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

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)                         该阶段已结束

SERVLETJSP(13天)      ←当前为该阶段第11天

SPRINGMYBATIS01(8天)

AJAX(2天)

~~SPRINGMYBATIS02(12天)~~（该阶段已被取消）

~~LINUX(2天)~~                        （该阶段已被取消）

~~NGINX(2天)~~                      （该阶段已被取消）

~~REDIS(2天)~~                       （该阶段已被取消）

~~T-STORE(5天)~~                            （该阶段已被取消）

预计合计：100天

预计剩余：16天 （13天正课+3天自习）

注1：上述信息随时可能发生更改，仅供参考。

注2：被取消表示若按照正常流程就业则直接上班，不用再继续到来机构上课。

反之，则需要继续到机构上完全部课程。

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

**2019年6月11日**

**Java就业班-第84天**

**8:30实训室2-早自习**

今日晨讲主题——冒泡排序和选择排序的效率比

今天是同桌的晨讲，主要展示了两种算法之间的排序效率差异，通过代码随机生成一个长度10W的乱序数组进行排序。然而在机构的演示电脑上跑出的实际的结果并没有太大的差距，我觉得理论上任何排序方法都会比冒泡排序要快才对吧……同桌最终还是没能解释出为什么会出现这种结果（结果是选择排序之比冒泡排序要快0.2秒）。

**9:02实训室2-正课**

开始上课，简单回顾了昨天的课程内容之后，开始进行Servlet过往所学的小结。

今天的主要内容就是对所学的知识进行一次相对完整的小复习，然后如果有多的时间就讲解一点新内容。

**17:32实训室2-下课时间**

今日课程笔记：

# 1.Servlet小结

## (1)Servlet基础

1)什么是Servlet?

2)如何写一个Servlet?

3)Servlet是如何执行的?

4)Http协议(了解)

a.什么是http协议?

b.数据包的结构

c.两种请求方式

## (2)Servlet核心

1)如何读取请求参数值?

2)Servlet输出中文，如何处理?

3)表单包含有中文参数值，如何处理?

4)转发与重定向

a.什么是重定向?

b.如何重定向?

c.重定向的特点

d.什么是转发?

e.如何转发?

f.转发的特点

g.转发与重定向的区别

5)生命周期

a.什么是Servlet的生命周期?

b.分成哪几个阶段?

c.相关的接口与类

6)Servlet上下文

a.什么是Servlet上下文?

b.如何获得Servlet上下文?

c.作用

7)Servlet线程安全问题

8)路径问题

9)容器如何处理请求资源路径?

10)如何让一个Servlet处理多种请求?

## (3)状态管理

1)什么是状态管理?

2)Cookie

a.什么是Cookie?

b.Cookie的工作原理

c.如何添加Cookie?

d.三个问题(编码问题、生存时间问题、路径问题)

e.如何读取Cookie?

f.Cookie的限制

3)Session

a.什么是Session?

b.Session的工作原理?

c.如何获得Session?

d.绑订数据

e.Session超时

f.删除Session

g.Session与Cookie的区别

## (4)数据访问

a.如何使用jdbc访问数据库

b.DAO

## (5)过滤器与监听器

a.什么是过滤器?

b.如何写一个过滤器?

c.过滤器的优先级

d.初始化参数

e.优点

f.什么是监听器?

g.如何写一个监听器?

## (6)案例

用户管理

登录、Session验证

# 2.jsp基础

## (1)什么是jsp?

## (2)如何写一个jsp?

1)html(css,js)

直接写即可。

2)java代码

a. <% %>

b. <%= %>

c. jsp声明 (a1.jsp)

<%! %>

3)隐含对象

a.什么是隐含对象?

b.为什么可以使用这些隐含对象?

c.有哪些隐含对象?

out,request,response,session,application

pageContext(页面上下文 a2.jsp,a3.jsp):

容器会为每一个jsp实例创建唯一的一个符合

PageContext接口要求的对象，该对象一般称之为

页面上下文。

页面上下文有两个特点:

唯一性： 一个jsp实例对应一个pageContext。

持久性: 只要jsp实例还在，pageContext就会一

直存在。

作用1: 绑订数据。

注：绑订到pageContext上的数据不能否被共享，只能被对应的jsp实例访问到。

作用2:

