KillTest

質量更高　服務更好

學習資料

http://www.killtest.net

一年免費更新服務

Exam : **1Z0-898**

Title : Java EE 6 Java Persistence API Developer Certified

Version : Demo

1.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
Explanation:
Entity Listeners and Callback Methods
A method may be designated as a lifecycle callback method to receive notification of entity lifecycle events. A lifecycle callback method can be defined on an entity class, a mapped superclass, or an entity listener class associated with an entity or mapped superclass.
Reference: How to inject a spring's service bean into a JPA Entity?
http://stackoverflow.com/questions/3747268/how-to-inject-a-springs-service-bean-into-a-jpa-entity

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

3.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
Explanation:
B: PersistenceExceptionis thrown by the persistence provider when a problem occurs. All instances of PersistenceException except for instances of NoResultException, NonUniqueResultException, LockTimeoutException, and QueryTimeoutException will cause the current transaction, if one is active, to be marked for rollback.
E: EntityExistsException is thrown by the persistence provider when EntityManager.persist(Object) is called and the entity already exists. The current transaction, if one is active, will be marked for rollback.
Reference: javax.persistence, Class PersistenceException
Reference: javax.persistence, Class EntityExistsException

4.An application that uses pessimistic locking calls an update Data method that results in a Lock Timeout Exception being thrown.
What three statements are correct? (Choose three)
A. The current transaction continues.
B. The current statement continues.
C. The current transaction is rolled back.

D. The current statement is rolled back.
E. The LockTimeoutException can NOT be caught.
F. The LockTimeoutException can be caught, and the update Data method retried.
Answer: A,D,F
Explanation:
LockTimeoutExceptionis thrown by the persistence provider when an pessimistic locking conflict occurs that does not result in transaction rollback. This exception may be thrown as part of an API call, at, flush or at commit time. The current transaction, if one is active, will be not be marked for rollback.
Reference: javax.persistence, Class LockTimeoutException

5.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
Explanation:
The Single Table per Class Hierarchy Strategy This strategy provides good support for polymorphic relationships between entities and queries that cover the entire entity class hierarchy.
Note: Enum InheritanceType:
*SINGLE_TABLE
A single table per class hierarchy.
*JOINED
A strategy in which fields that are specific to a subclass are mapped to a separate table than the fields that are common to the parent class, and a join is performed to instantiate the subclass.
*TABLE_PER_CLASS
A table per concrete entity class.
Incorrect:
Not D: There is no such thing as inheritanceType.POLYMORPHIC_JOIN_TABLE.
Reference: Entity Inheritance Mapping Strategies
javax.persistence, Enum InheritanceType