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

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(3天) ←当前为该阶段第2天

DATABASE(6天)

JDBC(3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：108天

预计剩余：54天

注：上述信息随时可能发生更改，仅供参考。

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

**2019年5月15日**

**Java就业班-第63天**

**8:30实训室2-早自习**

今日晨讲主题——线程锁synchronized

本想抄一下这位老哥写的代码的，但是这位老哥似乎是早就已经写好了的，拿上来就开讲了，也就没能完整的抄下来。

早自习项目经理找上我，询问我为什么最近晚自习都没来上课。说明了自己睡眠不足与失眠的问题后，项目经理提醒到，差不多还有1个月的时间（6月中旬）就要推荐工作了，剩下的时间抓把劲努力学一下。我很诧异，于是反问道“就我们这种水平还能找到工作？”，项目经理的解释是说可以找到，单指我们现在所处的这个机构平均每个月有100多人等待就职，大部分都是被机构的合作企业直接过来带走，也有少部分人走的社招流程，所以总体而言工作机会是肯定有的。这周五开始学数据库，数据库学完后面基本上就没有新内容了，剩下的也就是一些项目案例练习。

项目经理说，之所以提醒你这个事，是因为1月班因为春节的缘故推迟了推荐就业所以会和我们一起抢夺岗位，也就是说我们这一批会有200多人争夺岗位，所以需要抓紧学一下了。

言下之意，似乎是在说工作肯定有，但是你们运气不好要和1月班竞争，所以课程不会上到最后，在下个月中旬就需要开始陆续就职了。

？？？？

**9:23实训室2-正课**

开始上课。因为直播故障，课程回顾被跳过了。

**10:14实训室2-正课**

参考昨天评论区里老哥们的前车之鉴，于是我决定今天的课不听了，自学。

复习一下之前自学过的内容：

什么是JDBC？

JDBC(Java Database Connection)为java开发者使用数据库提供了统一的编程接口，它由一组java类和接口组成。是java程序与数据库通信的标准API。JDBC API使得开发人员可以使用纯java的方式来连接数据库，并执行操作。

关于JDBC的操作步骤。

数据库安装并配置完毕后，可以按照以下步骤进行数据库的操作：

1. 加载数据库驱动程序
2. 连接数据库
3. 使用语句进行数据库操作
4. 关闭数据库连接。

※数据库的连接是非常有限的，所以打开以后一定要记得关闭。

JDBC常用接口。

·DriverManager接口

DriverManager是JDBC的管理层，作用于用户和驱动程序之间。

DriverManager跟踪可用的驱动程序，并在数据库和相应的驱动程序之间建立连接。

·Connection接口

Connection与特定数据的连接（会话），在连接上下文中执行SQL语句并返回结果。

DriverManager的getConnection()方法建立在JDBC URL中定义的数据库Connection连接上。

·Statement接口

用于执行静态SQL语句并返回它所生成结果的对象。

三种Statement类：

(父类)-Statement

(子类)--PreparedStatement

(子类)--CallableStatement

常用的Statement方法：

-execute()：运行语句，返回是否有结果集

-executeQuery()：运行slect语句，返回ResultSet结果集。

-executeUpdate()：运行insert/update/delete操作，返回更新的行数

**11:50实训室2-正课**

Statement测试代码：

package jdbc;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

/\*\*

\* 测试执行SQL语句，以及SQL注入问题

\* @author Administrator

\*

\*/

public class demo02 {

public static void main(String[] args) {

try {

//加载驱动类

Class.forName("com.mysql.jdbc.Driver");

Connection cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/kamuisdatabase","root","123456");

//注：一般实际开发中不会使用Statment

Statement stmt = cn.createStatement();

//设置变量

String name = "Subaru";

String password = "sub123";

int age = 17 ;

String sex = "男";

String birthday = "1994-12-12";

//编写数据库执行语句

//插入语句

// String sql = "INSERT INTO user(name,password,age,sex,birthday)"

