Vendor: Oracle

Exam Code:1Z0-895

Exam Name: Java EE 6 Enterprise JavaBeans Developer Certified Expert

Version: Demo

QUESTION 1

Which two are true about the client view of a message-driven bean? (Choose two.)

A. References to message destinations can be infected.

B. References to message destinations cannot be looked up in the client\\'s JNDI namespace.

C. Clients of a message destination need to know that the destination is listened to by a pool of message consumers,

D. Clients of a message destination do NOT need to know that the destination is listened to by a message-driven bean.

Correct Answer: BC

Client components do not locate message-driven beans and invoke methods directly on them. Instead, a client accesses a message-driven bean through, for example, JMS by sending messages to the message destination for which the

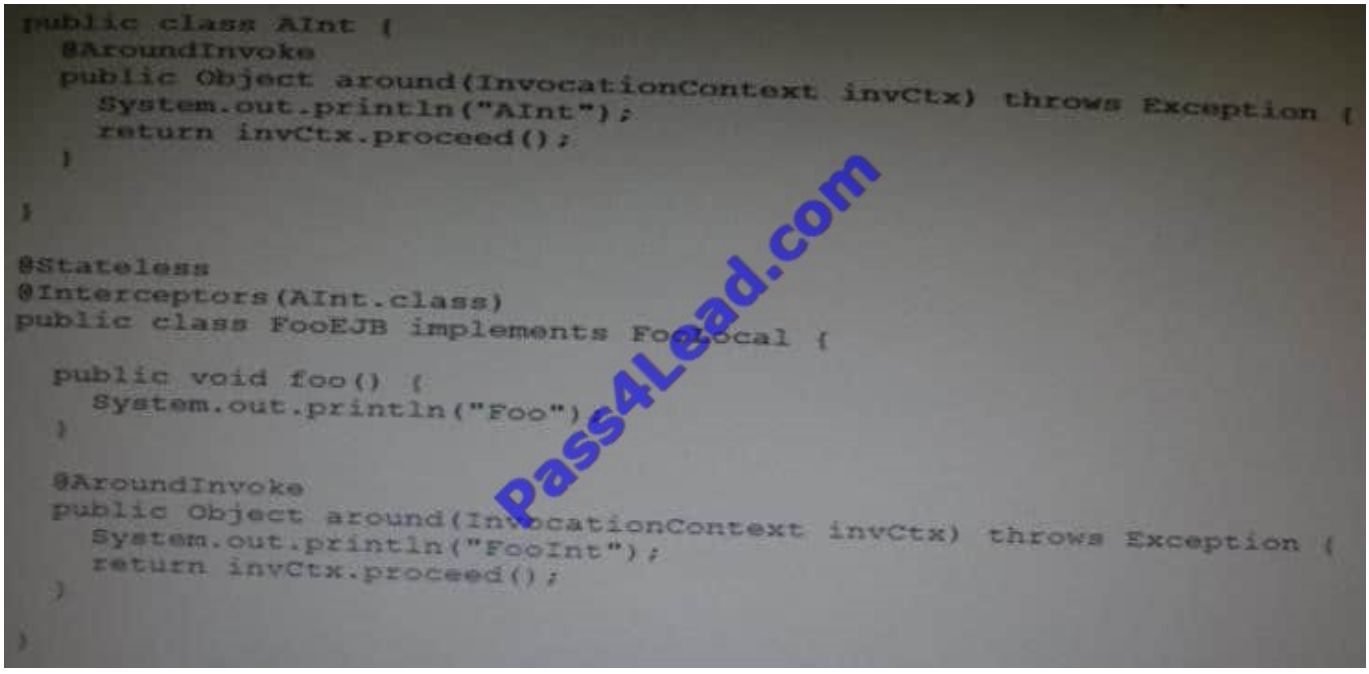
message-driven bean class is the MessageListener.

Incorrect answers:

Reference: The Java EE 6 Tutorial, What is a Message-Driven Bean?

QUESTION 2

A developer writes an interceptor class and a stateless session bean: A client acquires an EJB reference to the FooLocal business interface and invokes the foo() method one time. Which describes the output?