提供了一些方法，用来获得其它所有的隐含对象。

config(a4.jsp): ServletConfig

exception(a5.jsp,a6.jsp):

用来获异常的一个简短的描述。

page(了解):

jsp实例本身(jsp对应的servlet对象)

4)指令

a.什么是指令?

b.有哪些常用的指令?

page指令:

pageEncoding属性

import属性

contentType属性

errorPage属性：指定一个异常处理页面。当当前页面发生

异常，则容器会调用异常处理页面。

isErrorPage属性：缺省值是false,如果值为true,则可以

使用exception隐含对象。

session属性(a7.jsp)：缺省值是true,如果值为false,则

容器不再添加获得session对象的代码了，也就是说，

不能够使用session隐含对象了。

include指令:

file属性：指定被包含的文件名。

taglib指令:

引入jsp标签。

5)注释 (a8.jsp)

a. <!-- 注释内容 -->

被注释的内容如果是java代码，java代码会执行。

b. <%-- 注释内容 --%>

被注释的内容如果是java代码，java代码不会执行。

## (3)jsp是如何执行的?

step1.容器要将jsp转换成一个Servlet。

html(css,js) ----> service方法里，使用out.write输出。

<% %> -------> 照搬到service方法里。

<%= %> -------> service方法里，使用out.print输出。

<%! %> -------> 增加新的属性和方法。

step2.调用该Servlet。

# 3.jsp标签和el表达式

## (1)jsp标签是什么?

语法类似于html标签(开始标记、结束标签、有一些属性、有标签体内容),

用于替代jsp中的java代码。

注:

因为直接在jsp当中写java代码，不利于页面的维护(

比如，美工去修改带有java代码的jsp就很不方便),

所以，sun才推出了jsp标签技术规范。

使用jsp标签来代替java代码，jsp文件会变得简洁，

利于页面的维护，并且也利于代码的复用。

## (2)el表达式是什么?

一套简单的运算规则，用于给jsp标签的属性来赋值，也可以脱离

jsp标签，直接使用。

## (3)el表达式的使用

### 1)读取bean的属性

javabean(了解):

public 类

public 构造器

实现序列化接口

属性

get/set方法

方式一: (e1.jsp)

${user.username}

执行过程:

容器会依次从pageContext-->request-->

session-->application中查找绑订名为

"user"的对象,找到之后，调用该对象的

"getUsername"方法。

优点:

会将null转换成""输出。

如果依据绑订名找不到对应的对象，不会报

空指针异常（会输出""）。

指定查找范围:

默认情况下，容器会依次查找，可以通过

pageScope、requestScope、sessionScope、

applicationScope来指定查找的范围。

比如:${pageScope.user.username}

方式二: (e1.jsp)