// + " VALUES ('"+ name +"','"+ password +"',"+ age +

// ",'"+ sex +"','"+ birthday +"') ";

//测试SQL注入（危险：测试SQL漏洞的恶意语句）

String id = "5 or 1=1";

String sql = "DELETE from user where id=" + id ;

//执行数据库语句

stmt.execute(sql);

System.out.println("数据语句执行完毕！");

} catch (SQLException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

}

**12:05实训室2-午休时间**

难怪要学数据库的基本操作指令，实际上JDBC也就是通过java代码来向数据库发送指令执行。

到点了，吃饭去。

**14:10实训室2-正课**

开始上课了，还是在讲JQ。我还是继续学习JDBC吧。

测试PrepareStatement代码:

package jdbc;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

/\*\*

\* 测试PrepareStatement的基本用法

\* @author Administrator

\*

\*/

public class demo03 {

public static void main(String[] args) {

try {

//加载驱动类

Class.forName("com.mysql.jdbc.Driver");

Connection cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/kamuisdatabase","root","123456");

System.out.println("数据库已连接...");

String sql = "INSERT into user(name,password,age,birthday) values (?,?,?,?)";//?←占位符,可防止sql注入

PreparedStatement ps = cn.prepareStatement(sql);

//使用setObject可以省略掉指定数据类型的需要，通过参数可以自动识别对应数据类型。

ps.setObject(1, "Kamui");//参数索引是从1开始计算，并非从0开始计算

ps.setObject(2, "kam123");

ps.setObject(3, 25);

// ps.setDate(4, new Date(System.currentTimeMillis()));

ps.setObject(4, new Date(System.currentTimeMillis()));

System.out.println("开始执行插入指令...");

ps.execute();

System.out.println("指令执行完毕");

} catch (SQLException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

}

※在向插入数据时，如果某一数据类型没有默认值，则必须同时将其数据类型一同插入。例如：name无默认值，age默认值为0时，在插入age的时候，必须同时插入name的值，否则会报异常。

**14:50实训室2-课间休息**

不行了，我受不了了，快热到中暑了。我要出去走一走，吹吹风，室内已经快超过35度了。

**15:20实训室2-正课**

测试ResultStatement代码：

package jdbc;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

/\*\*

\* 测试ResultStatement的基本用法

\* @author Administrator

\*

\*/

public class demo04 {

public static void main(String[] args) {

Connection cn = null;

PreparedStatement ps = null ;

ResultSet rs = null ;

try {

//加载驱动类

Class.forName("com.mysql.jdbc.Driver");

cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/kamuisdatabase","root","123456");

System.out.println("数据库已连接...");

//查询

String sql = "select id,name,age from user where id>?";

ps = cn.prepareStatement(sql);

ps.setObject(1, 5);//把id大于5的数据取出

rs = ps.executeQuery();

while(rs.next()) {

System.out.println(rs.getInt(1) + "----" + rs.getString(2) + "----" + rs.getInt(3));

}

} catch (SQLException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}finally {//按照后开先关的顺序，关闭数据库

//注意一定要将异常捕获分开写，否则会导致抛出异常的时候其他链接无法没有执行关闭方法

if(rs!=null) {

try {

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(ps!=null) {

try {

ps.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(cn!=null) {

try {

cn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

}

**15:40实训室2-正课**

关于使用JDBC的批处理操作。

Batch

对于大量数据插入的批处理，建议使用Statement，因为PreparedStatement的预编译空间有限。当数量特别大时，会发生异常。

测试批量插入20000条数据到数据库：

package jdbc;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

/\*\*

\* 测试批处理的基本用法

\* @author Administrator

\*

\*/

public class demo05 {

public static void main(String[] args) {

Connection cn = null;

Statement sm = null ;

try {

//加载驱动类

Class.forName("com.mysql.jdbc.Driver");

cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/kamuisdatabase","root","123456");

System.out.println("数据库已连接...");

cn.setAutoCommit(false);//设为手动提交

long start = System.currentTimeMillis();

sm = cn.createStatement();

//生成2w条数据

for (int i = 0; i < 20000 ; i++) {

sm.addBatch("insert into user (name,password,age,sex,birthday) values ('Kamui"+ i +"','kam123',"+ i +",'男',now())");

}

//执行批处理

sm.executeBatch();

cn.commit();//提交事务

long end = System.currentTimeMillis();

System.out.println("插入20000条数据：耗时(ms)" + (end-start));

//输出结果：插入20000条数据：耗时(ms)1618

} catch (SQLException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

}finally {//按照后开先关的顺序，关闭数据库

//注意一定要将异常捕获分开写，否则会导致抛出异常的时候其他链接无法没有执行关闭方法

if(sm!=null) {

try {

sm.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(cn!=null) {

try {

cn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

}

**16:20实训室2-正课**

稍微摸了一会鱼，看了一下现在课上讲到哪里了。还在讲JQ。

……

好热啊……好不容易风干了的衣服，又汗湿了。

**17:21实训室2-正课**

已经开始写今天的课程笔记了，看样子是快下课了。

下课回去之后简单看一下吧。

**17:47实训室2-正课**

今日课程笔记：

###课程回顾

1. jQuery 是一个js语言所写的框架，可以让程序员写的更少 实现的更多

2. 页面加载完成事件 $(function(){})

3. js-jq var jq = $(js);

4. jq-js var js = jq[0];

5. 选择器

- 基础选择器 ： $("div") #id .class div,#id \*

- 层级选择器： div span div>span div+span div~span

.siblings("div") .prev() .prevAll() .next() .nextAll()

- 过滤选择器： div:first/last/even/odd/eq()/lt()/gt()/not()

- 内容选择器： div:has(p)/empty/parent/contains('xxx')

- 可见选择器： div:hidden/visible .show() .hide() .toggle()

- 属性选择器： div[id!='xxx']

- 子元素选择器： div:first-child div:last-child div:nth-child(n) 1

- 表单选择器： :input :password :radio :checkbox :checked input:checked :selected

6. 创建元素 var d = $("<div></div>");

7. 添加元素 父元素.prepend() 父元素.append();

8. 插入元素 兄弟元素.before() 兄弟元素.after();

9. 删除元素 被删除的元素对象.remove();

10. 文本相关 .text() .text("xxxx");

11. html相关 .html() .html("<xxx>sdfsdf</xxx>");

12. 样式相关 .css("color") .css("color","red"); .css({"width":"100px","height":"200px"});

13. 属性相关 .attr("class") attr("class","abc");

14. 获取元素的子元素们 .children()

15. 获取元素的父元素 .parent()

###jQuery事件相关

1. 常见事件

- 鼠标事件： 点击事件 click 鼠标移入事件mouseover 鼠标移出mouseout 鼠标按下mousedown 鼠标抬起mouseup 鼠标移动mousemove

- 键盘事件： 键盘按下 keydown 键盘抬起 keyup

- 状态改变事件： 页面加载完成$(function(){}) 获取焦点focus 失去焦点blur 表单提交submit 值改变change 窗口尺寸改变resize

2. 事件模拟

元素对象.trigger("事件名");

3. 鼠标移入移出事件合并

元素对象.hover(鼠标移入执行方法,鼠标移出执行方法);

###动画相关

if(this.value=="隐藏"){

$("img").hide(3000,function(){

alert('动画做完了');

});

}else if(this.value=="显示"){

$("img").show(2000,function(){

alert('xxx');

});

}else if(this.value=="淡出"){

$("img").fadeOut(1000);

}else if(this.value=="淡入"){

$("img").fadeIn(2000);

}else if(this.value=="上滑"){

$("img").slideUp(2000);

