试create table usertable(id int primary key, name varchar(20)); 也读不出来.

试试这个create table tb\_test(id int generated by default as identity,title varchar(10));会自增但是不是主键.也不行!! 这样也不行? 我新建的啥都不行, 只有之前的t\_user可以读出,为啥?我搜索不到people表,但是people表也可以读出。

原来如此! 原来是因为我在文件夹内文件夹也有一个数据库,坑人!有点功能就没法读出了吗. 无语子.原来是因为文件夹不对.下次要控制变量.

问题12: java.sql.SQLDataException: 类型 int 的字符串格式无效。

java.sql.SQLException: 未找到列 '0'。Caused by: ERROR S0022: 未找到列 '0'。

解决方法: 要用字符串 , id = rs.getInt("id"); 而不是id = rs.getInt(id);

问题13: name没有正确导入类中.

输出为:

12

we

12

wangqiang

People{id=2, name='null', age=12, address='null'}

finished

怎么让一个类 表示一个表?

解决方法: 整个表就放在类中的vector里就可以了 ,

问题14 : java.sql.SQLDataException: 类型 INTEGER 的字符串格式无效。preSt.setString(2, name); 会报错

参数位置“3”超出范围。此预定义语句的参数数量为“2”。

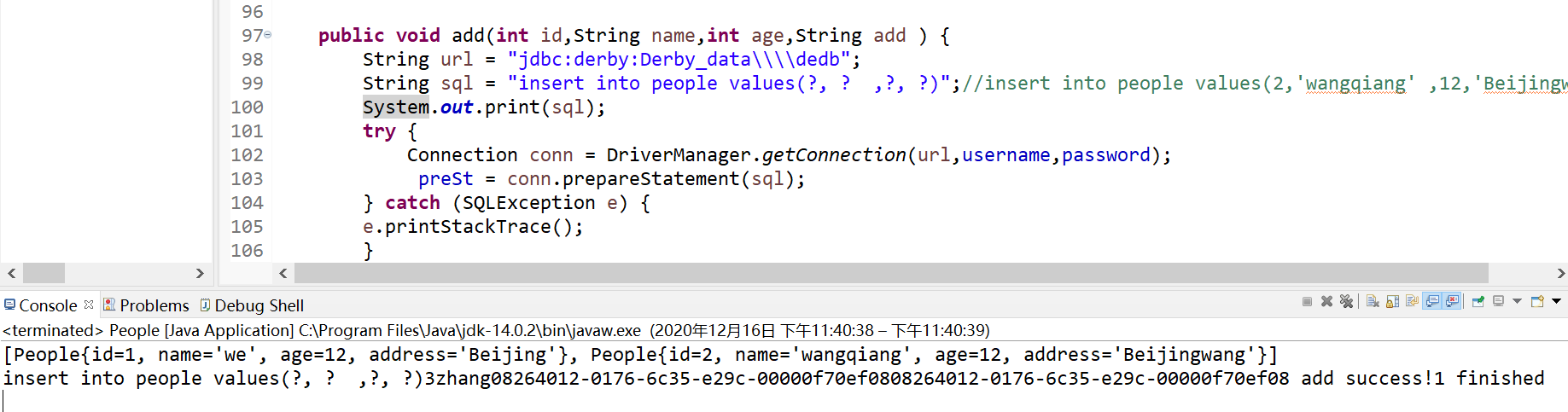
String sql = "insert into people values(?,' ? ' ,?,' ? ')";//insert into people values(2,'wangqiang' ,12,'Beijingwang');

System.out.print(sql);

try { Connection conn = DriverManager.getConnection(url,username,password);

preSt = conn.prepareStatement(sql);

这样预定义好像不对.可以通过看异常stack来分析,Java有很完整的异常栈, 分析起来还是比较方便的.



如上图,String sql = "insert into people values(?, ? ,?, ?)";

这样是对的 ,在java程序中不需要单引号.

