Chapter6测试题：12题(单选题)

1. 指出可用于针对在可嵌入容器中执行的会话bean的引用检索其命名上下的EJB容器类的方法（）。

Indicate the method that can be used to retrieve its named EJB container class for references to session beans that can be executed in an embeddable container

A: getContext()

B: getNamingContext()

C: getContectName()

D: getLookupContextName()

1. 核心JWebUnit API取决于包含\_\_\_\_\_类各种方法的Junit API。

Core JWebUnit API depends on the Junit API that contains the various methods of the \_\_\_\_\_ class

A: Assert

B: Assertions

C: Assess

D: AsserUnit

1. Sam正在为WEB应用程序开发登录页面，登录页面包含两个字段：username，password

当用户分别输入sampleuser和password123作为用户名和密码时，并单击submit按钮时，将显示欢迎页面，他希望通过JWebUnit类验证此功能，以下哪个选项将满足他的要求？

Sam is developing a login page for the WEB application. The login page contains two fields: username, password. When the user enters sampleuser and password123 as the user name and password, and when the submit button is clicked, the welcome page is displayed. He wants to verify this functionality through the JWebUnit class, which of the following options will satisfy his requirements

A: **@Test**

**public void testLoginSuccess(){**

**JWebUnit.beginAt("/index.html");**

**JWebUnit.assertTitleEquals("login");**

**JWebUnit.setTextField("username", "sampleuser");**

**JWebUnit.setTextField("password ", " password123");**

**JWebUnit.submit();**

**JWebUnit.assertTitleEquals("welcome");**

**}**

B: **@Test**

**public void testLoginSuccess(){**

**JWebUnit.beginAt("/index.html");**

**JWebUnit.assertTitleEquals("login");**

**JWebUnit.setTextField("use", "sampleuser");**

**JWebUnit.setTextField("pass ", " password123");**

**JWebUnit.assertTitleEquals("welcome");**

**}**

C: **@Test**

**public void testLoginSuccess(){**

**JWebUnit.assertTitleEquals("login");**

**JWebUnit.setTextField("username", "sampleuser");**

**JWebUnit.setTextField("password ", " password123");**

**JWebUnit.submit();**

**JWebUnit.assertTitleEquals("Error");**

**}**

D: **@Test**

**public void testLoginSuccess(){**

**JWebUnit.beginAt("/index.html");**

**JWebUnit.assertTitleEquals("login");**

**JWebUnit.setTextField("username", "sampleuser");**

**JWebUnit.setTextField("password ", " password123");**

**JWebUnit.assertTitleEquals("Error");**

**}**

1. Sam开发了EJB模块以将距离转化为不同单位，例如米，千米和英里。现在他需要测试该模块。为此，Sam在pom.xml文件中编写了以下代码段以添加服务器属性:

Sam developed the EJB module to convert the distance into different units, such as meters, kilometers and miles. Now he needs to test the module. To this end, Sam wrote the following code snippet in the pom.xml file to add the server properties

<properties>

<endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<glassfish.embedded-static-shell.jar>

D:\Program Files\glassfish-4.1.1\glassfish\lib\embedded\glassfish-embedded-static-shell.jar

</glassfish.embedded-static-shell.jar>

</properties>

现在他需要添加服务器依赖。Sam应该使用以下哪个代码段实现以上需求？( )

Now he needs to add server dependencies. Sam should use the following code snippet to achieve the above requirements

**A:**

<dependency>

<dependencies>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version>

<scope>system</scope>

<systemPath>${glassfish.embedded-static-shell.jar}</systemPath>

</dependencies>

</dependency>

**B:**

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version>

<scope>system</scope>

<systemPath>${glassfish.embedded-static-shell.jar}</systemPath>

</dependency>

**C:**

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version>

<scope>system</scope>

</dependency>

**D:**

<dependency>

<groupId>org.glassfish.extras</groupId>

<artifactId>glassfish-embedded-static-shell</artifactId>

<version>4.1</version>

<scope>system</scope>

<systemPath>${glassfish.jar}</systemPath>

</dependency>

1. Taylor创建了EJB模块以帮助您存储详细联系方式，例如名字和联系电话。他使用以下代码段创建了实

体Bean PhoneBook;

Taylor created the EJB module to help you store detailed contact information such as name and contact phone. He created the entity Bean PhoneBook using the following code snippet

public class PhoneBook implements Serializable{

private static final long serializableVersionUID=-1L;

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private String name;

private String number;

public PhoneBook(){

}

public PhoneBook(String name){

this.name=name;

}

public PhoneBook(String name,String number){

this.name=name;

this.number=number;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

}

………………………………..

………………………………..

