1Z0-895$^{\text{Q\&As}}$

Java EE 6 Enterprise JavaBeans Developer Certified Expert

Pass Oracle 1Z0-895 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/1Z0-895.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle Official Exam Center

QUESTION 1

While excepting a business method in a stateless session bean the container rolls back the method\\'s transaction. Which three are possible causes for the container\\'s behavior? (Choose three.)

A. The bean uses container-managed transactions and invokes EJBContext.setRollbackOnly.

B. The bean uses container-managed transactions and invokes EJBContext.getRollbackOnly.

C. The business method throws a java.lang.NullPointerException.

D. The business method throws a checked exception of a class type that is marked with the @ApplicationException annotation with the rollback element value true.

E. The business method throws a unchecked exception of a class type that is marked with the @ApplicationException annotation with the rollback element value true.

F. The bean uses container-managed transactions and throws a checked exception of a class type that is marked with the @ApplicationException annotation with the rollback element value false.

Correct Answer: ADF

A: setRollbackOnly

Mark the current transaction for rollback. The transaction will become permanently marked for rollback. A transaction marked for rollback can never commit. Only enterprise beans with container-managed transactions are allowed to use this

method.

Note:

*

In a stateless session bean with bean-managed transactions, a business method must commit or roll back a transaction before returning.

*

Bean-Managed Transactions 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.

*

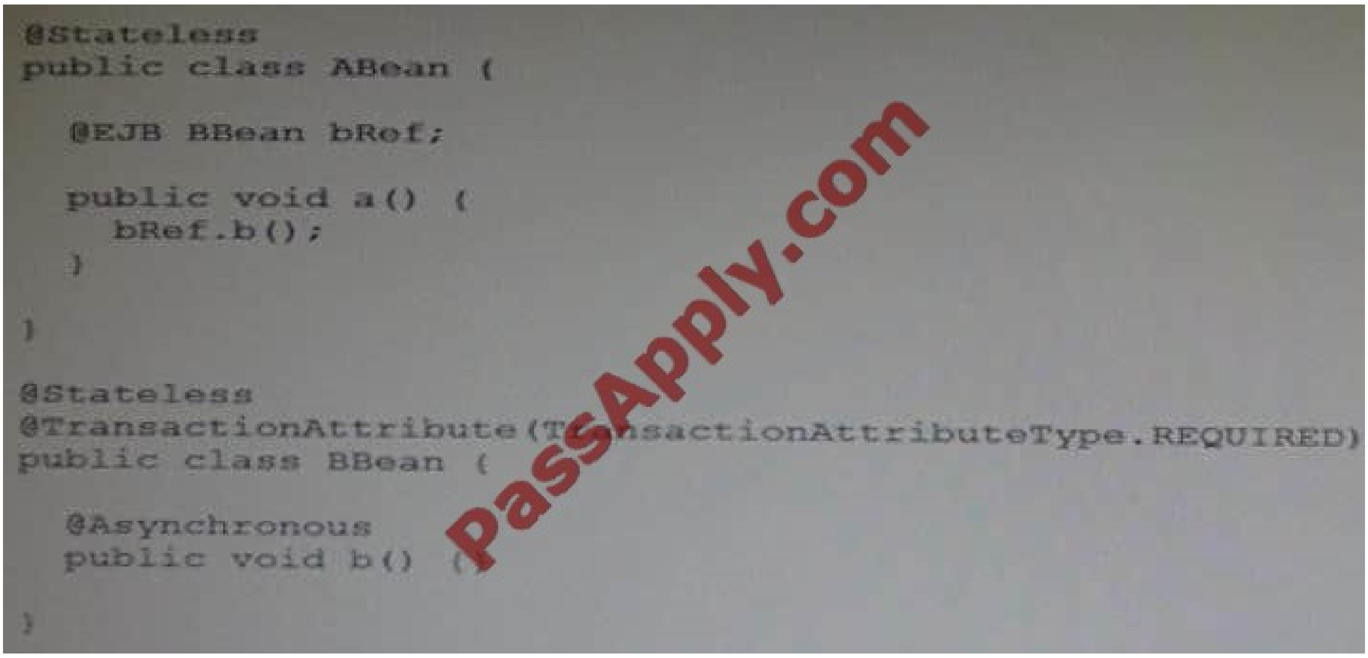
(incorrect) Unchecked runtime exceptions represent conditions that, generally speaking, reflect errors in your program\\'s logic and cannot be reasonably recovered from at run time. Incorrect:

B: getRollbackOnly

Test if the transaction has been marked for rollback only. An enterprise bean instance can use this operation, for example, to test after an exception has been caught, whether it is fruitless to continue computation on behalf of the current transaction. Only enterprise beans with container- managed transactions are allowed to use this method.

