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

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(8天) ←当前为该阶段第7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：62天

注：上述信息随时可能发生更改，仅供参考。

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

**2019年4月17日**

**Java就业班-第42天**

**8:30实训室2-早自习**

今日晨讲主题——线程间的交错执行执行中判断时应使用while而不能使用if。

今天晨讲的人员是这个班里的最强战力，计算机科班本科出身的第4组组长。内力深厚这一点所有人都非常佩服，但一直不知道他写代码厉不厉害。今日一见，晨讲内容无可挑剔，但是写代码的基本功就有些太次了……

**9:00实训室2-正课**

开始上课。

首先回顾了昨天的课程内容。

**9:15实训室2-正课**

新建一个项目：WebServer\_v10

今天主要学习并实现的的业务层面的功能。

README.txt--------

本版本主要工作：

从本版本开始，服务端开始支持处理业务的操作。例如用户注册，登录等操作。

以注册为例，流程大致为：

1用户打开注册页面，2在页面上输入注册信息，3点击注册按钮，4数据提交到服务端，5服务端将数据保存，6相应注册结果页面（注册成功或失败页面）。

本版本完成内容：

1.创建注册页面，并理解form表单应用

2.服务端HttpRequest可以解析用户提交的表单数据

实现：

1. 在webapps/myweb目录下新建注册页面：reg.html

2.重构HttpRequest解析请求的操作

由于reg.html页面以GET形式提交了一个表单，这时请求的请求行中URL部分的内容变味了如下的样子：

/myweb/reg?username=xxx&password=xxx&...

及请求的抽象路径中包含了用户传递过来的参数。

因此我们对抽象路径部分要做进一步的解析工作。

在HttpRequest中在定义三个属性：

String requestURI 保存url中"?"左侧的请求部分

String queryString 保存url中"?"右侧的参数部分

Map parameters 保存解析出来的每一个参数

创建一个方法：parseURL，用来进一步解析请求行中抽象路径url的内容。

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

**9:20实训室2-正课**

在webapps/myweb目录下新建注册页面：reg.html

代码：

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>用户注册</**title**>

</**head**>

<**body**>

<**center**>

<**h1**>欢迎注册</**h1**>

<**form** action=*"./reg"* method=*"get"*>

<!-- form表单

表单是用来将用户在页面中输入的内容提交给服务端使用的。

注意，只有被form标签包含的输入框的内容才会被提交。

form表单有两个重要的属性：

action:用来指定该表单提交的路径

method:用来指定表单提交的方式

method常用的值为：GET,POST

GET:地址栏形式提交，表单数据会被拼接到URL中提交给服务端

POST：打包形式提交，内容会包含在请求的消息正文中，对于提交

隐私数据或者有上传附件操作时要使用这种方式提交。

-->

<**table** border=*"1"*>

<**tr**>

<**td**>用户名</**td**>

<**td**><**input** name=*"username"* type=*"text"*></**td**>

</**tr**>

<**tr**>

<**td**>密码</**td**>

<**td**><**input** name=*"password"* type=*"password"*></**td**>

</**tr**>

<**tr**>

<**td**>年龄</**td**>

<**td**><**input** name=*"age"* type=*"text"*></**td**>

</**tr**>

<**tr**>

<**td**>昵称</**td**>

<**td**><**input** name=*"nickname"* type=*"text"*></**td**>

</**tr**>

<**tr**>

<**td** align=*"center"* colspan=*"2"*>

<**input** type=*"submit"* value=*"立即注册"*>&nbsp;&nbsp;&nbsp;&nbsp;

<**input** type=*"reset"* value=*"重置"*>

</**td**>

</**tr**>

</**table**>

</**form**>

</**center**>

</**body**>

</**html**>

**11:31实训室2-正课**

※在做某些需要模拟客户输入信息的测试的时候，切记一定要基于正常的实际情况进行测试。严禁随意输入如：asd/123123等信息进行测试。

**14:00实训室2-正课**

功能实现：

package com.webserver.http;

import java.io.IOException;

import java.io.InputStream;

import java.net.Socket;

import java.util.HashMap;

import java.util.Map;

import com.webserver.core.EmptyRequestException;