问题15 : 错误：java.sql.SQLSyntaxErrorException: 列“eeeq”不在 FROM 列表的任何表中，或者它出现在 join 规范内但超出 join 规范的作用域，或者它出现在 HAVING 子句中但不在 GROUP BY 列表中。如果这是 CREATE 或 ALTER TABLE 语句，则“eeeq”不是目标表中的列。我用单引号,语法错误：Encountered "\'ID\', 部分插入的到底该怎么写?

应该这么写

String sql = "insert into people (\"ID\" ) values (" + String.*valueOf*(id) + ") " ; //insert into people values(2,'wangqiang' ,12,'Beijingwang');

双引号内加个空格甚至都不行“AGE ”也会出错.

# 4 实验结果和分析

（使用图片和文字叙述实验结果，并对这些结果进行适当分析）

12月14日

完成对数据库的直接操作；建立连接、SQL语句等。

连接数据库 connect 'jdbc:derby:dedb;user=db\_user1;password=111111;create=true';

create table t\_user(uuid varchar(32), name varchar(10), age int, address varchar(40));

insert into t\_user values('B82A6C5244244B9BB226EF31D5CBE508', 'Miachel', 20, 'street 1');

insert into t\_user values('B82A6C5244244B9BB226EF31D5CBE509', 'Andrew', 35, 'street 1');

insert into t\_user values('B82A6C5244244B9BB226EF31D5CBE510', 'Orson', 47, 'street 1');

insert into t\_user values('B82A6C5244244B9BB226EF31D5CBE511', 'Rambo', 19, 'street 1');

连接数据库 connect 'jdbc:derby:dedb;user=db\_user1;password=111111;create=true';

create table ft\_user(ftid int NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1), name varchar(20), password varchar(40));// 建表设置id为主键并自增, **不能手动指定主键标识，必须由数据库控制。**

create table ft\_user (name generated **by default** as identity, password varchar(40));标识列的有效类型只能是 BIGINT、INT 和 SMALLINT。

insert into ft\_user(name,password) values( 'user1', '123');

insert into ft\_user(name,password) values( 'user2', '123');

insert into ft\_user(name,password) values( 'user3', '123');

insert into ft\_user(name,password) values( 'user4', '123');

select \* from ft\_user ;可以

select 1 from ft\_user where id = 1 limit 1; 寻找

create table ft\_user(id int not null, name varchar(20), password varchar(40));(我就普通地输入账号,密码,昵称)