A. Foo FooInt AInt

B. AInt Foo

C. AInt FooInt Foo

D. FooInt AInt Foo

Correct Answer: C

*

At the end of the chain of interceptors, the actual bean method gets called.

*

Interceptors can be bound in three different ways:

Default Class level Method level

In this question both class level and method level interceptors are used.

The class level interceptor intercepts before the method-level interceptor.

Note:

*

Interceptors are used in conjunction with Java EE managed classes to allow developers to invoke interceptor methods on an associated target class, in conjunction with method invocations or lifecycle events. Common uses of interceptors

are logging, auditing, and profiling.

*

An interceptor can be defined within a target class as an interceptor method, or in an associated class called an interceptor class. Interceptor classes contain methods that are invoked in conjunction with the methods or lifecycle events of the

target class.

Interceptor classes and methods are defined using metadata annotations, or in the deployment descriptor of the application containing the interceptors and target classes.

*

javax.interceptor.AroundInvoke

Designates the method as an interceptor method.

*

The target class can have any number of interceptor classes associated with it. The order in which the interceptor classes are invoked is determined by the order in which the interceptor classes are defined in the

javax.interceptor.Interceptors annotation.

Reference: Introduction to EJB3 Interceptors

Reference: The Java EE 6 Tutorial, Overview of Interceptors

QUESTION 3

A developer examines a list of potential enterprise applications and selects the most appropriate technologies to use for each application. For which two applications is EJB an appropriate solution? (Choose two.)

A. To render a GUI for mobile clients.

B. As a container for web-tier components including JSP.

C. As a Web service endpoint accessed by non-Java clients.

D. To receive and respond to HTTP Post requests directly from a web browser.

E. As an online shopping cart which can persist across multiple sessions with a single client.

Correct Answer: CE

QUESTION 4

A developer writes a stateless session bean with one local business interface and with container- managed transactions. All business methods have transaction attribute REQUIRED. The bean has an injected field sessionCtx of the type SessionContext. Which two operations are allowed in a business method of the bean? (Choose two.)

A. sessionCtx. getEJBObject

B. sessionCtx.setRollbackOnly

C. sessionCtx. getMessageContext

D. sessionCtx. getBusinessObject

E. sessionCtx. getEJBLocalObject

Correct Answer: BD

QUESTION 5

Which two annotations can be applied at the class, method, and field levels? (Choose two.)

A. @EJB

B. @Init

C. @Resource

D. @RolesAllowed

E. @PostActivate

Correct Answer: AC

A: javax.ejb.EJB Description

Target: Class, Method, Field

Specifies a dependency or reference to an EJB business or home interface.

You annotate a bean\\'s instance variable with the @EJB annotation to specify a dependence on another EJB.

C: javax.annotation.Resource Description

Target: Class, Method, Field

Specifies a dependence on an external resource, such as a JDBC data source or a JMS destination or connection factory.

Incorrect:

B: javax.ejb.Init Description Target: Method

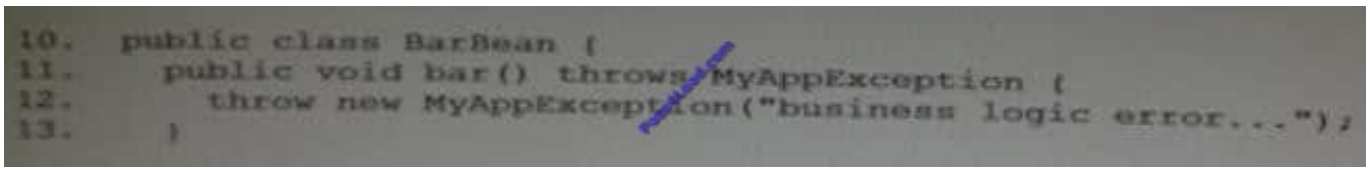
D: javax.annotation.security.RolesAllowed Description Target: Class, Method Specifies the list of security roles that are allowed to access methods in the EJB.

E: javax.ejb.PostActivate Description Target: Method Specifies the lifecycle callback method that signals that the EJB container has just reactivated the bean instance.

Reference: EJB 3.0 Metadata Annotations Reference

QUESTION 6

FooBean and BarBean are both EJB 3.x stateless session beans with bean-managed transaction demarcation. The business method foo in FooBean starts a UserTransaction and invokes the business method bar in BrBean. Given:



What is the expected result of this method invocation assuming control reaches Line 12?