/\*\*

\* 请求对象

\* 该类的每一个实例用于表示浏览器发送过来的一个请求内容，

\* 每个请求由三部分构成（请求行，消息头，消息正文）

\* @author Administrator

\*

\*/

public class HttpRequest {

/\*

\* 请求行相关信息

\*/

//请求方式

private String method;

//请求资源的抽象路径

private String url;

//请求时用的协议版本

private String protocol;

//保存url中"?"左侧的请求部分

private String requestURI;

//保存url中"?"右侧的参数部分

private String queryString;

//保存解析出来的每一个参数（key:参数名 value：参数值）

private Map<String,String> parameters = new HashMap<>();

/\*

\* 消息头相关信息

\*/

/\*

\* key:消息头名

\* value:消息头对应的值

\*/

private Map<String,String> headers = new HashMap<String, String>();

/\*

\* 消息正文相关信息

\*/

/\*

\* 与连接相关的属性

\*/

private Socket socket;

private InputStream in ;

/\*\*

\* 初始化HttpRequest对象

\* 初始化的过程就是解析请求的过程。

\* 当前HttpRequest对象就表示浏览器发送过来的

\* 这个请求内容了。

\*/

public HttpRequest(Socket socket) {

System.out.println("HttpRequest:开始解析请求……");

try {

this.socket = socket;

this.in = socket.getInputStream();

/\*

\* 解析请求的三步：

\* 1.解析请求行

\* 2.解析消息头

\* 3.解析消息正文

\*/

parseRequestLine();

parseHeaders();

parseContent();

} catch (Exception e) {

e.printStackTrace();

}

System.out.println("HttpRequest:解析请求完毕！");

}

/\*\*

\* 解析请求行

\* @throws EmptyRequestException

\*/

private void parseRequestLine() throws EmptyRequestException {

System.out.println("开始解析请求行……");

/\*

\* 1.通过输入流读取第一行字符串（请求行内容）

\* 2.将请求行内容按照空格拆分为三部分

\* 3.将三部分内容设置到对应属性上（method,url,protoco）

\*/

try {

String line = readLine();

System.out.println("开始解析请求行：" + line);

/\*

\* 判断是否为空请求

\*/

if("".equals(line)) {

throw new EmptyRequestException();

}

/\*

\* 后期循环接受客户端连接后，下面代码可能会出现

\* 数组下标越界，这是由于空请求引起的，后面会解决

\*/

String data[] = line.split("\\s");

this.method = data[0];

this.url = data[1];//这里会出现下标越界

this.protocol = data[2];

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

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

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

//进一步解析抽象路径部分

parseURL();

} catch (EmptyRequestException e) {

System.out.println("空请求……");

} catch (Exception e) {

e.printStackTrace();

}

System.out.println("解析请求行完毕！");

}

private void parseURL() {

System.out.println("开始解析抽象路径……");

/\*

\* 一个请求的请求行大众抽象路径部分有两种情况：

\* 1.不含有参数的，如：

\* /myweb/index.html

\* 2.含有参数的，如：

\* /myweb/reg?username=xxx&password=xxx&...

\* 因此我们对抽象路径进一步解析：

\* 1.首先判断url的值是否含有"？"

\* 1.1若不含有"?"则直接将url的值赋给属性requestURI,当前方法结束。

\* 1.2若含有"?"执行步骤2

\*

\* 2.将url按照"?"拆分为两部分，第一部分应当是请求部分，赋值给属性requestURI.

