**中软国际考题-笔试**

1. 单选题

1、在MVC设计模式中，JavaBean的作用是(B )。

A．Controller    
B．Model    
C．业务数据的封装   
D．View

2、在JSP中调用JavaBean 时不会用到的标记是: ( A )

A.<javabean> B.<jsp:useBean>

C.<jsp:setProperty> D.<jsp:getProperty>

3、JSP中调用JavaBean不会用到的标记是 （ A ）

(A)<javabean>  (B)<jsp:useBean>

(C)<jsp:setProperty>  (D)<jsp:getProperty>

4、关于JavaBean正确的说法是（A）

A.Java文件与bean所定义的类名可以不同，但一定要注意区分字母的大小写

B.在jsp引用文件中引用Bean，其实就是用<jsp:userBean>语句

C.被引用的Bean文件的文件名后缀为.java

D.Bean文件放在任何目录下都可以被引用

5、关于JavaBean,下列的叙述哪一项是不正确的（C）

A. JavaBean的类必须是具体的和公共的,并且具有无参数的构造器

B. JavaBean的类属性是私有的,要通过公共方法进行访问

C. JavaBean和Servlet一样,使用之前必须在项目的web.xml中注册

D. JavaBean属性和表单控件名称能很好地耦合,得到表单提交的参数

6、下列哪个作用范围的Bean，请求响应完成则该Bean即被释放,不同客户的Bean互不相

同（B）

A. application B. request C. page D. session

7、<jsp:useBean id=“” class=“” scope=“”>中默认scope是  ( A )

   A.  page

   B.  session

   C.  request

   D.  application

8、关于<jsp:setProperty name=“” property=“” param|value=“”/>的作用,以下说法正确的是（ A ）。

   A.  setProperty标准动作可以用来对JavaBean对象的属性赋值，替代调用setXxxx方法

   B.  setProperty标准动作可以用来取得JavaBean对象的属性值

C.  setProperty标准动作可以用来对session属性的值进行取值

   D.  setProperty标准动作可以用来取得请求参数的值

9． 对于<jsp:setProperty name=“” property=“” param|value=“”/>中的setProperty的name属性 ,以下说法正确的是（ A ）。

   A.  setProperty的name属性表示JavaBean对象的id值

   B.  setProperty的name属性表示JavaBean对象的param值

   C.  setProperty的name属性表示JavaBean对象的value值

   D.  setProperty的name属性表示JavaBean对象的参数值

10． 对于<jsp:setProperty name=“” property=“” param|value=“”/> 中的setProperty的property属性 ,以下说法正确的是（ D ）。

   A.  property表示需要调用的getXxx方法中的Xxx部分，将首字母变小写。

   B.  property表示需要调用的setXxx方法中的set部分。

   C. property表示需要调用的getXxx方法中的get部分。

   D. property表示需要调用的setXxx方法中的Xxx部分，将首字母变小写。

11.对于<jsp:setProperty name=“” property=“” param|value=“”/> 中的setProperty的param属性 ,以下说法正确的是（ D ）。

   A.  如果getXxx方法的返回值是某一个请求参数的值，则使用param属性指定请求参数名字即可

   B.  如果setXxx方法的参数是某一个请求参数的值，则使用param属性指定请求参数名字即可

   C.  如果JavaBean的某一方法的返回值是某一个请求参数的值，则使用param属性指定请求参数名字即可

   D.  如果JavaBean的某一属性是某一个请求参数的值，则使用param属性指定请求参数名字即可

12、下面哪个是scope的默认值（A）

A.page B.request C.session D. application E.response

13、以下哪个不属于javabean 标签 ( B)

