Oracle

Exam 1z0-898

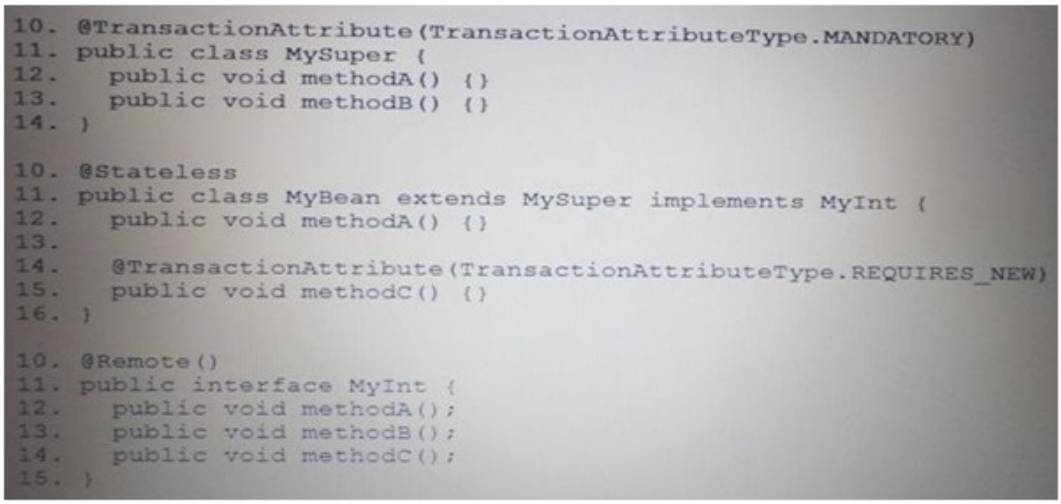
Java EE 6 Java Persistence API Developer Certified Expert Exam

Version: 8.0

[Total Questions: 33]

Question No : 1

Given the following stateless session bean implementation classes:



```
10. @TransactionAttribute(TransactionAttributeType.MANDATORY)
11. public class MySuper {
12.    public void methodA() {}
13.    public void methodB() {}
14. }

10. @Stateless
11. public class MyBean extends MySuper implements MyInt {
12.    public void methodA() {}
13.
14.    @TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)
15.    public void methodC() {}
16. }

10. @Remote()
11. public interface MyInt {
12.    public void methodA();
13.    public void methodB();
14.    public void methodC();
15. }
```

Assuming no other transaction-related metadata, what are the transaction attributes on methodB, and methodC respectively?

A. MANDATORY, MANDATORY, and MANDATORY
B. REQUIRED, MANDATORY, and REQUIRES_NEW
C. MANDATORY, MANDATORY, <and REQUIRES__NEW
D. REQUIRED, REQUIRES_NEW, and REQUIRES_NEW

Answer: B

Question No : 2

A developer wrote an entity class with the following method:

Private static Logger logger = Logger.getLogger (“myLogger”);

@PrePersist

@PreUpdate

Public void doA () {

Logger.info (“A”);

```
}

@PostPersist

@PostUpdate

Public void doB () {

logger.info (“B”);

}
```

What will the log message contain when an application does the following?

1. Begins a transaction
2. Creates the entity
3. Persists the entity
4. Commits the transaction
5. Begins the entity data
6. Modifies the entity data
7. Merges the entity
8. Commits the second transaction

A. A
A
B
B
B. A
B
A
B
C. A
B
B
A
B
D. The application will throw an exception because multiple lifecycle callback annotations applied to a single method.

Answer: B

Question No : 3

An application uses an application-managed entity manager. Which of the following is NOT true?

A. The application may specify whether the scope of the persistence context is extended.
B. The application must use EntityManagerFactory instances to create entity managers.
C. Entity manager instances must be explicitly closed.
D. The application may need to call EntityManager. joinTransaction If a JTA aware entity manager is used.

Answer: A

Question No : 4

A session bean business method throws an exception during execution.

Which two are responsibilities of the Bean Provider when throwing the exception? (Choose two.)

A. For application exceptions, ensure that if the current transaction commits there will be no loss of data integrity.
B. For application exceptions, ensure that the current transaction will commit.
C. For system errors, when the client is remote through a java.rmi.remoteException that wraps the original exception.
D. For checked exceptions from which the bean cannot recover, throw an EJBException that wraps the original exception.

Answer: A,D
Reference:http://java.boot.by/scbcd5-guide/ch10s02.html

Question No : 5

The developer wants to define a unidirectional relationship from the customer entity to the

order entity and map this relationship using a foreign key mapping strategy.