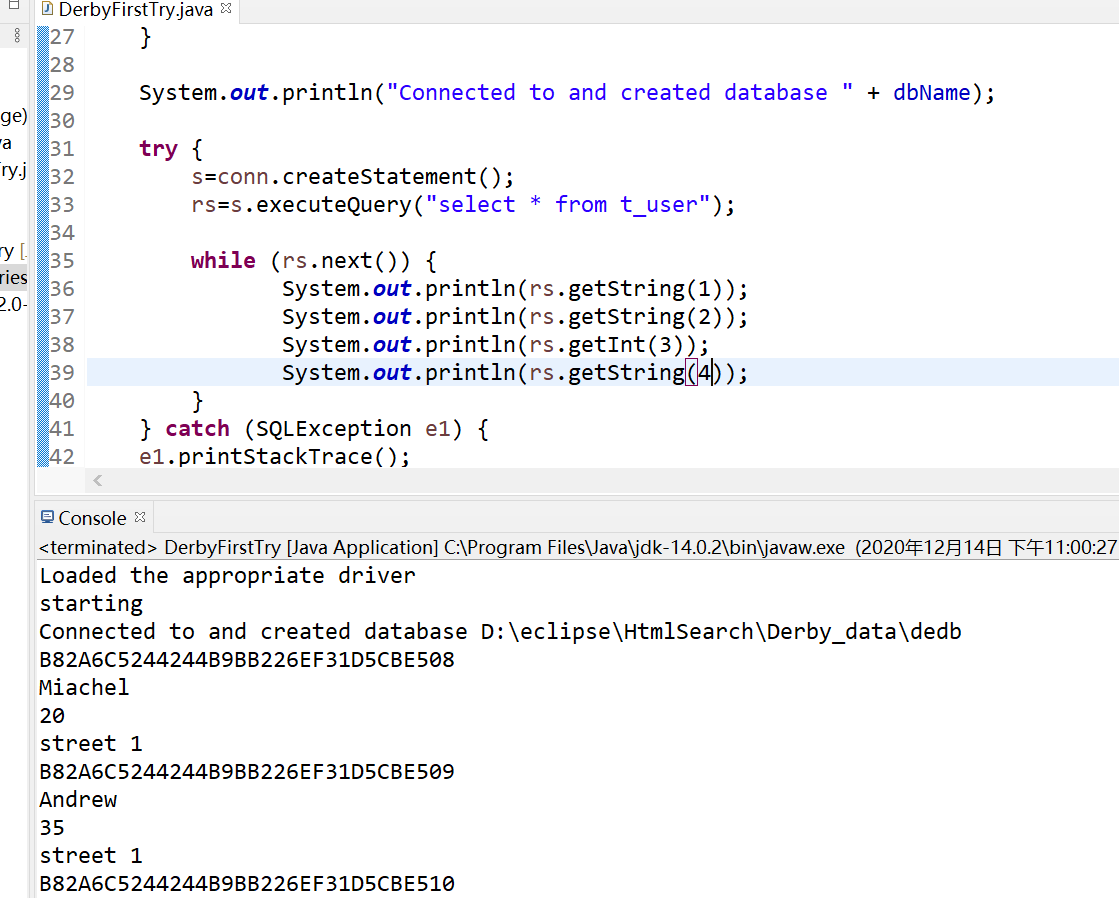
insert into ft\_user values(123, 'user4', '123');

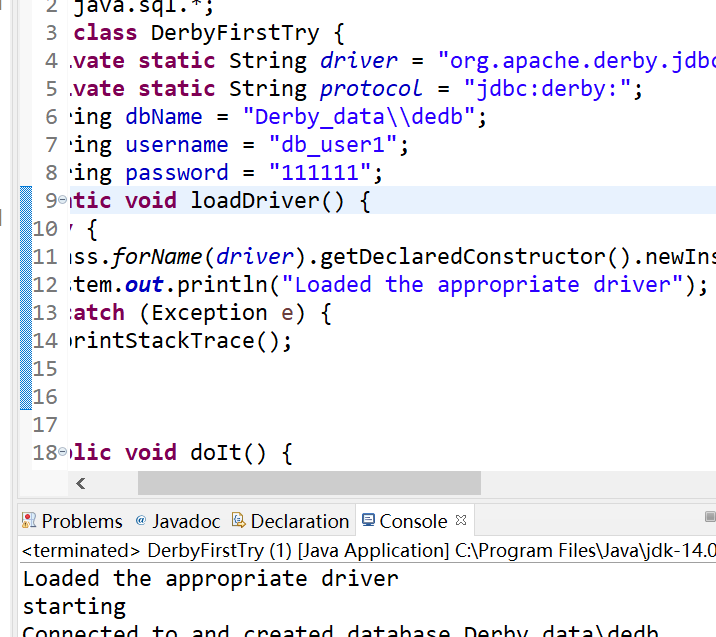
create table ft\_user(id int, name varchar(20), password varchar(40));//

create table tty2(uuid varchar(32), name varchar(10), age int, address varchar(40));

select 1 from user where id = xxx limit 1；会user表中遍历到第一个满足条件（id=xxx）时停止，返回结果为1，若未查到相应的结果，返回null

drop table ft\_user;删除表





12月15日, 创建新project, 如上图完成相对路径, 单独文件夹打包即可, 不再依赖绝对路径.

