All Tests (https://rank.sanfoundry.com/spring-tests/)
Top Rankers (https://rank.sanfoundry.com/top-rankers-spring/)
Practice Test 1 (https://rank.sanfoundry.com/spring-practice-test-1/)
Practice Test 2 (https://rank.sanfoundry.com/spring-practice-test-2/)
Practice Test 3 (https://rank.sanfoundry.com/spring-practice-test-3/)
Practice Test 4 (https://rank.sanfoundry.com/spring-practice-test-4/)
Practice Test 5 (https://rank.sanfoundry.com/spring-practice-test-5/)
Practice Test 6 (https://rank.sanfoundry.com/spring-practice-test-6/)
Practice Test 7 (https://rank.sanfoundry.com/spring-practice-test-7/)
Practice Test 8 (https://rank.sanfoundry.com/spring-practice-test-8/)
Practice Test 9 (https://rank.sanfoundry.com/spring-practice-test-9/)
Practice Test 10 (https://rank.sanfoundry.com/spring-practice-test-10/)
Mock Test 1 (https://rank.sanfoundry.com/spring-mock-test-1/)
Mock Test 2 (https://rank.sanfoundry.com/spring-mock-test-2/)
Mock Test 3 (https://rank.sanfoundry.com/spring-mock-test-3/)
Mock Test 4 (https://rank.sanfoundry.com/spring-mock-test-4/)
Mock Test 5 (https://rank.sanfoundry.com/spring-mock-test-5/)
Mock Test 6 (https://rank.sanfoundry.com/spring-mock-test-6/)
Mock Test 7 (https://rank.sanfoundry.com/spring-mock-test-7/)
Mock Test 8 (https://rank.sanfoundry.com/spring-mock-test-8/)
Mock Test 9 (https://rank.sanfoundry.com/spring-mock-test-9/)
Mock Test 10 (https://rank.sanfoundry.com/spring-mock-test-10/)

« Prev Page (https://www.sanfoundry.com/spring-questions-answers-problems-direct-jdbc/)
Next Page (https://www.sanfoundry.com/spring-quiz-questions-answers/) »

Spring Questions and Answers – JDBC Template

This set of Java Spring Multiple Choice Questions & Answers (MCQs) focuses on “JDBC Template”.

1. Class which declares a number of overloaded update() template methods to control the overall update process.
a) org.springframework.jdbc.core.JdbcTemplate
b) org.springframework.jdbc.core.*
c) org.springframework.jdbc.*
d) none of the mentioned
View Answer

Answer: a
Explanation: The org.springframework.jdbc.core.JdbcTemplate class declares a number of overloaded update() template methods to control the overall update process.

2. You implement this interface to override the statement creation task.

a) PreparedStatement

b) PreparedStatementCreator

c) PreparedCreator

d) None of the mentioned

View Answer

Answer: b

Explanation: The first callback interface to introduce is PreparedStatementCreator. You implement this interface to override the statement creation task and the parameter binding task of the overall update process.

3. When implementing the PreparedStatementCreator interface, you will get the database connection as the createPreparedStatement() method's argument.

a) True

b) False

View Answer

Answer: a

Explanation: All you have to do in this method is to create a PreparedStatement object on this connection and bind your parameters to this object.

4. It is better to implement the PreparedStatementCreator interface and other callback interfaces as inner classes if they are used within one method only.

a) True

b) False

View Answer

Answer: a

Explanation: This is because you can get access to the local variables and method arguments directly from the inner class, instead of passing them as constructor arguments.

5. PreparedStatementSetter, as its name indicates, create a PreparedStatement object on this connection the parameter as well as binding task of the overall update process.

a) True

b) False

View Answer

Answer: b

Explanation: The second callback interface, PreparedStatementSetter, as its name indicates, performs only the parameter binding task of the overall update process.

6. The JdbcTemplate class offers template method for batch update operations.

a) batchUpdate()

b) update()

c) all of the mentioned

d) none of the mentioned

View Answer

Answer: c

Explanation: It requires a SQL statement and a BatchPreparedStatementSetter object as arguments. In this method, the statement is compiled (prepared) only once and executed multiple times.

advertisement

7. The JdbcTemplate class declares a number of overloaded query() template methods to control the overall query process.
a) True
b) False
View Answer

Answer: c
Explanation: You can override the statement creation (task 2) and the parameter binding by implementing the PreparedStatementCreator and PreparedStatementSetter interfaces, just as you did for the update operations.