A. FooBean.foo method receives MyAppException.

B. The container discards the BarBean bean instance.

C. FooBean.foo method receives a javax.ejb.EJBException that wraps MyAppException.

D. FooBean.foo method receives javax.transaction.TransactionRolledbackException.

Correct Answer: D

The transaction will roll back. Note:

* In bean-managed transaction demarcation, the code in the session or message-driven bean explicitly marks the

boundaries of the transaction. Although beans with container-managed transactions require less coding, they have one limitation: When a method is executing, it can be associated with either a single transaction or no transaction at all. If this limitation will make coding your bean difficult, you should consider using bean-managed transactions.

Reference: The Java EE 5 Tutorial, Bean-Managed Transactions

QUESTION 7

Which is a valid Postconstruct method in a message-driven bean class?

A. @PostConstruct public boolean init() { return true; }

B. @PostConstruct private static void init () {}

C. @PostConstruct private void init ()

D. @PostConstruct public static void init () {}

Correct Answer: C

QUESTION 8

A developer wants to create a JMS message-driven bean that responds to javax.jms.TextMessage messages. Which two statements are true? (Choose two.)

A. The developer must implement the ejbCreate method.

B. The developer does NOT need to create a business interface for the bean.

C. The developer must implement a method that declares javax.jms.TextMessage as an argument.

D. The message-driven bean class must implement methods of the javax.jms.TextMessageListener interface.

E. The message-driven bean class must implement methods of the javax.ejb.MessageDrivenBean interface.

Correct Answer: BD

B: * Client components do not locate message-driven beans and invoke methods directly on them. Instead, a client accesses a message-driven bean through, for example, JMS by sending messages to the message destination for which the

message-driven bean class is the MessageListener.

D: * A message-driven bean is an enterprise bean that allows Java EE applications to process messages asynchronously. This type of bean normally acts as a JMS message listener, which is similar to an event listener but receives JMS

messages instead of events.

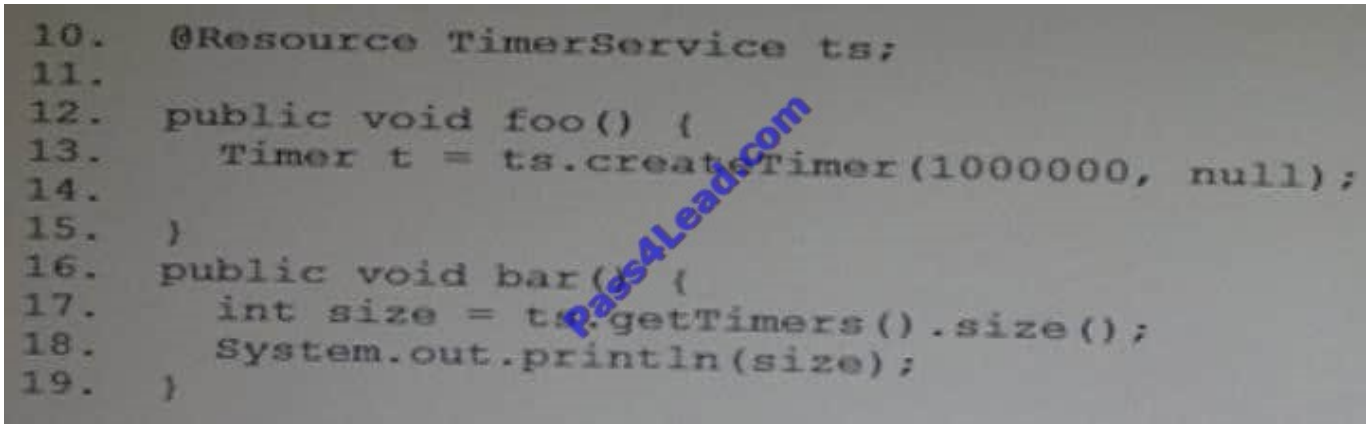
* In a fashion similar to a Message-Driven Bean (MDB) in the EJB world, the Message-Driven POJO (MDP) acts as a receiver for JMS messages. The one restriction (but see also below for the discussion of the MessageListenerAdapter

class) on an MDP is that it must implement the javax.jms.MessageListener interface.