## 封装类

Ij中输入以下语句

create table people(id int, name varchar(20), age int, address varchar(200));

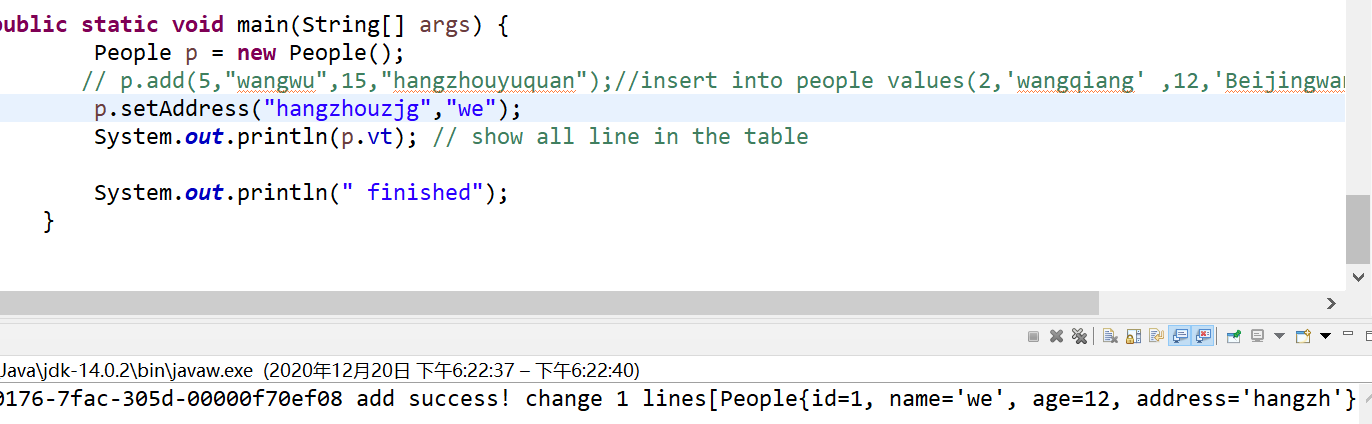
insert into people values(1,’we’ ,12,’Beijing’);// use ‘ instead of “”

insert into people values(2,'wangqiang' ,12,'Beijingwang');

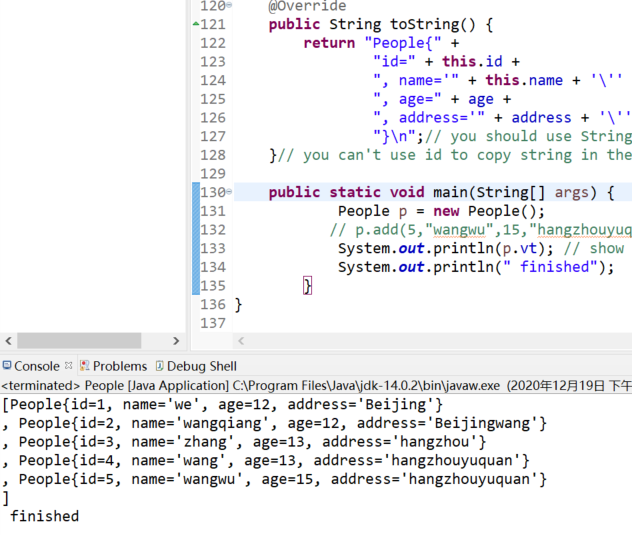
select age from people

update people set address = ? where name = ?;

update people set address = 'wenjiang' where name = 'we';



实现了一个类表示表的一行. Vector 输出这一行.



