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

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4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：24天 （20天正课+4天自习）

注：上述信息随时可能发生更改，仅供参考。

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

**2019年5月30日**

**Java就业班-第76天**

**8:50实训室2-早自习**

**今日晨讲主题——反射机制**

**…………**

**今天稍微起晚了，迟到了20分钟，没能赶上晨讲的大部分内容。**

**项目经理提醒到，今晚的摸底考试内容主要以JAVA和数据库为主，由就业部的人员来负责监考。**

**9:10实训室2-正课**

**开始上课。**

**今天上午的课程主要内容就是把前几天写的Servlet代码改一改——重构代码。**

**# 1.DAO的优点**

**a.方便测试**

**比如，将数据访问相关的代码写在Servlet里面，需要部署整个应用**

**才能测试。而将代码写在DAO类里面，可以直接测试。**

**b.方便分工协作**

**c.DAO屏蔽了底层数据存贮介质和数据访问技术，方便系统的维护。**

**即调用者不用关心底层的数据存贮介质是什么(可能是文件、各种**

**数据库、数据仓库)，也不用关心具体的数据访问技术(可能是jdbc,**

**mybatis等)。**

**课程代码：Maven项目Servlet03-lab2**

**重构后Java代码①：UserDAO**

**package dao;**

**import java.sql.Connection;**

**import java.sql.PreparedStatement;**

**import java.sql.ResultSet;**

**import java.util.ArrayList;**

**import java.util.List;**

**import entity.User;**

**import util.DBUtils;**

**/\*\***

**\* 封装了数据访问逻辑**

**\* @author Administrator**

**\***

**\*/**

**public class UserDAO {**

**/\*\***

**\* 将用户信息从数据库删除**

**\* @param user**

**\* @throws Exception**

**\*/**

**public void del(int id) throws Exception {**

**Connection conn = null ;**

**PreparedStatement ps = null ;**

**try {**

**conn = DBUtils.getConn();**

**ps = conn.prepareStatement("delete from t\_user where id=?");**

**ps.setInt(1, id);**

**ps.executeUpdate();**

**} catch (Exception e) {**

**e.printStackTrace();**

**throw e;**

**}finally {**

**DBUtils.close(null, ps, conn);**

**}**

**}**

**/\*\***

**\* 将用户的信息保存到数据库**

**\* @param user**

**\* @throws Exception**

**\*/**

**public void save(User user) throws Exception {**

**Connection conn = null;**

**PreparedStatement ps = null ;**

**try {**

**conn = DBUtils.getConn();**

**ps = conn.prepareStatement("insert into t\_user values(null,?,?,?)");**

**ps.setString(1, user.getUname());**

**ps.setString(2, user.getPwd());**

**ps.setString(3, user.getEmail());**

**ps.executeUpdate();**

**} catch (Exception e) {**

**e.printStackTrace();**

**throw e ;**

**}finally {**

**DBUtils.close(null, ps, conn);**

**}**

**}**

**/\*\***

**\* 将所有用户信息查询出来**

**\* @throws Exception**

**\*/**

**public List<User> findAll() throws Exception{**

**List<User> users = new ArrayList<User>();**

**Connection conn = null;**

**PreparedStatement ps = null ;**

**ResultSet rs = null ;**

**try {**

**conn = DBUtils.getConn();**

**ps = conn.prepareStatement("select \* from t\_user");**

**rs = ps.executeQuery();**

**while(rs.next()) {**

**User user = new User();**

**user.setId(rs.getInt("id"));**

**user.setUname(rs.getString("username"));**

**user.setPwd(rs.getString("password"));**

**user.setEmail(rs.getString("email"));**

**users.add(user);**

**}**

**} catch (Exception e) {**

**e.printStackTrace();**

**throw e ;**

**}finally {**

**DBUtils.close(rs, ps, conn);**

**}**

**return users;**

**}**

**}**

**重构后Java代码②：AddUserServlet**

**package web;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import dao.UserDAO;**

**import entity.User;**

**/\*\***

**\* 添加用户数据处理**

**\* @author Administrator**

**\***

**\*/**

**public class AddUserServlet extends HttpServlet {**

**public void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**request.setCharacterEncoding("utf-8");**

**response.setCharacterEncoding("utf-8");**

**response.setContentType("text/html;charset=utf-8");**

**PrintWriter out = response.getWriter();**

**// 读取用户信息**

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

**String pwd = request.getParameter("pwd");**

**String email = request.getParameter("email");**

**System.out.println(username);**

**// 将用户信息插入到数据库**

**UserDAO dao = new UserDAO();**

**try {**

**User user = new User();**

**user.setUname(username);**

**user.setPwd(pwd);**

**user.setEmail(email);**

**dao.save(user);**

**//重定向到用户列表**

**response.sendRedirect("list");**

**} catch (Exception e) {**

**/\***

**\* step1.记日志。**

**\* 将异常的所有信息记录下来，一般会记录到文件里面**

**\*/**

**e.printStackTrace();**

**/\***

**\* step2.看异常能否恢复，如果不能恢复**

**\* （比如，数据库服务暂停，网络中断等，**

**\* 一般把这样的异常称之为系统异常），**

**\* 则提示用户稍后重试。如果能够恢复，则立即恢复。**

**\*/**

**out.println("系统繁忙，请稍后再试。");**

**}**

**}**

**}**

**重构后Java代码③：ListUserServlet**

**package web;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import java.util.List;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import dao.UserDAO;**

**import entity.User;**

**/\*\***

**\* 查询所有用户信息数据处理**

**\* @author Administrator**

**\***

**\*/**

**public class ListUserServlet extends HttpServlet {**

**public void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**response.setContentType("text/html;charset=utf-8");**