\* 第二部分应当是参数部分，赋值给属性queryString

\*

\* 3.进一步解析queryString,将其按照"&"拆分为若干个参数，

\* 每一个参数再按照"="拆分为参数名与参数值。

\* 并将参数名作为key,参数值作为value保存到属性parameters这个Map中完成解析工作。

\*/

if(url.contains("?")==false) {

requestURI = url;

}else {

String data[] = url.split("\\?");

requestURI = data[0];

if(data.length>1)

queryString = data[1];

data = queryString.split("&");

for (String pm : data) {

String kv[] = pm.split("=");

if(kv.length>1)

parameters.put(kv[0], kv[1]);

else {

parameters.put(kv[0], null);

}

}

}

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

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

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

//进一步解析

System.out.println("解析抽象路径完毕！");

}

/\*\*

\* 解析消息头

\*/

private void parseHeaders() {

System.out.println("开始解析消息头……");

/\*

\* 1.循环调用readLine方法读取每一个消息头

\* 2.将消息头按照：“拆分，并将消息头的名字作为key，

\* 消息头的值作为value保存到属性headers这个Map中”

\* 3.如果调用readLine方法返回的是一个空字符串，

\* 则说明了本次单独读取到了CRLF，那么就可以停止解析消息头了。

\*/

try {

String line = null;

while(true) {

line = readLine();

if("".equals(line)) {

break;

}

String data[]= line.split(": ");

headers.put(data[0], data[1]);

}

} catch (Exception e) {

e.printStackTrace();

}

System.out.println("消息头：" + headers);

System.out.println("解析消息头完毕！");

}

/\*\*

\* 解析消息正文

\*/

private void parseContent() {

System.out.println("开始解析消息正文……");

System.out.println("解析消息正文完毕！");

}

/\*\*

\* 通过对应客户端的输入流读取一行字符串

\* （以CRLF结尾）

\* @return

\* @throws IOException

\*/

private String readLine() throws IOException {

//读取一行字符串，以CRLF结尾

StringBuilder builder = new StringBuilder();

//c1表示上次读取到的字符，c2表示本次读取到的字符

int c1 = -1,c2 = -1;

while((c2 = in.read())!=-1){

//是否连续读取到了CR,LF

if(c1==13 && c2==10) {

break;

}

builder.append((char)c2);

c1 = c2;

}

return builder.toString().trim();

}

public String getMethod() {

return method;

}

public String getUrl() {

return url;

}

public String getProtocol() {

return protocol;

}

/\*\*

\* 获取指定名字消息头对应的值

\* @param name

\* @return

\*/

public String getHeaders(String name) {

return headers.get(name);

}

}

**15:48实训室2-正课**

新建一个项目：WebServer\_v11

继续实现下一个业务层面的功能——注册。

README.txt--------

本版本主要工作：

实现注册业务操作

上一个版本中，HttpRequest已经将页面表单上的数据解析完毕。

因此，我们完成ClientHandler的第二步操作，处理请求环节。

这里要添加一个新的环节，当判断请求的是一个业务时，我们就处理这个业务，

否则在判断该请求是否为一个静态资源。

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

**16:40实训室2-正课**

新建一个负责处理注册业务的包和类——com.webserver.servlet→RegServlet

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");

//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();

} catch(Exception e) {

e.printStackTrace();

}

//3

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

response.setEntity(successPage);

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

}

}

**17:21实训室2-正课**

新建一个reg\_success.html用于展现注册成功后的信息反馈。

代码：

<**html**>

<**head**>

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

<**title**>注册成功</**title**>

<**body**>

<**h1** align=*"center"*>恭喜您注册成功！</**h1**>

</**body**>

</**head**>

</**html**>

**17:30实训室2-下课时间**

**功能实现：**

package com.webserver.core;

import java.io.File;

import java.net.Socket;

import java.util.HashMap;

import java.util.Map;

import com.webserver.http.HttpContext;

import com.webserver.http.HttpRequest;

import com.webserver.http.HttpResponse;

import com.webserver.servlet.RegServlet;

/\*\*

\* 用于处理客户端请求

\*

\* @author Administrator

\*

\*/

public class ClientHandler implements Runnable {

private Socket socket;

public ClientHandler(Socket socket) {

this.socket = socket;

}

public void run() {

try {

System.out.println("ClientHandler:开始处理请求");

//1.准备工作

//1.1实例化请求对象，解析请求

HttpRequest request = new HttpRequest(socket);