Vector 存储所有行的数据,

Vector<People> vt = new Vector<People>();

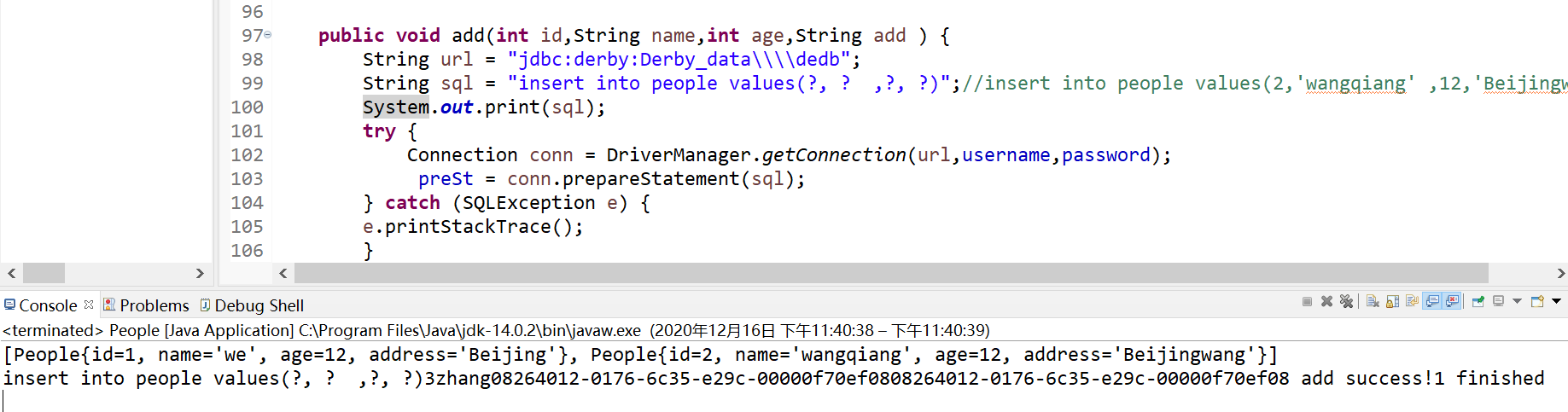
本身p这个对象,只有vt有效, 没有name这些属性.

toString() 实现基本的数据库查询、显示

set Address 实现修改address .

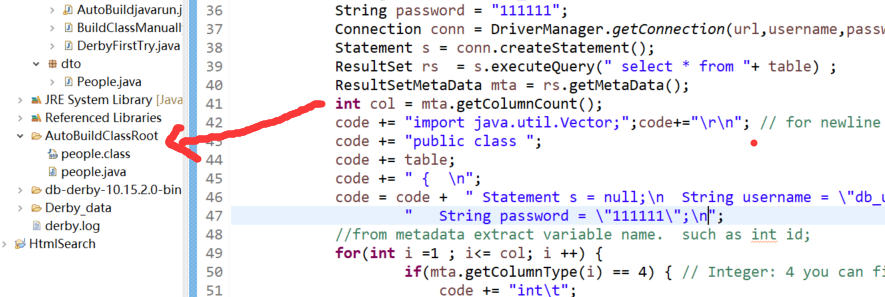
12.16日,

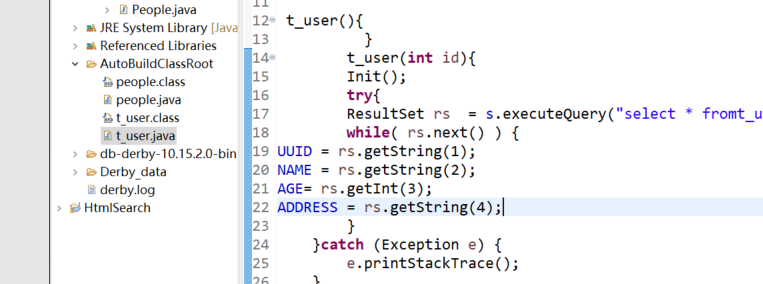
完成通过类的方法进行插入操作.



