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

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(9天) ←当前为该阶段第8天

WEBBASIC(7天)

JQUERY(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：100天

预计剩余：61天

注：上述信息随时可能发生更改，仅供参考。

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

**2019年4月19日**

**Java就业班-第44天**

**8:41实训室2-早自习**

今天因为调试直播音频所以晨讲还没有开始。

项目经理说明了一下，大概下个星期开始就会进入非常简单的web前端的课程内容，没有什么逻辑要求。就是一些死记硬背的标签内容。这一段时间讲的课程内容比较辛苦，每天都在接触新的知识是很正常的，所以到下个星期以后，课程内容变简单了以后时间就会宽裕很多，到时候再重新回过头来写一写现在讲的web开发项目，这是决定今后是否能够从事开发岗位的硬实力，一定要重视。

晨讲的主题是——封装特性。

代码和内容都是从视频上抄下来的……没什么好记的。

**9:05实训室2-正课**

开始上课。

今天还是接着昨天的课程继续拓展课程项目的业务开发。

今天的第一个开发任务就是写一个检测注册的用户名是否已存在的检测代码。

在开发之前，先熟悉一下如何添加超链接。

Index.html中添加如下代码即可：

<a href="./reg.html">注册会员</a>

<a href="http://acfun.cn">ACfun主站</a>

**10:36实训室2-正课**

功能实现：

package com.webserver.servlet;

import java.io.File;

import java.io.RandomAccessFile;

import java.util.Arrays;

import com.webserver.http.HttpRequest;

import com.webserver.http.HttpResponse;

/\*\*

\* 处理注册业务

\* @author Administrator

\*

\*/

public class RegServlet {

public void service(HttpRequest request,HttpResponse response) {

System.out.println("RegServlet:开始处理注册……");

/\*

\* 1.通过request获取用户在页面上输入的注册信息

\*

\* 2.将该用户的注册信息写入到文件user.dat中

\*

\* 3.设置response响应注册结果页面

\*/

//1

String username = request.getParameter("username");

String password = request.getParameter("password");

String nickname = request.getParameter("nickname");

int age = Integer.parseInt(request.getParameter("age"));

System.out.println("username:" + username);

System.out.println("password:" + password);

System.out.println("nickname:" + nickname);

System.out.println("age:" + age);

/\*

\* 2

\*/

try {

RandomAccessFile raf = new RandomAccessFile("./user.dat","rw");

/\*

\* 首先读取user.dat文件中的每个用户信息

\* 比对用户名与当前注册用户的名字是否一致，

\* 若一致则说明该用户名已经被使用，

\* 这时设置response响应用户提示页面：该用户已存在。

\* 否则执行下面原有的注册操作。

\*

\* 用户提示页面：reg\_fail.html

\*/

raf.seek(0);//指针归零

int sk = 0;//调节指针用变量

System.out.println("文本字节长度~~~~~~~~~~~~~~~~~~~~~~~~~~~："+raf.length());

for (int i = 0; i <=raf.length()/100; i++) {

byte name[] = new byte[32];//定义字节组为32

raf.read(name);//按字节组读取

String un = new String(name,"utf-8").trim();//读取username的字符串

if(un.equals(username)) {//判断

File failPage = new File("./webapps/myweb/reg\_fail.html");

response.setEntity(failPage);

break;

}

if(raf.read()==-1) {

//指针拨至最后

raf.seek(raf.length());

//写入用户名

byte data[] = username.getBytes("utf-8");

data = Arrays.copyOf(data, 32);

raf.write(data);

//写入密码

data = password.getBytes("utf-8");

data = Arrays.copyOf(data, 32);

raf.write(data);

//写入昵称

data = nickname.getBytes("utf-8");

data = Arrays.copyOf(data, 32);

raf.write(data);

//写入年龄

raf.writeInt(age);

raf.close();

//3

File successPage = new File("./webapps/myweb/reg\_success.html");

response.setEntity(successPage);

break;

}

raf.seek(sk+=100);//进入下一行

}

} catch(Exception e) {

e.printStackTrace();

}

System.out.println("RegServlet:处理注册业务完毕！");

}

}

注册失败页面：

<**html**>

<**head**>

<**meta** carset=*"utf-8"*>

<**title**>注册失败</**title**>

<**body**>

<**h1** align=*"center"*>该用户已存在！</**h1**>

<**a** href=*"./reg.html"*>返回</**a**>

</**body**>

</**head**>

**11:04实训室2-正课**

新建一个项目——WebServer\_v12

README.txt-----------------

本版本主要工作：

实现注册业务操作

登录流程：1用户访问登录界面，2在登录页面中输入登录信息（用户名与密码），3点击登录按钮，4登录数据提交到服务端处理，5根据登录结果相应页面。

实现：

1.准备页面

login.html:登录界面，表单要求用户输入用户名及密码form提交路径指定为"./login"