//1.2实例化响应对象

HttpResponse response = new HttpResponse(socket);

// 2.处理请求

/\*

\* 2.1通过request获取requestURI,用来得知用户请求的资源的路径

\* 2.2从webapps目录下根据该资源路径找到对应资源

\* 2.3判断该资源时是否真实存在

\* 2.4存在则响应该资源

\* 2.5不存在则响应404页面

\*/

// myweb/index.html

String path = request.getRequestURI();

//判断该请求是否为请求业务处理

if("/myweb/reg".equals(path)) {

//处理注册操作

System.out.println("ClientHandler:开始注册~~~~~~~~~~~~~~~~~~~~------+++++++++++++++++++++++++++++++++++++++！");

RegServlet servlet = new RegServlet();

servlet.service(request, response);

}else {

// 通过路径找到webapps目录下对应资源。

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

// 判断用户请求的资源是否真实存在

if (file.exists()) {

// 将该资源以标准的HTTP响应格式发送给客户端

// 将要相应的资源设置到response的entity属性上

System.out.println("ClientHandler:资源已找到！");

//获取资源名称

response.setEntity(file);

String fileName = file.getName();

System.out.println("资源名：" + fileName + "-----------!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");

//截取资源名的后缀。

String ext = fileName.substring(fileName.lastIndexOf(".")+1);

System.out.println("资源名后缀：" + ext + "--------------!!!!!!!!!!!!!!!!!!!!!!!!!!!!");

//添加响应头

response.putHeader("Content-Type", HttpContext.getMimeType(ext));

response.putHeader("Content-Length", file.length()+"");

} else {

System.out.println("ClientHandler:资源不存在！");

File file404 = new File("webapps/root/404.html");

// 设置状态代码和描述

response.setStatusCode(404);

response.setStatusReason("NOT FOUND");

// 设置响应正文为404页面

response.setEntity(file404);

}

}

// 3.发送响应

response.flush();

System.out.println("ClientHandler:处理完毕！");

} catch (Exception e) {

e.printStackTrace();

} finally {

// 处理完毕后与客户端断开连接

try {

socket.close();

} catch (Exception e2) {

e2.printStackTrace();

}

}

}

}

**17:45实训室2-休息时间**

今天的课程就到这里了，主要是围绕课程项目的业务处理方面需要用到的功能和编写过程进行了讲解。

理解起来并不困难，有流程图帮助理解的话就更不是问题了。

在实际开发中应该也有需要帮助理解用的各种流程图吧。

晚饭没什么胃口，休息一会准备晚自习写代码，试试看能不能把前天的练习题写完。

**20:58实训室2-晚自习**

没想到，我还是遇到瓶颈了，集合的功能我还是不怎么会用……

到底要怎样才能取出保存到List集合中的对象的属性呢？（我为什么不用数据库呢？）

我甚至都有点没弄明白我为什么要写个对象集合。

自己把自己绕晕了……

或许我应该放弃这道题偏题，有这时间我还不如多看会儿书，毕竟买了那么多书都还没看几页。

**21:10实训室2-晚自习**

晚自习也下了，混子们大都撤了。

看到左右的同桌都在盯着屏幕郁闷着，我也就顺便帮旁边的同桌解决了几个代码上的小问题。

然后他们俩也走了。

话说晚自习开始前班主任好像过来讲了些什么，这周六要强制参加个什么娱乐运动会，机构自己办的，会把所有的学员召集在一起，尽管大部分人表示没兴趣然而是强制的……然后就开始收那些申请要考证的学员的学费了。

结果来看，要考证的人不多，班里人数的一半不到，而且主要是一些学习水平中等偏下的学员交了钱。

**21:30实训室2-晚自习**

今天就这样吧，明天把书都带过来看好了，一天下来也确实怪累的。回去还要看书的话，精神上真有些受不了。

今日感想总结：从开始培训到今天为止正好2个月了，我从未如此渴望过回家玩游戏，果然失去了才会知道珍惜……

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