**PrintWriter out = response.getWriter();**

**// 将所有用户信息查询出来**

**UserDAO dao = new UserDAO();**

**try {**

**List<User> users = dao.findAll();**

**// 依据查询到的用户信息，输出表格**

**out.println("<table width='60%' border='1' cellpadding='0'>");**

**out.println("<tr>"**

**+ "<td>ID</td>"**

**+ "<td>用户名</td>"**

**+ "<td>密码</td>"**

**+ "<td>邮箱</td>"**

**+ "<td>操作</td>"**

**+"</tr>");**

**for (User u : users) {**

**int id = u.getId();**

**String username = u.getUname();**

**String password = u.getPwd();**

**String email = u.getEmail();**

**out.println("<tr>"**

**+ "<td>" + id + "</td>"**

**+ "<td>" + username + "</td>"**

**+ "<td>" + password + "</td>"**

**+ "<td>" + email + "</td>"**

**+ "<td><a href='del?id=" + id + "'>删除"+ "</td>"**

**+ "</tr>");**

**}**

**out.println("</table>");**

**out.println("<p><a href='addUser.html'>添加用户</a></p>");**

**} catch (Exception e) {**

**e.printStackTrace();**

**out.println("系统繁忙，稍后重试。");**

**}**

**}**

**}**

**重构后Java代码④：DelUserServlet**

**package web;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import dao.UserDAO;**

**/\*\***

**\* 删除用户数据处理**

**\* @author Administrator**

**\***

**\*/**

**public class DelUserServlet extends HttpServlet{**

**public void service(HttpServletRequest request , HttpServletResponse response) throws ServletException,IOException{**

**response.setContentType("text/html;charset=utf-8");**

**PrintWriter out = response.getWriter();**

**//获取要删除的id参数**

**int id = Integer.parseInt(request.getParameter("id"));**

**UserDAO dao = new UserDAO();**

**try {**

**dao.del(id);**

**//重定向到用户列表**

**response.sendRedirect("list");**

**} catch (Exception e) {**

**e.printStackTrace();**

**out.println("系统繁忙，请稍后再试。");**

**}**

**}**

**}**

**# 2.Servlet的生命周期**

**## (1)什么是Servlet的生命周期?**

**容器如何创建Servlet对象、如何对其进行初始化操作、如何**

**调用其方法来处理请求，以及如何销毁该对象的整个过程。**

**14:00实训室2-正课**

**下午内容主讲Servlet的生命周期。这一部分的内容相当重要，必须牢记。**

**## (2)Servlet的生命周期分成哪几个阶段?**

**### 1)实例化**

**a.什么是实例化?**

**容器调用Servlet的构造器，创建Servlet对象(实例)**

**b.什么时候实例化?**

**情形1:**

**默认情况下，容器收到请求之后，才会创建相应的实例。**

**### 2)初始化**

**a.什么是初始化?**

**容器在创建好Servlet实例之后，会立即调用该实例的init方法。**

**注：**

**该方法只会调用一次！**

**b.GenericServlet已经提供了init方法的实现。**

**将容器传递过来的ServletConfig对象保存下来了，并且提供了**

**一个方法(getServletConfig)来获得ServletConfig对象。**

**c.如何实现自已的初始化处理逻辑?**

**只需要重写GenericServlet的init方法即可。**

**注:**

**重写的是不带参的init!**

**d.初始化参数**

**step1.配置初始化参数**

**<init-param>**

**<param-name>company</param-name>**

**<param-value>IBM</param-value>**

**</init-param>**

**step2.使用ServletConfig对象提供的getInitParameter**

**方法来读取初始化参数。**

**![](init.png)**

**### 3)就绪(调用)**

**a.什么是就绪?**

**容器调用Servlet实例的service方法来处理请求。**

**b.HttpServlet已经提供了service方法的实现。**

**依据请求类型，调用对应的doXXX方法。**

**注：**

**比如，get请求会调用doGet方法，post请求会调用**

**doPost方法。这些doXXX方法只是抛出了异常。这些**

**doXXX方法需要由开发人员来重写。**

**![](service.png)**

**### 4)销毁**

**a.什么是销毁?**

**容器在删除Servlet实例之前，会调用该实例的destroy方法。**

**该方法只会调用一次。**

**b.如何实现自已的销毁处理逻辑?**

**重写GenericServlet的destroy方法即可。**

**## (3)生命周期相关的接口与类**

**a.Servlet接口**

**init(ServletConfig config);**

**service(ServletRequest req,ServletResponse res);**

**destroy();**

**b.GenericServlet抽象类**

**实现了Servlet接口中的部分方法(init,destroy)。**

**c.HttpServlet抽象类**

**继承了GenericServlet，实现了service方法。**

**课程代码：maven项目Servlet04-lab**

**Java代码：**

**package web;**

**import java.io.IOException;**

**import java.rmi.ServerException;**

**import javax.servlet.ServletConfig;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**//加载和实例化**

**public class SomeServlet extends HttpServlet{**

**public SomeServlet() {**

**System.out.println("SomeServlet's constructor输出");**

**}**

**//初始化**

**@Override**

**public void init() throws ServletException {**

**System.out.println("SomeServlet's init输出");**

**}**

**//调用**

**@Override**

**protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {**

**System.out.println("SomeServlet's doGet输出");**

**}**

**//销毁**

**@Override**

**public void destroy() {**

**System.out.println("SomeServlet's destroy输出");**

**}**

**public void service1(HttpServletRequest request,HttpServletResponse response) throws ServerException,IOException{**

**System.out.println("SomeServlet's service输出");**

**/\***

**\* 通过继承自GenericServlet提供的方法来获得ServletConfig对象。**

**\*/**

**ServletConfig config = getServletConfig();**

**System.out.println(config);**

**/\***

**\* 使用ServletConfig对象提供的方法来读取初始化参数。**

**\*/**

**String company = config.getInitParameter("company");**

**System.out.println(company);**

**}**

**}**

**Web.xml配置代码：**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**

**xmlns="http://java.sun.com/xml/ns/javaee"**

**xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"**

**version="2.5">**

**<display-name>Servlet04-lab</display-name>**

**<servlet>**

**<servlet-name>SomeServlet</servlet-name>**

**<servlet-class>web.SomeServlet</servlet-class>**

**<!-- 配置初始化参数。 -->**

**<init-param>**

**<param-name>company</param-name>**

**<param-value>IBM</param-value>**

**</init-param>**

**<!-- 配置启动加载。当容器启动之后，**

**会依据参数值的大小进行创建相应的实例。**

**参数值是一个大于等于零的整数。值越小越先被创建。 -->**

**<load-on-startup>1</load-on-startup>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>SomeServlet</servlet-name>**

**<url-pattern>/some</url-pattern>**

**</servlet-mapping>**

**</web-app>**

**17:10实训室2-正课**

**# 练习 计算BMI指数,并依据BMI指数，返回体重状况。**

**bmi = 体重 (公斤) / 身高(米) / 身高(米)**

**如果bmi < min,体重过轻；**

**如果bmi > max,体重过重;**

**否则，体重正常。**

**提示：**

**<init-param>**

**<param-name>max</param-name>**

**<param-value>25</param-value>**

**</init-param>**

**<init-param>**

**<param-name>min</param-name>**

**<param-value>19</param-value>**

**</init-param>**

**准备下课了，讲师留了一道练习题当做晚自习作业。**

**18:43实训室2-晚自习**

**然而项目经理通知，等会19:00点摸底考试开考。这次的考试有监考，我有预感这场考试，要凉。**

**20:36实训室2-晚自习**

**摸底考试是笔试，有选择题、概念简答题和编程题组成，编程题一共有4题分别是随机数组排序、字符串截取、js取值和JDBC嵌套查询代码，全部要求手写。除了选择题和简答题我可以勉强应付，编程题基本上都没有写出来。（禁止百度和查笔记。）略有些恼火，也确实不甘，但也无可奈何。**

**20:50实训室2-晚自习**

**今天就到这里了，心态并没有什么波动，即使是考烂了也丝毫不觉得奇怪。毕竟真实的水平如此，也无需做过多的自我安慰。再接再厉吧。**

**其他人？比我更次。**

**今日感想总结：所以到底什么是序列化和反序列化？？**

**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