${user['username']}

等价于${user.username},另外，[]里面可以出现

绑订名，比如${user[s1]}。[]里面还可以出现从0

开始的下标，用于访问数组中的指定下标的某个元素。

### 2)做一些简单的运算 (e2.jsp)

a.算术运算

+,-,\*,/,%

注: + 只能求和！

b.关系运算

c.逻辑运算

d.empty运算

# 练习

Employee类(ename,salary,age),使用el表达式输出员工

的所有属性值。

**18:45实训室2-晚自习**

班主任强调明天就要收学员们制作的个人简历了，希望各位能利用今天晚自习的时间将简历制作好，尽可能提前发送到班主任邮箱。

我也正好准备把自己以前的简历更新修改一下，改好了直接给她发过去得了。

**19:45实训室2-晚自习**

我正在修改简历排版，而班主任突然出现在我的背后否定了我的简历样式……啥？

理由是没有写项目经验，并对全班学员讲到，项目经验一定要写上。

我就奇了怪了，我们哪TM有项目经验？到目前为止连基础都还没有整明白，还强行让我们写项目经验，难道是要写上我能熟练编写java基础代码吗？

面对质疑，班主任表示稍后会通知项目经理抽空教各位如何写简历，随即便离开教室了……

我觉得不用了，昨天上午的简历制作教学课程时强调的内容，现在就像不是自己说过的话一样。要编就直说嘛，咱们别玩这些虚的好吧。

…………

哦对了，明天到我进行晨讲来着，既然简历的提交明天才是期限，那就明天再说吧。

继续呆在这里也没太大意义了，还是回宿舍准备一下明天的晨讲内容吧。（在教室里反而容易受到干扰。）

**21:42宿舍-休息**

明天早上晨讲的内容终于是准备好了，就讲一道经典的面试题，给HashMap排序。题目看似很简单，但仔细琢磨了之后，我发现这道题涵盖的基础知识面是相当的广，我自身也是因为这道题把我难住才让我对集合有了很深刻的认识。

题目和代码如下：（User类代码就不贴出了，应该不影响代码的阅读）

package Sort;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.HashMap;

import java.util.LinkedHashMap;

import java.util.List;

import java.util.Map.Entry;

import java.util.Set;

/\*\*

\* 面试题：HashMap排序

\* 已知一个HashMap<Integer,User>集合，User有name(String)和age(int)属性。

\* 请写一个方法实现对HashMap排序的功能，该方法接收HashMap<Integer,User>为形参，

\* 返回类型为HashMap<Integer,User>,要求对HashMap中的User的age倒序进行排序。

\* 排序时key=value键值不得拆散。

\* @author Administrator

\*

\*/

public class HashMapTest {

public static void main(String[] args) {

HashMap<Integer, User> users = new HashMap<>();

users.put(1,new User("张三", 22));

users.put(2,new User("李四", 23));

users.put(3,new User("王五", 24));

users.put(4,new User("赵六", 25));

System.out.println("排序前：");

System.out.println(users);

//8.创建新的HashMap用于接收排序后的方法返回值

HashMap<Integer, User> result = hashMapSort(users);

System.out.println("排序后：");

System.out.println(result);

}

//排序方法

public static HashMap<Integer, User> hashMapSort(HashMap<Integer,User> map) {

//1.将map集合转为Set集合

Set<Entry<Integer,User>> set = map.entrySet();

//2.将set转为List集合，借此使用Collections工具类的排序方法sort()

List<Entry<Integer,User>> list = new ArrayList<>(set);

//3.用lambda表达式编写Comparator获取对象的排序规则(按照age倒序排序)

Comparator<Entry<Integer,User>> com = (o1,o2)->

o2.getValue().getAge()-o1.getValue().getAge();

//4.使用Collections工具的sort方法对list进行排序

Collections.sort(list,com);

//5.创建一个新的有序的HashMap子类集合

LinkedHashMap<Integer,User> linked = new LinkedHashMap<>();

//6.利用foreach循环将存储在list中有序排列后的数据放入LinkedHashMap中

for (Entry<Integer, User> entry : list) {

linked.put(entry.getKey(), entry.getValue());

}

//7.返回结果

return linked;

}

}

**22:18宿舍-休息**

相信这个晨讲内容还是有点分量的，至少可以证明给项目经理看，我的基础并没有上次摸底考试的结果显示的那么弱……

好吧，其实还是很弱的……盲目自信不可取。

今天没有听课，上午一直在听音乐摸鱼，而下午则是自学看视频学习web开发的项目案例，自我感觉效果还不错。

目前是准备跟着视频一起编写代码把项目搭建起来，然后再自由发挥魔改一下，最后尝试部署上线。

这样一套下来应该对自身水平的提升会有非常显著的效果吧，毕竟这个web项目的工作量我估计得至少花上3天的时间才能让全部的功能代码跑起来，哪怕是个搬砖体力活也应该能够获得不少的感悟才是。

当前的课程似乎是要结束了，但是从笔记上来看，似乎还是有很多其他的知识是没有讲到的，鉴于这个讲师的讲课进度比较慢，慢到给人一种0.5倍速播放视频的那种感觉，我认为机构是刻意舍弃掉了一部分应当学习的内容。当然，现在看来也是一点都不奇怪的事情了……

既然明天要晨讲，今天就还是早点休息吧。

今日感想总结：今天看到一个连打字都不利索的妹子在机构的训练营上课，我才深刻的认识到，原来机构是真的为了业绩什么人都敢招。

今日评分：3/10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