    A．<jsp:useBean>

   B. <jsp:javaBean >

C. <jsp:setProperty>

D. <jsp:getProperty>

14. JavaBean的属性可以是任意类型，并且一个JavaBean可以有 ( C )

A. 一个属性 B. 两个属性 C .多个属性 D. 3个属性

15、关于JavaBean正确的说法是（ A ）

A. Java文件与Bean所定义的类名可以不同但一定要注意区分字母的大小写

B. 被引用的Bean文件的文件名后缀为.java

C. Bean文件放在任何目录下都可以被引用

D. 在JSP文件中引用Bean其实就是用<jsp:useBean>语句

16、在JSP中调用JavaBean时不会用到的标记是（ D ）

A. <jsp:setProperty>

B. <jsp:getProperty>

C. <jsp:useBean>

D. <javabean>

17、关于JavaBean正确的说法是（ D ）

A. Bean文件放在任何目录下都可以被引用

B. 被引用的Bean文件的文件名后缀为.java

C. 在JSP文件中引用Bean其实就是用<jsp:useBean>语句

D. Java文件与Bean所定义的类名可以不同但一定要注意区分字母的大小写

1. 多选题
2. 关于JavaBean下列描述正确的是？（AB）

A、可以实现代码的重复利用

B、可以在内部、网内或者是网络之间进行传输

C、不可以以其他部件的模式进行工作

D、不太好维护和使用

2、关于javaBean，下列的叙述哪些是正确的？（ABD）

A. JavaBean的类必须是具体的和公共的

B. JavaBean的类属性是私有的，要通过公共方法进行访问

C. JavaBean和Servlet一样，使用之前必须在项目的web.xml中注册

D. JavaBean属性和表单控件名称能很好地耦合，得到表单提交的参数

3、以下哪些属于JavaBean开发的优点？（A、B、D）

A. 简化jsp的设计与开发，提高程序的可读性和可维护性

B. JavaBean是可以在多个应用中可重复使用的组件

C. JavaBean可以替代Servlet

D. Getter和setter方法用于访问bean的属性信息

4、在JSP 中调用JavaBean 时会用到的标记是:(BCD)  
(A)<javabean>  (B)<jsp:useBean> (C)<jsp:setProperty>(D)<jsp:getProperty>

5、JavaBean与其它Java类相比而言具有哪些特征（ABCD）

A. 提供一个默认的无参构造函数。

B. 需要被序列化并且实现了Serializable接口。

C. 可能有一系列可读写属性。

D. 可能有一系列的"getter"或"setter"方法。

6、JavaBean遵循的代码规范,以下哪些选项正确（ A,B,C,D）

   A.  必须是public类

   B.  必须有一个无参的public的构造方法

   C.  返回属性的方法为getXxxx()格式

D.  设置属性的方法为setXxx(形式参数)格式

7、 下面哪些是scope的范围？（A、B 、C、D）

A.page B.request C.session D. application E.response

8、以下满足JavaBean创建类的条件是？(A C )

    A．该类需要有getter 和 setter 方法

    B．该类需要main 方法

    C．该类中可以有构造方法

    D．该类中不可以存在构造方法

9、关于JavaBean正确的说法是（ AC ）【选两项】

A. JavaBean文件的文件名后缀为.java

B. Bean文件放在任何目录下都可以被引用

C. Java文件与Bean所定义的类名可以不同但一定要注意区分字母的大小写

D. 在JSP文件中引用Bean其实就是用<jsp:useBean>语句

1. 填空题
2. javaBean的属性可以分为以下四类：\_\_\_ Simple(简单的)\_\_\_\_\_、\_\_\_\_\_Indexed(索引的) \_\_\_\_\_、\_\_\_\_\_ Bound(绑定的)\_\_\_\_\_、\_\_\_\_\_Constrained(约束的) \_\_\_\_\_
3. JavaBean有四个scope, 它们分别\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_ ( page,request,session,application)
4. <jsp:useBean>\_\_\_\_\_操作可以定义一个具有一定生存范围以及一个唯一id的JavaBean的实例
5. JavaBean中用一组\_\_\_\_\_\_\_\_\_\_\_\_设置Bean的私有属性值，\_\_\_\_\_\_\_\_\_\_\_\_获得Bean的私有属性值。(Setter方法、Getter方法)
6. 对于<jsp:setProperty name=“” property=“” param|value=“”/> ,如果JavaBean的setXxx方法的参数是一个常量，则使用\_\_\_\_\_\_\_\_\_\_属性指定即可。(value)
7. Java Bean对象名字用 指定，类型用 指定。（id class）
8. Java Bean中 每个属性通常会有 getter 方法和setter方法
9. 一个JavaBean由三部分组成实现java.io.serializable接口、提供无参数的构造方法、提供getter和setter方法访问它的属性
10. 判断题
11. 在JavaBean中对于Boolean属性，除了可以使用getXXX方法外，还可以使用isXXX方法来获取这个属性（T）
12. JavaBean的绑定机制可以通知监听器一个JavaBean组件的属性发生了变化。（T）
13. JavaBean中一个Indexed索引属性表示一个数组集合值，它不可以使用get方法来取得数组中的值。（F）（可以使用）
14. <jsp:getProperty>中的name及property区分大小写。（ T ）
15. 在JavaBean中有很多方法，其中包含了主方法。（ F ）
16. JavaBean中的属性既可以是public型，也可以是private型。（ F ）
17. <jsp:getProperty>中的name 和property区分大小写. ( T )
18. 编写好了JavaBean，该类可以直接在JSP中调用即可。 （ F ）
19. JavaBean中有很多方法，例如main方法. ( F )
20. <jsp:useBean>操作可以定义一个具有一定生存范围以及一个唯一id的JavaBean的实例.( T )
21. 客户在某个页面修改session作用范围Bean的属性,在其他页面,该Bean的属性会发生同样的变化,不同客户之间的Bean也发生变化。（ F ）
22. JavaBean可以用来封装数据，也可以用来处理具体的业务逻辑。（ T ）
23. setProperty标准动作可以对一些常见数据类型直接转换（T ）
24. JavaBean是用Java语言描述的软件组件模型，实际上是一个Java SE的类（ T ）
25. Java Bean对象名字用id指定，类型用class指定( T )
26. 在jsp:setProperty标准动作可以用来对JavaBean对象的属性赋值，替代调用setXxxx方法 （T）
27. 在<jsp:setProperty name=“”/>中name属性表示JavaBean对象的name值 （F）
28. 在<jsp:setProperty param|value=“”/>中如果setXxx方法的参数是某一个请求参数的值，则使用param属性指定请求参数名（T）
29. 在<jsp:setProperty param|value=“”/>中如果setXxx方法的参数是一个常量，则使用value属性指定即可。（T）
30. jsp:setProperty标准动作不可以对一些常见数据类型直接转换( F )
31. <jsp:getProperty name=”” property=””/>中getProperty标准动作用来调用JavaBean对象的getXxx方法，name是JavaBean对象的id值。 ( T )
32. JSP中的JavaBean类，属性必须是公共，可以没有公共的setter\getter方法。( F )
33. JavaBean 是一个类（ T ）
34. Java可以不用显示定义 构造方法( T )
35. 在JSP中使用JavaBean的标签是<jsp:useBean >。（ √ ）