Reference: The Java EE 6 Tutorial, What Is a Message-Driven Bean?

QUESTION 9

A developer implements a stateless session bean as a timed object. The bean contains two local business methods with the transaction attribute REQUIRED.



A client begins a UserTransaction and calls the foo local business method. The foo method returns five seconds later. The client rolls back the transaction and then calls the bar local business method. Assuming there have been no other client invocations on the stateless session bean, what is the value of the size variable when control reaches Line 18?

A. 0

B. 1

C. -1

Correct Answer: A

An enterprise bean usually creates a timer within a transaction. If this transaction is rolled back, the timer creation also is rolled back. The transaction will roll back. The creation of the timer will rock back. There will be zero timers.

Note: The UserTransaction interface defines the methods that allow an application to explicitly manage transaction boundaries. rollback() rolls back the transaction associated with the current thread.

Note 2: size()

Returns the number of elements in this collection. If this collection contains more than Integer.MAX_VALUE elements, returns Integer.MAX_VALUE.

Reference: The Java EE 6Tutorial. PartNo: 821 Transactions andTimers

---

QUESTION 10

Which is a correct way to define a runtime exception as an EJB 3.x application exception?

A. public class MyAppException extends javax.ejb.EJBException

B. @ApplicationException public class MyAppException extends javax.ejb.EJBException

C. public class MyAppException extends javax.lang.EJBException

D. @ApplicationException public class MyAppException extends javax.lang.EJBException

Correct Answer: B

Use the @javax.ejb.ApplicationException annotation to specify that an exception class is an application exception thrown by a business method of the EJB. The EJB container reports the exception directly to the client in the event of the application error.

Note: java.lang.Object java.lang.Throwable java.lang.Exception java.lang.RuntimeException javax.ejb.EJBException javax.ejb public class EJBException extends java.lang.RuntimeException The EJBException is thrown to report that the invoked business method or callback method could not be completed because of an unexpected error (e.g. the instance failed to open a database connection).

Example:

The following ProcessingException.java file shows how to use the @ApplicationException annotation to specify that an exception class is an application exception thrown by one of the business methods of the EJB:

```
package examples;

import javax.ejb.ApplicationException; /**

*

 Application exception class thrown when there was a processing error

*

 with a business method of the EJB. Annotated with the

*

 @ApplicationException annotation. */

@ApplicationException()

public class ProcessingException extends Exception {
```

Reference: Programming WebLogic Enterprise JavaBeans, Version 3.0 programming Application Exceptions

---

QUESTION 11

A developer impalements an asynchronous implementation for calculating insurance proposals. The input data for the calculations is made available on a single message queue. Two types of insurance proposals will be calculated: car and

life. Message with data for other insurance types are posted on the queue but should be left on the queue by this implementation.

Which statement is true?

A. The developer will NOT succeed because all messages will be consumed from the queue.

B. The developer can implement a push-back mechanism if the message is of the wrong type.

C. The developer can use a messageSelector to receive only the car and life data message if the JMS body contains selectable data.

D. The developer can use a messageSelector to receive only the car and life data message if the header contains properties to make selection.

Correct Answer: D

A JMS message selector allows a client to specify, by header field references and property references, the messages it is interested in. Only messages whose header and property values match the selector are delivered. What it means for a message not to be delivered depends on the MessageConsumer being used (see QueueReceiver and TopicSubscriber). Reference: javax.jms Interface Message

---

QUESTION 12

Which statement is correct about a Java EF client of a message driven bean?

A. The client can use JNDI to obtain a reference to a message destination.

B. The client can use JNDI to obtain a reference to a dependency injection.

C. The client can use JNDI to obtain a reference to a message-driven bean instance.

D. The client can use JNDI to look up a reference to the message-driven bean\\'s home interface.

Correct Answer: A

Try our product !

100% Guaranteed Success

100% Money Back Guarantee

365 Days Free Update

Instant Download After Purchase

24x7 Customer Support

Average 99.9% Success Rate

More than 800,000 Satisfied Customers Worldwide

Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: