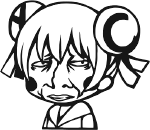
Java就业班-第70天  链接：<https://www.acfun.cn/a/ac10248250>

IT培训机构实地观察日记 Java就业班-第71天

**前言声明：**

**此系列文章绝非网络推广文、软文、黑或洗白文等或出于任何商业目的而投稿的文章，仅出于UP主为记录并分享自身真实经历的目的而发表，不为文中出现的任何事物作出保证，全文均属于完全个人主观内容，无任何刻意诋毁或吹捧意图。文本、照片中会尽量避免出现敏感信息，如有不合适内容举报还请及时联系我立即删除。**

       如果有同一个地点，同一个时间，甚至与我在同一个班的兄弟、朋友、Acer们，还请继续当作没有看见，请尽量配合避免暴露不必要的真实信息，以免造成一些不必要的麻烦，十分感谢。

或者换句话说，有什么问题请直接冲我来，我们厕所解决，谢谢！

更新规律：

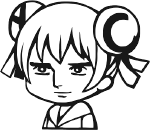
       基本上有课程的当天晚上(18：00---24：00)之间会进行更新投稿。休息日一般为星期天与节假日，无特殊理由将不会更新。如有任何建议、问题还请大家随意在评论区里留言，或者直接私信我，我看到后会尽快给予回复，感谢大家支持！

==========================================================

**友情提示：**

**本日记中包含大量低质量代码，**

**对代码内容不感兴趣的朋友可以放心跳过至最后部分，丝毫不影响阅读。**

**(不过那也就没什么内容了。但是可以省时间。)**

---------------------------------------------------

计划课程安排：

FUNDAMENTAL(7天)         该阶段已结束

OOP(11天)                 该阶段已结束

JAVASE01(10天)        该阶段已结束

JAVASE02(12天)        该阶段已结束

WEBBASIC(11天)       该阶段已结束

JQUERY(2天)            该阶段已结束

DATABASE(5天)      该阶段已结束

JDBC(2天)             ←当前为该阶段第2天

SERVLETJSP(12天)

SPRINGMYBATIS01(8天)

AJAX(2天)

SPRINGMYBATIS02(12天)

LINUX(2天)

NGINX(2天)

REDIS(2天)

T-STORE(5天)

预计合计：105天

预计剩余：47天

注：上述信息随时可能发生更改，仅供参考。

---------------------------------------------------

**2019年5月24日**

**Java就业班-第71天**

**8:30实训室2-早自习**

直播用电脑又一次故障，所以今天早自习无法晨讲。

昨天错过了晚自习班主任的毕业小动员，但今天早上我把收集到的大致情况整理了一下。

大致计划如下：

1.  以后每周一晚自习都要抽查上周知识点

2.  月底前会对重点学员进行访谈

3.  本月底会由项目经理单独出题进行考试

4.  每周至少刷一套笔试题并考试

5.  下月开始进行简历的制作课程

6.  下月开始对每名学员进行就业意愿的一对一摸底

7.  模拟面试的前一周进行简历定稿和安排复习计划

8.  下月底进行模拟面试和就业动员会以及毕业考试

9.  推荐就业时间待定。

**9:25实训室2-正课**

开始上课。

——数据库连接池DBCP

**9:45实训室2-正课**

好吧，这回是主播端出问题了，一节课就这么耗过去了。

**10:17实训室2-正课**

——Database Connectivity Pool

新建一个Maven项目JDBC02，Demo01代码：

package jdbc;

import java.io.InputStream;

import java.sql.Connection;

import java.util.Properties;

import org.apache.commons.dbcp.BasicDataSource;

public class Demo01 {

    public static void main(String[] args) throws Exception {

       //创建数据源对象

       BasicDataSource ds = new BasicDataSource();

       //读取配置文件

       Properties p = new Properties();

       InputStream ips = Demo01.class.getClassLoader().getResourceAsStream("jdbc.properties");

       p.load(ips);

       String driver = p.getProperty("driver");

       String url = p.getProperty("url");

       String username = p.getProperty("username");

       String password = p.getProperty("password");

       //设置连接信息

       ds.setDriverClassName(driver);

       ds.setUrl(url);

       ds.setUsername(username);

       ds.setPassword(password);

       ds.setInitialSize(3);//设置初始链接数量

       ds.setMaxActive(5);//设置最大连接数量

       //获取连接对象 注意导错包 异常抛出

       Connection conn = ds.getConnection();