Which one of the pairs of classes below correctly achieves this task?

A. @Entity public class Customer {
@Id int customerId;
@OneToMany @JoinColumn (name = “CUST_ID”) Set <Order> orders;
. . .
}
@Entity public class order {
@Id int orderId;
. . .
}
B. @Entity public class Customer {
@Id int customerId;
@OneToMany Set <Order> orders;
. . .
}
@Entity
@JoinColumn (names = “CUST-ID”, referencedColumnName = “customerId”)
public class order {
@Id int order Id;
. . .
}
C. @Entity public class Customer {
@Id int customerId;
@OneToMany (JoinColumn = @joinColumn (name = “CUST_ID”) Set <Order> orders;
. . .
}
@Entity public class order {
@Id int orderId;
. . .
}
D. @ Entity public class Customer {
@Id int customerId;
@OneToMany (JoinColumn = @JoinColumn (name = “CUST_ID”), table = “”ORDER) Set <Order> orders;
. . .
}
@Entity public class order {
@Id int orderId;
. . .
}

Answer: A

Question No : 6

A developer is creating an entity which is mapped to a table that has a primary key constraint defined on two character columns and would like to use mapping defaults as much as possible to simplify the code.

Which two mapping options can be chosen? (Choose two.)

A. Use an @id property that constructs a private field as a concatenation of two columns.
B. Use a separate class to map those two columns and use an @idclass annotation to denote I primary key field or property in the entity.
C. Use a separate @Embeddable class to map those two columns and use an @EmbeddedId annotation to denote a single primary key field or property in the entity.
D. Use a separate @Embeddable class to map those two column and add two fields or properties the entity, each marked as @id, that correspond to the fields or properties in the embeddable class.
E. Use a separate class to map those two columns. Specify that class using @Idclass annotation on the entity class. Add two fields or properties to the entity, each marked as @Id, that correspond to the fields or properties in that separate class.

Answer: C,E

Question No : 7

The department entity has a unidirectional OneToMany relationship to the employee entity. The developer wants to model this relationship as a java.util.map such that the key of map is true employee name. The primary key of the Employees entity is empId, an integer.

Which of the following is correct?

A. @OneToMany (targetEntity = Employee.class)
@MapKeyClass (string.class)
map employees;
B. @OneToMany @mapKey (name, “name”) map < integer, Employee> Employees;
C. @OneToMany @MapKeyJoinColumn (name = “name”) map <String, Employee> employees;
D. @OneToMany @mapsId (name = “name”) map <String, Employee> employees;

Answer: B

Question No : 8

The developer wants to override the default mappings for an embeddable class Address used by the customer entity.

The Address class is defined as follows:

@Embeddable public class Address (

private String street;

private String city;

private String country;

. . .

)

Assume that NO mapping descriptor is present. Which code segment below shows the correct way to override the default mapping for address?

A. @AttributeOverrides ({
@AttributeOverride (name = “street”, column = @Column (name = ADDR_STREET)),
@AttributeOverride (name = “city, column = @Column (name = ADDR_CITY)),
@AttributeOverride (name = “country, column = @Column (name = ADDR_COUNTRY)),
}}
@Embedded Address addr;
B. @ AttributeOverrides ({
@AttributeOverride (name = “street”, column = @Column (name = “name_STREET”)),
@AttributeOverride (name = “city, column = @Column (name = “name_CITY”)),
@AttributeOverride (name = “country, column = @Column (name = “name_COUNTRY”)),
}}
@Embedded Address addr;
C. @ AttributeOverrides ({
@AttributeOverride (name = “street”, column (name = “name_STREET”)),
@AttributeOverride (name = “city, column (name = “name_CITY”)),
@AttributeOverride (name = “country, column (name = “name_COUNTRY”)),
}}
@Embedded Address addr;

D. @AttributeOverrides ({
@AttributeOverride (name = “addr.street”, column = @Column (name = ADDR_STREET)),
@AttributeOverride (name = “addr.city”, column = @Column (name = ADDR_CITY)),
@AttributeOverride (name = “addr.country”, column = @Column (name = ADDR_COUNTRY)),
}}
@Embedded Address addr;

Answer: B
Reference:http://docs.oracle.com/javaee/5/api/javax/persistence/AttributeOverrides.html

Question No : 9

Which two of the following statements are true of embeddable classes? (Choose two)

A. An embeddable class must not be used to represent the state of another embeddable class.
B. Null comparison operations over embeddable classes are not supported in the Java Persistence query language.
C. An embeddable class must not contain a relationship to an entity.
D. An embeddable class can be the key of a Map relationship.