五、简答题

# 1、JavaBean变量可以通过方法调用或setProperty和getProperty标签设置或读取。为什么还要其他方法？

答案：JavaBean变量有三种方式可被JSP访问；可通过将变量名引用到mybean.myVariable里，从scriptlet直接访问变量；也可以通过恰当的set或get方法设置或访问变量的值；最后，可以使用setProperty和getProperty标签。

引用变量或使用方法提供了尽可能的灵活，允许把变量的值结合在表达式，语句和方法调用里。标签比较简单，允许工具容易操作JavaBean。一般而言，标签，HTML,XML或JSP使得创建开发工具处理文档就更容易了。

2、使用JavaBean时需要注意哪些点？（5分）

答案：

(1)懂得将Bean放在哪个目录下:在Resin 中JavaBean 放在doc\web-inf\classes目录中。

(2)懂得如何使用JBuilder 定义一个Bean.

(3) Java文件和Bean 所定义的类名一定要相同，并且是大小写敏感。

(4) Bean中要声明公共方法，与Bean的名字相同。

(5)懂得如何在JSP文件中引用JavaBean,其实就是<jsp:useBean>的语句。

(6)一定要紧记Java 是区分大小写的。

3、编写JavaBean的基本规范有几条？请分别说明.

1. javabean是一个公共类
2. 实现序列化接口
3. javabean类具有一个公共的无参的构造方法
4. javabean所有的属性定义为私有的
5. 在javabean中，需要对每个属性提供两个公共方法，假设属性名是XXX，要提供的两个方法是：setXXX（）：用来设置属性XXX的值--------getXXX（）用来获取属性XXX的值（若包括类型是boolean，则方法名为isXXX（））
6. 定义javabean时，通常放在一个命名的包下。

4、JavaBean与一般java类的区别是什么?

Java Bean 是可复用的组件，对Java Bean并没有严格的规范，理论上讲，任何一个Java类都可以是一个Bean。但通常情况下，由于Java Bean是被容器所创建，所以Java Bean应具有一个无参的构造器，另外，通常Java Bean还要实现Serializable接口用于实现Bean的持久性;可以认为javaBean是具有如上特性的Java类.

5、如何在JSP页面上使用JavaBean?

<jsp:useBean>在JSP页面中实例化或者在指定范围内使用JavaBean

<jsp:useBean id="标示符" class="类名" scope="作用范围">

<jsp:setProperty>给已经实例化的JavaBean对象的属性赋值

<jsp:setProperty name="JavaBean实例名" property="指定属性名" value="指定属性值" param="request对象中的参数名"/>

<jsp:getProperty>获取指定JavaBean对象的属性值

<jsp:getProperty name="JavaBean实例名" property="属性名"/>

6、简述JavaBean的4个作用域范围

①page：仅在当前页面有效

②request：可以通过HttpRequest.getAttribute()方法获得JavaBean对象

③session：可以通过HttpSession.getAttribute()方法获得JavaBean对象

④application：可以通过application.getAttribute()方法获得JavaBean对象

7、简述request、session和application作用域范围的Bean生命周期（5分）

答案： request请求发起到响应结束；session会话开启到超出会话有效间隔时间或会话被销毁，application服务器启动到服务器关闭。

8、介绍在JSP中如何使用JavaBean  
答案：JavaBean的格式要求:必须有包、public class 声明、属性封装、无参构造、setter\getter  
import 语句导入或者通过<jsp:useBean>

9、scope的范围

答案: page、request、session、application，分别为PageContext范围、HttpServletRequest范围、HttpSession范围、ServletContext范围。如果不指定scope的值，默认为page范围。

10、什么是JavaBean？

答案：JavaBean是一个遵循特定写法的Java类，它通常具有如下特点：

这个Java类必须具有一个无参的构造函数

属性必须私有化。

私有化的属性必须通过public类型的方法暴露给其它程序，并且方法的命名也必须遵守一定的命名规范。

六、程序实现题

1、编写一个简单计算器的后台JavaBean代码（两个数的加减乘除）

|  |
| --- |
| **public** **class** SimpleCalculator {  **private** String first;//第一个操作数  **private** String second;//第二个操作数  **private** **double** result;//计算结果  **private** String operator;//运算符 + - \* /  **public** String getFirst() {  **return** first;  }  **public** **void** setFirst(String first) {  **this**.first = first;  }  **public** String getSecond() {  **return** second;  }  **public** **void** setSecond(String second) {  **this**.second = second;  }  **public** String getOperator() {  **return** operator;  }  **public** **void** setOperator(String operator) {  **this**.operator = operator;  }  **public** **double** getResult() {  **return** result;  }  **public** **void** setResult(**double** result) {  **this**.result = result;  }  //根据不同的运算符进行计算  **public** **void** calculate(){  **double** one=Double.*parseDouble*(first);  **double** two=Double.*parseDouble*(second);  **try**{  **if**(operator.equals("+")){  result=one+two;  }**else** **if**(operator.equals("-")){  result=one-two;  }**else** **if**(operator.equals("\*")){  result=one\*two;  }**else** **if**(operator.equals("/")){  result=one/two;  }  }**catch**(Exception e){  System.***out***.println(e);  }  }  } |

2、为登录过程编写一个JavaBean,要求如下 :（10分）

(1)定义-个包，将该bean编译后生成的类存入该包中。

(2)设计两个属性name和pass

(3)设计访问属性的相应方法。

|  |
| --- |
| package mybean;  public class TestBean {    private String name = null;  private String pass = null;  public TestBean(){}    public void setName(String value){  name = value;  }    public void setPass(String value){  pass = value;  }  public String getname{  return name;  }  public String getpass{  return pass;  }  } |

3、请设计一个用户账户的bean要求如下:

1. 将该bean存放一个独立的包下
2. 该bean下有两个属性:accountName,accountPwd；将属性私有化
3. 提供getter和setter方法

参考答案：

**package** com.etc.bean;

**import** java.io.Serializable;

/\*\*

\* Account bean定义

\* **@author** knowno

\*

\*/

**public** **class** Account **implements** Serializable{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 5659939915458771174L;

**private** String accountName;

**private** String accountPwd;

**public** String getAccountName() {

**return** accountName;

}

**public** **void** setAccountName(String accountName) {

**this**.accountName = accountName;

}

**public** String getAccountPwd() {

**return** accountPwd;

}

**public** **void** setAccountPwd(String accountPwd) {

**this**.accountPwd = accountPwd;

}

}

4、代码填写题,请填写代码片段：

package com.etc.bean;

public class PersonBean **implements Serializable**{

private String firstName;

private String lastName;

private int age;

**public Person(){}**

public Person(String firstName,String lastName,int age){

this.firstName = firstName;

this.lastName = lastName;

this.age = age;

}

public String getFirstName()

{

return this.firstName;

}

public void setFirstName(String firstName)

{

this.firstName = firstName;

}

public String getLastName()

{

return this.lastName;

}

public void setLastName(String lastName)

{

this.lastName = lastName;

}

public int getAge()

{

return this.age;

}

public void setAge(int age)

{

this.age = age;