       System.out.println(conn);

    }

}

再将Demo01代码封装到新建的DBUtils工具类中。

package jdbc;

import java.io.IOException;

import java.io.InputStream;

import java.sql.Connection;

import java.util.Properties;

import org.apache.commons.dbcp.BasicDataSource;

/\*\*

 \* JDBC工具类

 \*

 \* @author Administrator

 \*

 \*/

public class DBUtils {

    /\*

     \* getConn()连接数据库，返回Connection对象conn;

     \*/

    private static BasicDataSource ds ;

    static {

       // 创建数据源对象

       ds = new BasicDataSource();

       // 读取配置文件

       Properties p = new Properties();

       InputStream ips = Demo01.class.getClassLoader().getResourceAsStream("jdbc.properties");

       try {

           p.load(ips);

           String driver = p.getProperty("driver");

           String url = p.getProperty("url");

           String username = p.getProperty("username");

           String password = p.getProperty("password");

           // 设置连接信息

           ds.setDriverClassName(driver);

           ds.setUrl(url);

           ds.setUsername(username);

           ds.setPassword(password);

           ds.setInitialSize(3);// 设置初始链接数量

           ds.setMaxActive(5);// 设置最大连接数量

       } catch (IOException e) {

           e.printStackTrace();

       }

    }

    public static Connection getConn() throws Exception {

       // 获取连接对象 注意导错包 异常抛出

       Connection conn = ds.getConnection();

       System.out.println(conn);

       return conn;

    }

}

**10:29实训室2-正课**

——预编译插入数据方法PreparedStatement

-好处：

1.可以避免字符串的拼接，是代码结构更为简洁，不易出错

2.可以避免SQL注入风险（因为在编译时已经把SQL逻辑锁死，不会被用户输入的内容改变逻辑）

-如果SQL语句中有变量则使用PreparedStatement如果没有变量则使用Statement。

代码：

package jdbc;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.Statement;

import java.util.Scanner;

/\*\*

 \* 向数据库插入信息。

 \* @author Administrator

 \*

 \*/

public class Demo02 {

    public static void main(String[] args) {

       Scanner sc = new Scanner(System.in);

       System.out.println("请输入用户名");

       String name = sc.nextLine();

       System.out.println("请输入密码");

       String password = sc.nextLine();

       //create table jdbcuser(id int primary key auto\_increment,username varchar(10),password varchar(10));

       try(Connection conn = DBUtils.getConn()) {

           //原始方法

//         Statement stat = conn.createStatement();

//         String sql = "insert into jdbcuser values (null,'" + name + "','" + password + "')";

//         stat.executeUpdate(sql);

           //预编译方法

           String sql = "insert into jdbcuser values (null,?,?)";

           PreparedStatement ps = conn.prepareStatement(sql);

           //替换掉问号

           ps.setString(1, name);

           ps.setString(2, password);

           //执行SQL

           ps.executeUpdate();

           System.out.println("注册成功");

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

}

**11:41实训室2-正课**

——什么是SQL注入

代码：

package jdbc;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.Scanner;

public class Demo03 {

    public static void main(String[] args) {

       Scanner sc = new Scanner(System.in);

       System.out.println("请输入用户名");

       String name = sc.nextLine();

       System.out.println("请输入密码");

       String password = sc.nextLine();

       //验证登录

       try(Connection conn = DBUtils.getConn()) {

           //不推荐方法，理由：有SQL注入漏洞

//         Statement stat = conn.createStatement();

//         String sql = "select count(\*) from jdbcuser where username='" +

//         name + "' and password='" + password + "' or '1'='1'";

//         System.out.println(sql);

//         ResultSet rs = stat.executeQuery(sql);

           //推荐方法，理由：可防止SQL注入

           String  sql = "select count(\*) from jdbcuser where username=? and password=?";

           PreparedStatement ps = conn.prepareStatement(sql);

           ps.setString(1, name);

           ps.setString(2, password);

           ResultSet rs = ps.executeQuery();

           while(rs.next()) {

              int count = rs.getInt(1);//获取数据

              if(count>0) {

                  System.out.println("登陆成功");

              }else {

                  System.out.println("登录失败");