Answer: B,D

Question No : 10

Refer to the Exhibit.

```
10. @Entity
11. // insert code here
12. public class Booking {
13.    private Date bookingDate;

And:

15. @Entity
16. public class FlightBooking extends Booking {
17.   private String flightNumber;

And:

20. @Entity
21. public class HotelBooking extends Booking {
22.    private String hotelName;
```

A developer wants to have bookingdata stored in the table BOOKING, flightnumber in table FLIGHTBOOKING, and hotel name in HOTELBOOKING.

Which code, inserted at line 11 of class Booking, is appropriate for his strategy?

A. @Joined
B. @SingleTable
C. @TablePerClass
D. @Inheritance (strategy = JOINED)
E. @Inheritance (strategy = SINGLE_TABLE)
F. @Inheritance (strategy = TABLE_PER_CLASS)

Answer: D

Question No : 11

A stateless session bean's business method invokes EJBContext.setRollBackOnly and receives an IllegalStateException.

Under which of these conditions could this be possible?

A. The business method is marked with the MANDATORY transaction attribute.
B. The business method is marked with the NONSUPPORTED transaction attribute.
C. This Is NOT possible; a stateless session bean cannot invoke EJBContext.SetRollBackOnly.
D. The bean has no metadata (in annotations 01 deployment descriptor) which specifies the transaction attribute for the method.

Answer: B

Question No : 12

Entity lifecycle callback methods may be defined in which three classes? (Choose three)

A. Embedded classes
B. Entity classes
C. Abstract classes
D. Entity listener classes
E. Mapped superclasses
F. Concrete non-entity superclasses

Answer: B,D,E
Reference:http://stackoverflow.com/questions/3747268/how-to-inject-a-springs-service-bean-into-a-jpa-entity(See the answer, first paragraph)

Question No : 13

Given the following code:

```
Public void create () {
try {
doA () {
} catch (PersistenceException e) {}
try (doB) ();
} catch (PersistenceException e) {}
}
```

Calling method doA will cause an NonUniqueResultException to be thrown. Calling method doB will cause an EntityExistsException to be thrown.

What two options describe what will happen when the create method is called within an application ' uses container managed transactions? (Choose two)

A. Method doB will never be called.
B. The current transaction will continue after doA executes.
C. The current transaction will continue after doB executes.
D. The current transaction will be marked for rollback when doA is called.
E. The current transaction will be marked for rollback when doB is called.

Answer: B,E

Question No : 14

A developer has created a deep entity class hierarchy with many polymorphic relationships between entitles. Which inheritance strategy, as defined by the inheritanceType enumerated type, will be most performed in this scenario?

A. Single table-per-class-hierarchy (InheritanceType.SINGLE_TABLE)
B. Joined-subclass (inheritanceType. JOINED)
C. Table-per-concrete-class (inheritanceType.TABLE_PER_CLASS)
D. Polymorphic join table (inheritanceType. POLYMORPHIC_JOIN_TABLE)

Answer: A

Question No : 15

An application that uses container-managed transaction demarcation creates a query within an active transaction and receives a QueryTimeoutException. Which of those scenarios describes what happens to the active transaction?

A. The statement and the transaction continue.
B. The query is recreated within the current transaction.
C. The statement and the transaction are marked for rollback.
D. The statement is rolled back, but the transaction continues.

Answer: D
Reference:http://docs.jboss.org/hibernate/jpa/2.1/api/javax/persistence/QueryTimeoutException.html

Question No : 16

FooBean and BarBean are both EJB 3.0 stateless beans with container-managed transaction demarcation. All business methods in FooBean have transaction attribute REQUIRED, and all business methods in BarBean have transaction attribute REQUIRED_NEW. The business method foo in FooBean invokes the Business method bar in BarBean.

Given:

10. Public class BarBean {

11. public void bar () {

12. throw new RuntimeException (“unexpected error . . . ”);

13. }}

Which is true about the method of invocation assuming execution reaches line 12?

A. FooBean.foo method receives javax.ejb.EJBException.
B. The BarBean bean instance is in ready state for the next invocation.
C. FooBean.foo method receives javax -ejb. EJBTtansactionRolledbackException.
D. FooBean.foo method receives the original RuntimeException thrown from BarBean.bar method.

Answer: A

Question No : 17

Given two entities with many-to-many bidirectional association between them: