

**长沙理工大学**

城南学院

高级web技术实验

**实验项目名称**：作业3

**年级**：大二 **班级**：计科1802

**学号**：201886250210 **姓名**：柴继石

**实验日期**：2020**年5月29日**

**指导老师：**曾爱群

|  |
| --- |
| **1、实验目的**  服务器编程 |
| **2、实验内容**  习题6 |
| 3、实验平台  IntelliJ IDEA |
| 4、程序清单、调试和测试结果及分析  **一、简答题**  **1.简述Java程序设计语言的特点**  答：Java具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。Java可以编写桌面应用程序、Web应用程序、分布式系统和嵌入式系统应用程序等。  **2.为什么说Java是一种完全面向对象的程序设计语言？**  答：jvm运行后的一切都是对象。  **3.在面向对象技术中，回答如下问题：**  **（1）什么是类？什么是对象？简述它们之间的关系。**  答：类：具有相同特性（数据元素）和行为（功能）的对象的抽象就是类。因此，对象的抽象是类，类的具体化就是对象，也可以说类的实例是对象，类实际上就是一种数据类型。1653类具有属性，它是对象的状态的抽象，用数据结构来描述类的属性。类具有操作专，它是对象的行为的抽象，用操作名和实现该操作的方法来描述。  对象：对象是人们要进行研究的任何事物，它不仅能表示具体的事物，还2113能表示抽象的规则、计划或事件。对象具有状态，一个对象用数据值来描述它的状态。对象还有操作，用于改变对象的状态，对象及其操作就是对象的行为。对象实现了数据和操5261作的结合，使数据和操作封装于对象的统一体中。  类与对象的关系就如模具和铸件的关系，类的实例化的结果就是对象，而对对象的抽象就是类，属类描述了一组有相同特性（属性）和相同行为的对象。  **（2）创建一个对象有哪些方法？**  答：使用new语句实例化一个对象；  通过反射机制创建对象；  通过clone()方法创建一个对象；  通过反序列化的方式创建对象。  **（3）什么是类的构造函数？构造函数的一般功能是什么？**  答： 当新对象被创建的时候，会调用构造函数。每一个类都有构造函数。在程序员没有给类提供构造函数的情况下，java编译器会为这个类创建一个默认的构造函数。构造函数 ，是一种特殊的方法。主要用来在创建对象时初始化对象， 即为对象成员变量赋初始值，总和new运算符一起使用在创建对象的语句中。特别的一个类可以有多个构造函数 ，可根据其参数个数的不同或参数类型的不同来区分它们，即构造函数的重载。  **4．列举主要的Java开发环境，它们的JDK相同吗？**  答:Eclipse，MyEclipse，IntelliJ Idea，JBuilder。JDK相同  **5.说明Web应用中三层体系结构，并说明其优势。**  三层体系结构：展示层（HTML/JSP页面）、中间控制层（servlet/JSP/Java类）、数据层Mysql等数据库）  在B/S三层体系结构模式下，客户端不再需要安装特定的客户端应用程序，取而代之的是通用的Web浏览器软件，所有的用户业务逻辑都被部署在新的中间件上。中间层往往是一组公共网关接口（CGI）程序，接受web浏览器发送的数据，按照业务逻辑对数据进行处理，并向浏览器返回处理结果。客户端浏览器只负责数据的输入，客户端脚本程序对输入数据进行有效性验证，然后发送到服务器端，此时web服务器就不仅要发送网页，他还负责将王爷发送给脚本引擎，以执行其中的脚本程序，这被视为是Web服务器的一种扩展。  6.（1）引入Date类  （2）代码抛出异常时跳转到errorPage.jsp页面  (3)允许服务器端在本页面创建session 7. 1、request对象 request 对象是 javax.servlet.httpServletRequest类型的对象。 该对象代表了客户端的请求信息，主要用于接受通过HTTP协议传送到服务器的数据。（包括头信息、系统信息、请求方式以及请求参数等）。request对象的作用域为一次请求。  **2、response对象**  response 代表的是对客户端的响应，主要是将JSP容器处理过的对象传回到客户端。response对象也具有作用域，它只在JSP页面内有效。  **3、session对象**  session 对象是由服务器自动创建的与用户请求相关的对象。服务器为每个用户都生成一个session对象，用于保存该用户的信息，跟踪用户的操作状态。session对象内部使用Map类来保存数据，因此保存数据的格式为 “Key/value”。 session对象的value可以使复杂的对象类型，而不仅仅局限于字符串类型。  **4、application对象**   application 对象可将信息保存在服务器中，直到服务器关闭，否则application对象中保存的信息会在整个应用中都有效。与session对象相比，application对象生命周期更长，类似于系统的“全局变量”。  **5、out 对象**  out 对象用于在Web浏览器内输出信息，并且管理应用服务器上的输出缓冲区。在使用 out 对象输出数据时，可以对数据缓冲区进行操作，及时清除缓冲区中的残余数据，为其他的输出让出缓冲空间。待数据输出完毕后，要及时关闭输出流。  **6、pageContext 对象**  pageContext 对象的作用是取得任何范围的参数，通过它可以获取 JSP页面的out、request、reponse、session、application 等对象。pageContext对象的创建和初始化都是由容器来完成的，在JSP页面中可以直接使用 pageContext对象。  **7、config 对象**  config 对象的主要作用是取得服务器的配置信息。通过 pageConext对象的 getServletConfig() 方法可以获取一个config对象。当一个Servlet 初始化时，容器把某些信息通过 config对象传递给这个 Servlet。 开发者可以在web.xml 文件中为应用程序环境中的Servlet程序和JSP页面提供初始化参数。  **8、page 对象**  page 对象代表JSP本身，只有在JSP页面内才是合法的。 page隐含对象本质上包含当前 Servlet接口引用的变量，类似于Java编程中的 this 指针。  **9、exception 对象**  exception 对象的作用是显示异常信息，只有在包含 isErrorPage="true" 的页面中才可以被使用，在一般的JSP页面中使用该对象将无法编译JSP文件。excepation对象和Java的所有对象一样，都具有系统提供的继承结构。exception 对象几乎定义了所有异常情况。在Java程序中，可以使用try/catch关键字来处理异常情况； 如果在JSP页面中出现没有捕获到的异常，就会生成 exception 对象，并把 exception 对象传送到在page指令中设定的错误页面中，然后在错误页面中处理相应的 exception 对象。 8. 1.需求确定 2.分析与设计         (1)架构分析与设计         (2)业务逻辑分析         (3)业务逻辑设计         (4)界面设计 3.开发环境搭建 4.开发-测试-开发-测试9.前者是声明标签，创建的是成员变量。后者是小脚本标签，创建的是局部变量。 10.(1)假设有两个字符串s1,s2，用equals（）方法判断是否相等：  s1.equals(s2);  (2)对空字符串使用trim（）方法会报空指针异常。  解决：加个if判断。if（s1 ！= “”） s1.trim();  11.request.getParameter(“key”)系列方法。  12.可通过表单，URL，application，request，session，cookie传值。  二．编程题  1.页面：    代码：  <!doctype html> <html lang="en"> <head>  <!-- Required meta tags -->  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">   <!-- Bootstrap CSS -->  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">   <script src="https://cdn.jsdelivr.net/npm/jquery@3.4.1/dist/jquery.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>  <script src="https://lib.sinaapp.com/js/jquery/2.0.2/jquery-2.0.2.min.js"></script>   <title>model</title>   <script type="text/javascript">  var ***xmlHttp***;  function createXMLHttpRequest(){  if(***window***.***XMLHttpRequest***){  ***xmlHttp*** = new ***XMLHttpRequest***();  }else {  ***xmlHttp*** = new ***ActiveXObject***("Microsoft.XMLHTTP");  }  }  function validate() {  createXMLHttpRequest();  var username = ***document***.getElementById("username").value;  var url = "validationServlet?username="+username;  alert(url);  ***xmlHttp***.open("GET",url,true);  ***xmlHttp***.onreadystatechange = handleStateChange;  ***xmlHttp***.send(null);  }  function handleStateChange() {  if (***xmlHttp***.readyState == 4) {  if (***xmlHttp***.status == 200) {  var message = ***xmlHttp***.responseXML.getElementsByTagName("message")[0].firstChild.data;   var messageArea = ***document***.getElementById("results");  messageArea.innerHTML = "<p>"+message+"</p>";  }  }  }  function checkUsernameValidation(){  $.ajax({  url:"validationServlet",  type:"POST",  data:***document***.getElementById("username"),  success:function (data) {  alert(data);  },  error:function () {  alert("error");  },  dataType:"text"  });  }   ***window***.onload = function () {  ***document***.getElementById("button").onsubmit=function () {  return checkUsername() && checkPassword() ;  }  }  function checkUsername() {  var username = ***document***.getElementById("username").value;  var reg = /^\S{1,8}$/;  var flag = reg.test(username);  var uPrompt = ***document***.getElementById("username\_prompt");  if (flag) {  uPrompt.innerHTML = "";  return true;  } else {  uPrompt.innerHTML = "用户名格式有误,格式：1-8个任意字符";  return false;  }  }   function checkPassword() {  var password = ***document***.getElementById("password").value;  var reg = /^\w{8,16}$/;  var flag = reg.test(password);  var uPrompt = ***document***.getElementById("password\_prompt");  if (flag) {  uPrompt.innerHTML = "";  return true;  } else {  uPrompt.innerHTML = "密码格式有误。格式：8-16个英文字符或数字";  return false;  }  }  </script>  </head> <body> <div class="container-fluid">  <form method="post" action="userRegisterServlet">  <div class="row">  <div class="col-md-4"></div>  <div class="col-md-4">  <h1 style="margin-top: 100px; text-align: center">新用户注册</h1>  <div class="input-group input-group-lg" style="margin-top: 50px;">  <div class="input-group-prepend">  <span class="input-group-text">用户名</span>  </div>  <input type="text" id="username" name="username" onblur="checkUsername()" class="form-control" aria-label="Sizing example input" aria-describedby="inputGroup-sizing-lg">  <div id="username\_prompt" style="margin-top: 5px"></div>  </div>  <div class="input-group input-group-lg" style="margin-top: 50px; ">  <div class="input-group-prepend">  <span class="input-group-text" >密码</span>  </div>  <input type="password" id="password" name="password" onblur="checkPassword()" class="form-control" aria-label="Sizing example input" aria-describedby="inputGroup-sizing-lg" >  <div id="password\_prompt" style="margin-top: 5px"></div>  </div>  <button type="submit" id="button" class="btn btn-secondary" style="margin-top: 50px; margin-left: 210px">提交</button>  </div>  <div class="col-md-4" style="margin-top: 210px">  <a href="#" onclick="validate()">检查用户名是否可用</a>  <div id="results"></div>  </div>  </div>  </form> </div>  </body> </html> |