              }

           }

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

}

**11:47实训室2-正课**

——批量操作

代码：

package jdbc;

import java.sql.Connection;

import java.sql.Statement;

import org.junit.Test;

public class Demo04 {

    @Test

    public void test01(){

       try(Connection conn = DBUtils.getConn()) {

           Statement stat = conn.createStatement();

           String sql1 = "insert into jdbcuser(username) values('AAA')";

           String sql2 = "insert into jdbcuser(username) values('BBB')";

           String sql3 = "insert into jdbcuser(username) values('CCC')";

           stat.addBatch(sql1);

           stat.addBatch(sql2);

           stat.addBatch(sql3);

           stat.executeBatch();//执行批量操作

           System.out.println("执行完成！");

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

}

**14:05实训室2-正课**

代码2：

@Test

    public void test02() {

       try (Connection conn = DBUtils.getConn();){

           String sql = "insert into jdbcuser(username) values(?)";

           PreparedStatement ps = conn.prepareStatement(sql);

           for(int i = 1 ; i <=100 ; i++) {

              ps.setString(1, "name" + i);

              ps.addBatch();

              //避免内存溢出

              if(i % 20==0) {

                  ps.executeBatch();

              }

           }

           ps.executeBatch();//批量执行

           System.out.println("执行完成！");

           System.out.println("方法2执行完毕");

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

**14:30实训室2-正课**

练习：实现简单的数据查询功能

代码：

package jdbc;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Scanner;

public class Demo05 {

    public static void main(String[] args) {

       Scanner sc = new Scanner(System.in);

       System.out.println("输入查询页数：");

       int page = sc.nextInt();

       System.out.println("输入查询条数：");

       int count = sc.nextInt();

       try(Connection conn = DBUtils.getConn()) {

           String sql = "select \* from jdbcuser limit ?,?";

           PreparedStatement ps = conn.prepareStatement(sql);

           //替换问号

           ps.setInt(1, (page-1)\*count);

           ps.setInt(2, count);

           ResultSet rs = ps.executeQuery();

           while(rs.next()) {

              String username = rs.getString("username");

              System.out.println(username);

           }

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

}

**14:53实训室2-正课**

练习：实现模拟转账功能

代码：

//创建数据表

create table jdbcaccount(id int,name varchar(10),money int);

insert into jdbcaccount values(1,'Kamui',5000),(2,'Subaru',10);

package jdbc;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

public class Demo06 {

    public static void main(String[] args) {

       //事务相关

       try(Connection conn = DBUtils.getConn()) {

           //关掉自动提交 开启事务

           conn.setAutoCommit(false);

           Statement stat = conn.createStatement();

           String sql1 = "update jdbcaccount set money=money-2000 where id=1";

           stat.execute(sql1);

           String sql2 = "update jdbcaccount set money=money+2000 where id=2";

           stat.execute(sql2);

           //查询Kamui余额是否大于0

           String sql3 = "select money from jdbcaccount where id=1";

           ResultSet rs = stat.executeQuery(sql3);

           while(rs.next()) {

              int money = rs.getInt("money");

              if(money>0) {//钱够

                  System.out.println("转账成功");

                  conn.commit();//提交操作

              }else {//钱不够

                  System.out.println("转账失败");

                  conn.rollback();//回滚操作

              }

           }

       } catch (Exception e) {

       }

    }

}

**15:15实训室2-正课**

——获取自增主键的值

代码：

package jdbc;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

public class Demo07 {

    public static void main(String[] args) {

       try(Connection conn = DBUtils.getConn()) {

           Statement stat = conn.createStatement();

           String sql =  "insert into jdbcuser values(null,'Kamui','aaabbb')";

           stat.executeUpdate(sql,Statement.RETURN\_GENERATED\_KEYS);

           //获取自增主键值

           ResultSet rs = stat.getGeneratedKeys();

           while(rs.next()) {

              int id = rs.getInt(1);

              System.out.println(id);

           }

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

}

**15:40实训室2-正课**

练习：实现保存队员功能

代码：

//创建数据表

Create table team(id int primary  key auto\_increment,name varchar(10));

Create table player(id int primary key auto\_increment,name varchar(10),team\_id int);

package jdbc;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.Scanner;

public class Demo08 {

    public static void main(String[] args) {

       Scanner s = new Scanner(System.in);

       System.out.println("请输入战队名称");

       String teamName = s.nextLine();

       System.out.println("请输入队员名称");