}else if(this.value=="下滑"){

$("img").slideDown(2000);

}else{//自定义

$("img").animate({"left":"200px"},1000)

.animate({"top":"200px"},1000)

.animate({"left":"0px"},1000)

.animate({"top":"0px"},1000)

.animate({"width":"200px"},1000)

.animate({"width":"100px"},1000)

.fadeOut(3000,function(){

$("img").remove();

});

}

});

});

###jQuery课程回顾

1. 写的更少 实现的更多

2. js-jq var jq = $(js)

3. jq-js var js = jq[0]

4. 选择器

- 基础选择器 div #id .class div,#id \*

- 层级选择器 div span div>span div+span div~span

.siblings() .prev() .prevAll() .next() .nextAll()

- 过滤选择器 div:first/last/even/odd/eq()/lt()/gt()/not()

- 内容选择器 div:has(p)/empty/parent/contains('xxx')

- 可见选择器 div:hidden/visible .show() .hide() .toggle()

- 属性选择器 div[id!='xxx']

- 子元素选择器 div:first-child/last-child/nth-child(n)

- 表单选择器 :input :radio :password :checkbox :checked input:checked :selected

5. 创建元素 var d = $("<div></div>");

6. 添加 .prepend() .append()

7. 插入 兄弟.before() 兄弟.after()

8. 删除 元素.remove();

9. 文本相关 .text()

10. html相关 .html()

11. 样式 .css("color") .css("color","red"); .css({"":"","":""})

12. 属性 .attr("属性名","值")

13. 上级元素 parent（） 子元素们 .children()

14. 事件

- 去掉on

- 事件模拟 元素.trigger("事件名");

- 移入移出合并 元素.hover(function(){},function(){});

15. 动画

隐藏hide(3000) 显示 show() 淡出fadeOut() 淡入fadeIn() 上滑slideUp() 下滑slideDown() 自定义animate({},时间);

**19:30宿舍-休息**

回宿舍前独自一人撸了一顿烤串。好久没有吃顿舒服的饭，这下满足了。虽然烤串点的特辣，辣的我满身大汗，但久违的幸福感突然涌上心头，把眼泪给冲出来了。也想起了很多往事。

原本以为今天就是前端课程的最后一天了，结果被告知明天还有一天JQ的课程。不过也无所谓了，因为后天的数据库课程也是这个沙雕讲师负责授课。

从这个讲师开始讲前端的课程开始到现在，我一直处于一种颓废的状态，对学习的热情也降到了极低，上课也就是抄一下代码，然后就开始摸鱼。仿佛已经与班上的混子融为了一体。越来越觉得自己的水平，放在IT界里的存在感真的是太渺小了。

撸串的时候，周围比较安静，两瓶啤酒下肚也是开始感觉到了一丝寂寞。稍微反思了一下自己近期的所作所为，似乎感觉自己开始在慢慢地与自己最初的目标拉开距离，开始对自己的固执产生那么一丝的疑惑。难道我还是太天真了？

…………

…………

我是在慢慢失去信仰，和之前撸串时瞎鸡儿乱想的一切都没有任何的关系，只是因为我的欲望没有那么强烈了，我的信仰耐久值红了！但并不代表我真的就开始捞了！在顾及那些闲言碎语的时候，和我一同培训并且彼此相互竞争的那个傻缺计算机本科生，在我不知道的时候水平已经超过我了！这TM能忍？哔都被他给装了，这TM能忍！？

……………

……………

无论未来如何，无论我变得有多捞、多懒、多废、多特么中二，既然我说过我不会放弃，那么我就不会停下。

我这一路活到现在超越并实现了了无数个曾经的目标，我向我自己证明了我想要东西的总能拿到手。

所以也不差这一次。

肺腑之言，无惧嘲讽。

今日感想总结：为了实现梦想（欲望），就应该全力以赴（不择手段）。

今日评分： \*/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