login\_sucees.html:登陆成功提示页面

login\_fail.html:登录失败提示页面

无论是用户名输入错误还是密码输入错误，都统一提示登录失败页面，提示内容：用户名或密码不正确

2.在ClientHandler判断请求，若是登录操作也实例化一个处理类LoginSeverlet,并调用其service方法处理

3.创建LoginServlet,在service中完成登录业务。

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

**12:02实训室2-正课**

我已经把功能实现了。

过程很简单，把注册的功能代码直接复制过去稍作修改即可。

估计下午讲师还会讲，代码也没有什么好贴的，详细情况可以在github查看源代码。

**15:05实训室2-正课？**

已经超过平时上课时间1个多小时了，还是不见讲师的人。就上午提出的这点破需求真的有那么难实现吗？

下午天气也是热到原地爆炸，还是不肯开空调，我们这群人就像沙雕一样的坐在教室里受热修仙。

还上不上课了？？不上课我就走的啊？没工夫在这浪费时间。

一想到明天还要整啥趣味运动会我就来气……

**16:15实训室2-正课**

终于开始讲课了……真TNND能混。

**16:24实训室2-正课**

由于LoginServlet和RegServlet中有很多重复代码，所以现在开始重构代码。

新建一个新项目：WebServer\_v13

README.txt------------------

本版本主要工作：重构代码

不同的业务我们会定义相应的Servlet来处理，

因此我们要规范一下所有Servlet，提出一个超类，

并在父类中定义所有Servlet都会使用的功能。

实现：

1.在com.websever.servlet包中定义类：HttpServlet

该类是一个抽象类.

2.在HttpServlet中定义抽象方法：service，用于规定所有的Servlet都必须具备该方法。

3.使现有的Severlet都继承HttpServlet

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

**16:51实训室2-正课2**

功能实现：

package com.webserver.servlet;

import java.io.File;

import com.webserver.http.HttpRequest;

import com.webserver.http.HttpResponse;

/\*\*

\* 所有Servlet的父类(抽象类)

\* @author Administrator

\*

\*/

public abstract class HttpServlet {

//service抽象方法

public abstract void service( HttpRequest request , HttpResponse response) ;

/\*\*

\* 跳转指定路径

\* （TomCat的该方法实际上是通过HttpRequest获取的转发器dispatcher对应的方法）

\*

\* @param path

\* @param request

\* @param response

\*/

public void forward(String path,HttpRequest request , HttpResponse response) {

File file = new File("./webapps" + path);

response.setEntity(file);