QUESTION 2

Given two stateless session beans, ABean and BBean:



```
@Stateless
public class ABean {

   @EJB BBean bRef;

   public void a() {
     bRef.b();
   }

}

@Stateless
@TransactionAttribute(TransactionAttributeType.REQUIRED)
public class BBean {

   @Asynchronous
   public void b() {}

}
```

A client that is not executing within a transaction acquires an EJB reference to ABean and invokes the a() method on time. How many distinct transactions are started by the container after all processing has completed?

A. 0

B. 1

C. 2

D. 3

Correct Answer: B

Note: REQUIRED If a client invokes the enterprise bean\\'s method while the client is associated with a transaction context, the container invokes the enterprise bean\\'s method in the client\\'s transaction context. Reference: Enum TransactionAttributeType

QUESTION 3

A stateful session bean contains a number of instance variables. The types of instance variables A and B are serializable. Instance variable B is a complex type which is populated by many business calls, and can, therefore, not be refilled by the client without starting all over. A helper instance variable C is defined as having a Serializable type, and can hold all the information which is in variable B. for example, B is of type XML-DOM tree and C of Type String.

Which two solutions, when combined, maintain the state of the session bean over a passivation and activation by the container? (Choose two.)

A. The value of helper variable C is used to create the value of Instance variable B in the beans no-arg constructor.

B. The value of helper variable C is used to create the value of instance variable B in a @postcreate annotated method.

C. The value of helper variable C is used to create the value of instance variable B in a @postActivate annotated method.

D. Instance variable A must be made null and instance variable B must be converted to a Serializable type and assigned to another instance variable in a @preDestroy annotated method.

E. Instance variable A must be defined transient. Instance variable B must be converted to a Serializable type, set to null, and assigned to the instance variable C in a @PrePassivate annotated method.

Correct Answer: CE

QUESTION 4

A developer creates a stateless session bean, EmployeeServiceBean, and its interface, EmployeeService. The session bean uses two annotated entity classes, Employee.class and Department.class.

Which two package options can the developer use when creating a deployable EAR? The proposed directory structure is listed for each option. (Choose two)

A. Emp.ear emp-ejb.jar META-INF/persistence.xml EmployeeService.class EmployeeServiceBean.class lib/emp-classes.jar Employee.class Department.class

B. Emp.ear META-INF/orm.xml EmployeeService.class EmployeeServiceBean.class Employee.class Department.class

C. Emp.ear emp-ejb.jar META-INF/persistence.cml Employee.class Department.class EmployeeService.class EmployeeServiceBean.class

D. Emp.ear emp-ejb.jar persistence.xml Employee.class Department.class EmployeeService.class EmployeeServiceBean.class

Correct Answer: AC

QUESTION 5

Which two statements are correct about stateless session beans? (Choose two.)

A. The bean class may declare instance variables.

B. The lifetime of the bean instance is controlled by the client.

C. The container may use the same bean instance to handle multiple business method invocations at the same time.

D. The container may use the same bean instance to handle business method invocations requested by different clients, but not concurrently.

Correct Answer: AC

* A: Stateless session beans are EJB\\'s version of the traditional transaction processing applications, which are executed using a procedure call. The procedure executes from beginning to end and then returns the result. Once the

procedure is done, nothing about the data that was manipulated or the details of the request are remembered. There is no state.

These restrictions don\\'t mean that a stateless session bean can\\'t have instance variables and therefore some kind of internal state. There\\'s nothing that prevents you from keeping a variable that tracks the number of times a bean has been called or that tracks data for debugging. An instance variable can even hold a reference to a live resource like a URL connection for writing debugging data, verifying credit cards, or anything else that might be useful.

A stateless session bean is relatively easy to develop and also very efficient.

C:

Stateless session beans require few server resources because they are neither persistent nor dedicated to one client. Because they aren\\'t dedicated to one client, many EJB objects can use just a few instances of a stateless bean. A

stateless session bean does not maintain conversational state relative to the EJB object it is servicing, so it can be swapped freely between EJB objects. As soon as a stateless instance services a method invocation, it can be swapped to

another EJB object immediately. Because there is no conversational state, a stateless session bean doesn\\'t require passivation or activation, further reducing the overhead of swapping. In short, they are lightweight and fast!

* The Lifecycle of a Stateless Session Bean Because a stateless session bean is never passivated, its lifecycle has only two stages: nonexistent and ready for the invocation of business methods. The EJB container typically creates and maintains a pool of stateless session beans, beginning the stateless session bean\\'s lifecycle. The container performs any dependency injection and then invokes the method annotated @PostConstruct, if it exists. The bean is now ready to have its business methods invoked by a client.

At the end of the lifecycle, the EJB container calls the method annotated @PreDestroy, if it exists (not B). The bean\\'s instance is then ready for garbage collection.

Try our product !

Need Help