}

}

5、代码填写题，请填写代码片段

注意：PersonBean为题目2的PersonBean

<jsp:useBean id="**person**"

class="**com.etc.PersonBean**">

<jsp:setProperty name=" **person** " **property="firstName"**

value="小白"/>

<jsp:setProperty name=" **person** " **property="lastName"**

value="张"/>

<jsp:setProperty name=" **person** " **property="age"**

value="10"/>

</jsp:useBean>

<p>名字:

<jsp:getProperty name=" **person** " **property="firstName"**/>

</p>

<p>姓:

<jsp:getProperty name=" **person** " **property="lastName"**/>

</p>

<p>年龄:

<jsp:getProperty name=" **person** " **property="age"**/>

</p>

6、编写一个用于封装账户数据且格式标准的JavaBean类。（10分）

已知账户属性：账号、密码、性别、开户日期、余额

|  |
| --- |
| public class Acount implements Serializable {  private String accountNumber;  ……  public void setAccountNumber(String accountNumber) {  this.accountNumber = accountNumber;  }  Public String getAccountNumber() {  return accountNumber;  }  ……  } |

7、编写一个简单的JavaBean，通过EL的存取运算符访问其属性。

|  |
| --- |
| 文件userLogin.java定义了一个用户Bean，包含姓名、密码和电子邮件3个属性，其代码如下：  package bean;  public class userLogin {  private String userName;  private String password   ;  private String email;  public userLogin() {  userName = "李平";  password = "123";  email = "liping@sohu.com";  }  public void setuserName(String userName) {  this.userName = userName;  }  public String getuserName() {  return this.userName;  }  public void setpassword(String password) {  this.password = password;  }  public String getpassword() {  return this.password;  }  public void setemail(String email) {  this.email = email;  }  public String getemail() {  return this.email;  }  } |
| 文件accessBean.jsp使用EL的存取运算符获取JavaBean的属性，其代码如下：  <%@ page contentType="text/html; charset=gb2312"%>  （2）<%@ page import ="bean.userLogin" %>  <jsp:useBean id="user" class="bean.userLogin" scope="session">  </jsp:useBean>  <html>  <head>  <title>使用存取运算符访问JavaBean的属性</title>  </head>  <body>  <!--通过EL存取运算符访问JavaBean的属性-->  用户名：(3) ${user.username}  <br>  密码：(4)${user.password}  <br>                  电子邮箱：(5)  ${user.email}                  <br>           </body>  </html> |

8、已知JavaBean类Product类其属性name、desc、quantity、price及符合规范的getters/setters方法：

（1）、在onsale.jsp页面中，把addProduct.jsp中表单信息封装到一个JavaBean对象product中，并把请求转发到showdetail.jsp页面。使用到JavaBean标准动作

（2）、showdetail.jsp页面中，把product对象的属性进行显示，使用到标准动作。

（10分）

|  |
| --- |
| 1、<jsp:useBean id=*"product" class="com.chinasofti.ch11.vo.Product" scope="request">*  </jsp:useBean>  <jsp:setProperty name=*"product" property="name" param="name" />*  <jsp:setProperty name=*"product" property="desc" param=“desc" />*  <jsp:setProperty name=*"product" property="quantity" param="quantity" />*  <jsp:setProperty name=*"product" property="price" param="price" />*  <jsp:forward page=*"showdetail.jsp"></jsp:forward>*  2、<jsp:useBean id="product" class="com.chinasofti.ch11.vo.Product" scope="request">  </jsp:useBean>  上架商品信息：<br>  商品名称：<jsp:getProperty name="product" property="name"/><br>  商品描述：<jsp:getProperty name="product" property="desc"/><br>  商品数量：<jsp:getProperty name="product" property="quantity"/><br>  商品价格：<jsp:getProperty name="product" property="price"/><br> |

9、编写一个javabean 属性如下:用户名 和 密码

public class User {

private String username;

private String password;

public User() {

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

10、编写一个简单的JavaBean程序。

|  |
| --- |
| import java.io.\*;  public class SampleBean1{  private String str;  public SampleBean1(){  };  public void setStr(String str) {  this.str = str;  }  public String getStr(){  return str;  }  }    sample1.jsp  <%@ page language="java" contentType="text/html; charset=gb2312"%>  <jsp:useBean id="splBean" class="ch6.SampleBean1"/>  <html>  <head>  <title>一个简单的JavaBean程序</title>  </head>  <body>  <center>  <%  splBean.setStr("这是我的第一个JavaBean程序");  %>  <%=splBean.getStr()%>  </center>  </body>  </html> |