## 自动编译成类

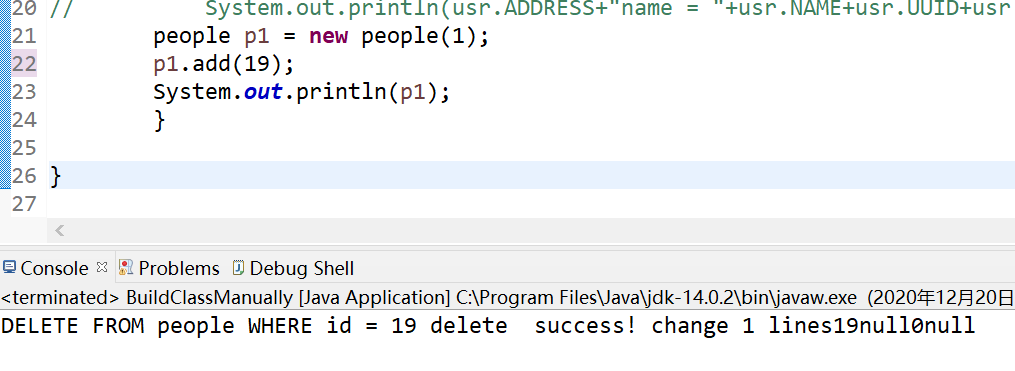
完成对应表产生java文件并自动编译成类. 产生2个自动类实例.





对于任意指定的表自动生成相应的类的Java程序。实现基本的数据库查询、显示与增add()、删delete()、改setxx()等常规操作。

用powershell命令行ij输入insert into people ("ID") values( 5 );

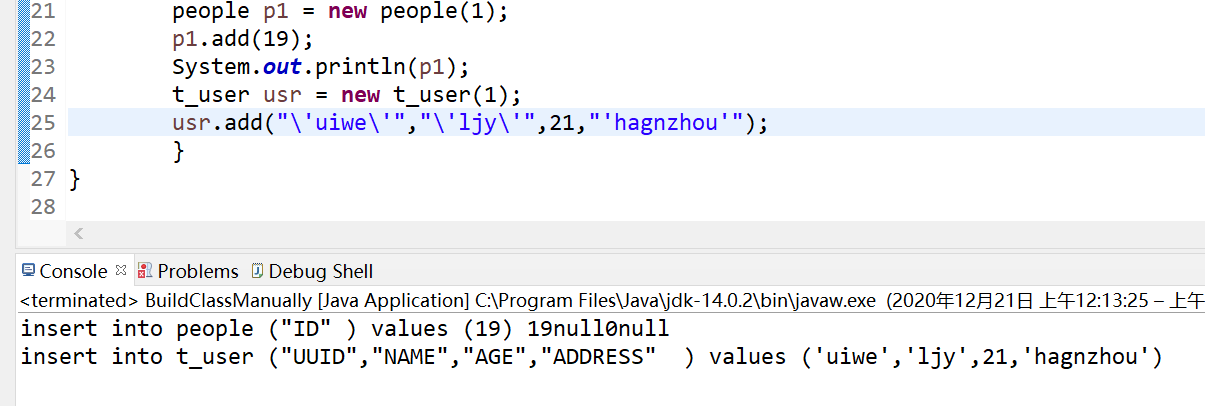


自动生成类,可以删除, 插入,显示.

Sql语句带入变量后为 insert into people ("ID" ) values (19);

可以自动产生较复杂的sql语句.

String sql = "insert into t\_user (\"UUID\",\"NAME\",\"AGE\",\"ADDRESS \" ) values ("+UUID+NAME+String.valueOf(AGE)+ADDRESS+ ") " ;



完成多输入的类生成.

注意,这个类的使用有固定的格式

p1.add(19,"weiu",20,"shier"); 会报错列“WEIU”不在 FROM 列表的任何表中 需要usr.add("\'uiwe\'","\'ljy\'",21,"'hagnzhou'"); 嵌套单引号