Enthuware Mobile Test Studio

Home | Certifications | **Test** | Review | Progress | Notes

Standard Tests - Last Day Test (Unique) : 2019-10-24 19:49

Q 69 of 86 Mark I/O Fundamentals - Serialization enthuware.ocpjp.v8.2.1712

As a part of an application, you have serialized and stored some objects of a class in the database. At another place in the same application, you deserialize those objects.

After a few months you determine that you need to add one new String field in the class.

Which of the following statements are correct regarding the above described situation?

Answered Incorrectly You had to select 1 option(s)

- The objects serialized earlier cannot be deserialized to the updated class objects.
- Objects serialized earlier can be deserialized to the updated class objects by adding a new `serialVersionUID` field with a value of 0 to the updated class.
- No objects serialized earlier can be deserialized to the updated class. Objects serialized earlier will be deserialized to the updated class objects but the newly added field will be null.
- **Objects serialized earlier can be deserialized only if the original class explicitly defined a `serialVersionUID` field and if the updated class maintains the same value for that field.**

  Every class that implements Serializable should explicitly define a static final `serialVersionUID` field of type `long`. For example,
  ```
  public static final long serialVersionUID = 1;
  ```
  If you make changes to the class and if you still want old objects to be successfully deserialized into the updated class objects, you should keep the same value for `serialVersionUID`.
  However, this is not a must.

- **It is possible to deserialize the older objects into the update class objects even if the original class did not explicitly define the `serialVersionUID` field.**

  If a class that implements `Serializable` does not explicitly define `serialVersionUID` field, the compiler automatically adds this field. It assignes this field a value that is computed based on the attributes of the class such as the fields and the implemented interfaces.

  It is possible to determine this value by various means. For example, by using the serialver tool provided by the JDK:
  ```
  serialver MyClass
  ```
  Or by using `ObjectInputStream.readClassDescriptor()` method.

  Once you get this number, you can assign the same number to `serialVersionUID` in your updated class.

Previous | Next | Evaluate | Finish | Review

Add/Edit Note