8. The primary interface that allows you to process the current row of the result set.
a) PreparedStatementSetter
b) PreparedStatementCreator
c) RowCallbackHandler
d) All of the mentioned
View Answer

Answer: c
Explanation: RowCallbackHandler is the is the primary interface that allows you to process the current row of the result set.

9. RowCallbackHandler purpose is to map a single row of the result set to a customized object.
a) True
b) False
View Answer

Answer: b
Explanation: RowCallbackHandler is the is the primary interface that allows you to process the current row of the result set.

10. Method of RowMapper interface in which, you have to construct the object that represents a row and return it as the method's return value.
a) mapRow()
b) query()
c) update()
d) none of the mentioned
View Answer

Answer: a
Explanation: From the viewpoint of reuse, it's better to implement the RowMapper interface as a normal class than as an inner class. In the mapRow() method of this interface, you have to construct the object that represents a row and return it as the method's return value.

11. RowMapper implementation which can automatically map a row to a new instance of the specified class.
a) BeanPropertyRowMapper
b) BeanPropertyRow
c) All of the mentioned
d) None of the mentioned
View Answer

Answer: a
Explanation: Note that the specified class must be a top-level class and must have a default or no-argument constructor. It first instantiates this class and then maps each column value to a property by matching their names.

12. Method which provides list of maps.
a) queryForList()
b) update
c) query()
d) all of the mentioned
View Answer

Answer: a
Explanation: Without the help of RowMapper, you can still call the queryForList() method and pass in a SQL statement. The returned result will be a list of maps. Each map stores a row of the result set with the column names as the keys.

13. Spring JDBC framework offers a convenient class, to simplify your DAO implementation.
a) org.springframework.jdbc.core.support
b) org.springframework.jdbc.core.support.JdbcDaoSupport
c) all of the mentioned
d) none of the mentioned
View Answer

Answer: b
Explanation: Spring JDBC framework offers a convenient class, org.springframework.jdbc.core.support.JdbcDaoSupport, to simplify your DAO implementation. This class declares a jdbcTemplate property, which can be injected from the IoC container or created automatically from a data source.

14. The org.springframework.jdbc.core.support.JdbcDaoSupport class has a setDataSource() method and a setJdbcTemplate() method.
a) True
b) False
View Answer

Answer: a
Explanation: Your DAO class can extend this class to have these methods inherited.

15. Method to retrieve the JDBC template.
a) setJdbcTemplate()
b) getTemplate()
c) getJdbc()
d) getJdbcTemplate()
View Answer

Answer: d
Explanation: In your DAO methods, you can simply call the getJdbcTemplate() method to retrieve the JDBC template. You also have to delete the dataSource and jdbcTemplate properties, as well as their setter methods, from your DAO class, because they have already been inherited. Again, for simplicity's sake, only the change to the insert() method is shown.

Sanfoundry Global Education & Learning Series – Java Spring.
To practice all areas of Java Spring, here is complete set of 1000+ Multiple Choice Questions and Answers (https://www.sanfoundry.com/1000-spring-questions-answers/).

« Prev Page - Spring Questions and Answers – Problems with Direct JDBC (https://www.sanfoundry.com/spring-questions-answers-problems-direct-jdbc/)
» Next Page - Spring Questions and Answers – Using the Simple JDBC Template and Handling Exceptions (https://www.sanfoundry.com/spring-quiz-questions-answers/)

Deep Dive @ Sanfoundry:

1. **Programming Questions and Answers (https://www.sanfoundry.com/programming-questions-answers/)**
2. **Java Programming Examples on Multithreading (https://www.sanfoundry.com/java-programming-examples-multithreading/)**
3. **Java Programming Examples on Classes (https://www.sanfoundry.com/java-programming-examples-classes/)**
4. **Object Oriented Programming Questions and Answers (https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/)**
5. **C++ Programming Examples on STL (https://www.sanfoundry.com/cpp-programming-examples-stl/)**
6. **C# Programming Examples on Interfaces (https://www.sanfoundry.com/csharp-programming-examples-on-interfaces/)**
7. **Java Programming Examples on Java.Lang (https://www.sanfoundry.com/java-programming-examples-java-lang/)**