接着，他创建了会话bean以定义会话，这些方法可持久保存详细联系方式并检索存储详细信息，如以下代码段所示：

Next, he creates a session bean to define the session, which preserves the detailed contact details and retrieves the store details, as shown in the following code snippet

public class MySessionBean {

@PersistenceContext(unitName = "com.com.niit.entity")

private EntityManager em;

public void persist(Object object){

em.persist(object);

}

public int verify(){

Query q=em.createNamedQuery("PhoneBook.findAll");

Collection entities=q.getResultList();

int s=entities.size();

for (Object o : entities) {

PhoneBook se=(PhoneBook)o;

System.out.println("Found:"+se.getName()+" "+se.getNumber());

}

return s;

}

public void insert(PhoneBook pb){

PhoneBook e=new PhoneBook(pb.getName(),pb.getNumber());

em.persist(e);

}

}

现在他需要测试以上会话bean.使用以下哪个选项将满足他的需求？

Now he needs to test the above session bean. Which of the following options will be used to meet his needs

**A:**

public class MySessionBeanTest {

public MySessionBeanTest() {

}

………………………………………………

…………………………………….

@Test

public void testVerify() throws Exception{

EJBContainer container=javax.ejb.embeddable.EJBContainer.createEJBContainer();

MySessionBean instance=

(MySessionBean)container.lookup("java:global/classes/MySessionBean");

PhoneBook phb=new PhoneBook("Androw","9192939495");

instance.insert(phb);

Assert.assertTrue(res==1);

}

}

**B:**

public class MySessionBeanTest {

public MySessionBeanTest() {

}

@Test

public void testVerify() throws Exception{

EJBContainer container=javax.ejb.embeddable.EJBContainer.createEJBContainer();

MySessionBean instance=

(MySessionBean)container.getContext ("java:global/classes/MySessionBean");

PhoneBook phb=new PhoneBook("Androw","9192939495");

instance.insert(phb);

Assert.assertTrue(res==1);

}

}

**C:**

public class MySessionBeanTest {

public MySessionBeanTest() {

}

@Test

public void testVerify() throws Exception{

EJBContainer container=javax.ejb.embeddable.EJBContainer.createEJBContainer();

MySessionBean instance=

(MySessionBean)container.get().lookup("java:global/classes/MySessionBean");

PhoneBook phb=new PhoneBook("Androw","9192939495");

instance.insert(phb);

Assert.assertTrue(res==1);

}

}

**D:**

public class MySessionBeanTest {

public MySessionBeanTest() {

}

@Test

public void testVerify() throws Exception{

EJBContainer container=javax.ejb.embeddable.EJBContainer.createEJBContainer();

MySessionBean instance=

(MySessionBean)container.getContext().lookup("java:global/classes/MySessionBean");

PhoneBook phb=new PhoneBook("Androw","9192939495");

instance.insert(phb);

Assert.assertTrue(res==1);

}

}

1. 使用静态导入的时候JwebUnit类的常用静态方法中用于测试带有给定文本的按钮是否存在于当前窗口：

When using static import, the JwebUnit class uses a static method for testing whether a button with a given text exists in the current window

A: assertButtonPresent(String buttonId)

B: assertButtonPresentWithText(String text)

C: assertElementPresent(String anId)

D: clickButton(String button)

7,指出可用与模拟具有指定ID的按钮的Click事件的方法。

Indicates the method of using the Click event with the button that simulates a button with the specified ID

A: clickButtonWithID(String buttonId)

B: clickButtonWithElement(String buttonId)

C:clickButton(String buttonId)

D:clickButtonElement(String buttonId)

8,指出可用于创建和初始化可嵌入EJB容器的方法。

Indicates how you can use to create and initialize methods that can be embedded in an EJB container

A: createContainer()

B: createEJBContainer()

C: createEmbeddableContainer()

D: createEmbeddableEJBContainer()

9,指出可用于设置与WEB应用程序的对话开始处的URL的方法

Indicates the method that can be used to set the URL at the beginning of the conversation with the web application

A: beginAt(String aRelativeURL)

B: beginning(Sting aRelativeURL)

C: beginningAt(String aRelativeURL)

D: beginsAt(String aRelativeURL)

10,指出可用于模拟具有指定按钮的Click事件的当前表单的提交的方法。

Indicates a method that can be used to simulate the submission of a current form with a Click event for a specified button

A: submitForm(String buttonName)

B: submitClick(String buttonName)

C: formSubmit(String buttonName)

D: submit(String buttonName)

11,\_\_\_\_\_\_\_\_标记表示将用于测试目的的JWebUnit 框架API的版本。

The \_\_ flag indicates the version of the JWebUnit framework API that will be used for testing purposes

A: <dependencies>

B:<artifactId>

C:<version>

D:<scop>

12,\_\_\_\_\_\_\_\_用于检索命名上下文以查询正在可嵌入式容器中执行的会话bean的引用。

\_\_\_\_\_\_\_ Used to retrieve a naming context to query a reference to a session bean that is being executed in an embedded container

A: getcontext(String parameter)

B: getContext()

C: createEJBContainer()

D: getConnection()