}

}

此外还优化了一下代码的可读性：

在HttpConstext中新增CR·LF的常量定义。

在HttpResponse中新增一个自动以CR,LF结尾的println()方法。

**17:15实训室2-正课**

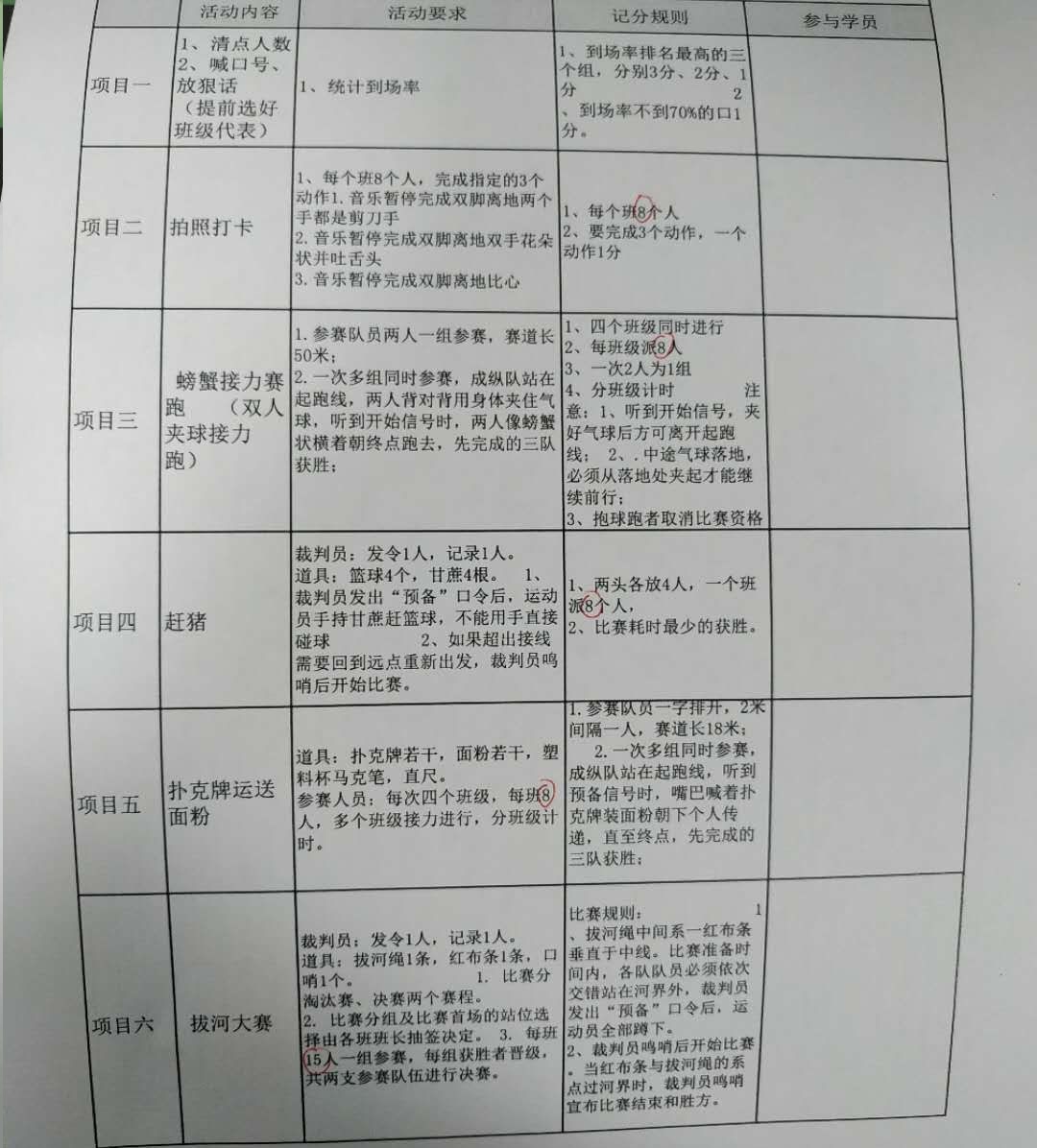
今天的课程到这里了，真是混了一天啊……

热点的不行了，我得出去吹会儿风，透不过气来了。

**18:28实训室2-晚自习**

回到教室准备晚自习看书，然而发生了一个小事件。

关于那个沙雕趣味运动会，我先把图贴出来吧：



简单说明背景一下，班主任表示这项运动是强制所有人参加的，不参加就算旷课，班里4个组，每个组员必须参加一项，组长必须全部参加。

然后就是现在，一个脾气暴躁的组长和班主任怼起来了！

总结一下双方的意思就是：

班主任：“我辛辛苦苦加班加点组织活动，想让你们在学习之余放松一下，你们这是什么态度啊？”

脾气暴躁的组长：（视线盯着代码，纹丝不动。）“这种明显是找我们帮你们搞宣传的活动倒是真的有脸说，搞之前也不通知，上来就是一句全员强制参加。别的不提，老子交钱来这里是学技术的，不是陪你们玩的！”

班主任：（秒怂）“好好好，你们爱怎么样怎么样，想搞学习明天就给我准时到搞自习。不来的一律按旷课处理！”

…………

不管怎么样，我还是蛮感谢那位暴躁老妹的，毕竟我只是在全程在围观，如果真的强制，我就准备直接旷课了。看样子明天的自习会清净不少。混子和怂包们都去参加了……

**18:43实训室2-晚自习**

行了，看书吧——《深入理解Java虚拟机》。

如果有好的部分或是不太理解的部分我会尽量摘记下来。

“世界上并没有完美的程序，但我们并不因此而沮丧，因为写程序本来就是一个不断追求完美的过程。”

JDK从1.5版本开始，官方在正式文档与宣传上已经不再使用类JDK1.5的命名，只有在程序员内部使用的开发版本号（Developer Version,例如java –version的输出）中才继续沿用1.5、1.6、1.7的版号，而公开版本号（Product Version）则改为JDK5、JDK6、JDK7的命名方式。

世界上第一款商用Java虚拟机：Sun Classic VM（诞生于1996年1月23日）

“Java语言很慢”的印象是由于初期的Classic VM解释器和编译器不能配合工作，这就意味着如果要使用编译器执行，编译器就不得不对每一个方法、每一行代码都进行编译，而无论他们的执行频率是否具有编译的价值。→（JDK1.4时Classic VM彻底被HotSpot VM取代至今。）

三大商业虚拟机（俗称“高性能Java虚拟机”）：sun系列的HotSpot和JRockit，IBM的J9。←（仅限通用平台）

在同一个虚拟机上运行的其他语言与java之间的交互一般都比较容易，但非java语言之间的交互一般都比较繁琐，dynalang项目（←up主没能查到这是个什么东西，如果有知道的大佬还请分享一下，感谢。）就是为了解决这个问题而出现的。

函数式编程的一个重要有点就是这样的程序天然的适合（多核）并行运行。

（up百度查阅了一些关于函数式编程和面向过程编程的介绍和范例代码，但是没能弄明白这两者之间的差异。目前我自己乱推测的差异是：函数式编程主要体现在对多核并行的利用，而面向过程式编程的初期由于当时CPU还是以单核高频为主所以并没有考虑到后续的多核普及现象。←不知道是否正确，望大神纠正。）

**20:40实训室2-晚自习**

啃完了第一章介绍部分，往后似乎需要配置好运行环境和安装Linux系统才方便继续往下阅读。看着时间也差不多了，今天的代码还没有上传。就读到这里了吧。

明天的自习应该还是复习一下这一周的课程项目代码比较好，毕竟能看得懂不能算是理解了，只有自己也能写出来才算嘛。

今日感想总结：讲师给自己找了个理由说讲快了，所以今天的课并不着急。当众直播摸鱼还有理由了可还行。

今日评分：4/10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