       String playerName = s.nextLine();

       try (Connection conn = DBUtils.getConn()){

           String  sql1 = "insert into team values(null,?)";

           PreparedStatement ps1 = conn.prepareStatement(sql1,Statement.RETURN\_GENERATED\_KEYS);

           ps1.setString(1, teamName);

           ps1.executeUpdate();

           //获取自增主键值

           ResultSet rs = ps1.getGeneratedKeys();

           while(rs.next()) {

              int  teamId = rs.getInt(1);

              //保存队员

              String sql2 = "insert into player values(null,?,?)";

              PreparedStatement ps2 = conn.prepareStatement(sql2);

              ps2.setString(1, playerName);

              ps2.setInt(2, teamId);

              ps2.executeUpdate();

           }

           System.out.println("数据插入成功！");

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

}

**16:20实训室2-正课**

——获取数据库元数据

代码：

package jdbc;

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.Statement;

public class Demo09 {

    public static void main(String[] args) {

       try (Connection conn = DBUtils.getConn()) {

           // 数据库元数据

           DatabaseMetaData dbmd = conn.getMetaData();

           System.out.println("数据库名称：" + dbmd.getDatabaseProductName());

           System.out.println("驱动版本：" + dbmd.getDriverVersion());

           System.out.println("连接地址：" + dbmd.getURL());

           System.out.println("用户名：" + dbmd.getUserName());

           String sql = "select \* from emp";

           Statement stat = conn.createStatement();

           ResultSet rs = stat.executeQuery(sql);

           // 获取表的元数据

           ResultSetMetaData rsmd = rs.getMetaData();

           // 得到表的字段数量

           int count = rsmd.getColumnCount();

           for (int i = 0; i < count; i++) {

              String name = rsmd.getColumnName(i + 1);

              String type = rsmd.getColumnTypeName(i + 1);

              System.out.println(name + ":" + type);

           }

       } catch (Exception e) {

           e.printStackTrace();

       }

    }

}

**16:33实训室2-正课**

至此为止，这个沙雕讲师的任务完成。明天他就会出现在我们教室，听到这个消息的学员们都已经开始磨刀霍霍，迫不及待要准备开喷了！

离下课还有1个多小时，讲师没什么好讲的了，于是开始聊（吹）他自己了。

简单的记录一下他的语录：

1.  我在北京毕业的第一份工资就是6500,一年后工资涨到1万2，二年后就已经2万2了。

2.  我后来创业失败，自行开过一个ios的培训班。

3.  我曾经干过很长一段时间的997工作。

4.  在我开始工作不到1年的时候就开始兼职当老师了。

5.  我在学生时代自己写过一个分享文件的网站，功能不是很轻大，但是资源分享很方便（适合开车），由此给学校的老师留下了很深的印象，以至于带领我也走上老师的道路。

6.  我能给你们的建议不多，就一条，也不花里胡哨，很实在：向钱（技术）看齐。

7.  技术=金钱。当你觉得你学不到什么东西的时候，你就该换工作了。

8.  靠自学学成的人我很佩服，但也没见过几个。最有效的自学时伴随着压力而成的。

9.  不用担心你会的不多，因为工作上你不会的必须学。

10.这一个月的教学总体而言我很开心。

以上。

懒得吐槽了，反正明天就能看到这个讲师本尊了。

**16:52实训室2-下课时间**

提前下课了。

**18:27宿舍-休息**

我打算租一台服务器，和班上的几个学员组队写点小项目玩玩，所以今天应该是我最后一次翘晚自习了吧。

由于我的种种翘课行为，以及今天在上课的时候摸鱼被班主任发现并质问，我觉得这个人已经开始看我不顺眼了…

今天过得很平淡，任何有趣的事情都没有发生。明天是和讲师的见面会，肯定是没法搞学习的了，估计下午就可以回去了。

从下周开始就是Severlet和JSP的课程了，还不知道讲师讲课水平如何，框架部分貌似是只能自学了。祝我好运。

今日感想总结：老实说，我开始有点厌倦这种生活了。幸好还剩下1个多月的时间，不然我感觉我可能会把这个爱好给毁掉。

今日评分：5/10分



今天就到这里，明天继续投稿。

感谢支持！

================================================================

**Acer程序员群：790482850**

**up主的java学习日记word文件及全部代码（随日记同步更新）：**

Adress：github.com/wt62635