Sr.Java Web Developer Job Interview Questions And Answers

Interview Questions Answers

[About Interview Questions Answers](#)

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Sr.Java Web Developer will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Sr.Java Web Developer Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Sr.Java Web Developer category. To ensure quality, each submission is checked by our team, before it becomes live. This [Sr.Java Web Developer Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
https://twitter.com/InterviewQA

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
Support@InterviewQuestionsAnswers.ORG

Sr.Java Web Developer Interview Questions And Answers Guide.

Question - 1:

What is executeQuery(String query)?

Ans:

Statement executeQuery(String query) is used to execute Select queries and returns the ResultSet. ResultSet returned is never null even if there are no records matching the query. When executing select queries we should use executeQuery method so that if someone tries to execute insert/update statement it will throw java.sql.SQLException with message "executeQuery method can not be used for update".

View All Answers

Question - 2:

Do you know what is JSON? Can you represent JSON as Java Object?

Ans:

JSON stands for Javascript object notation and is used to initialize Javascript objects. Yes it can be used as Java object also. Provide more info here

View All Answers

Question - 3:

Tell me how do you differentiate between Core Java and Enterprise Java?

Ans:

Core Java is something that provides the API's like regular expression, String handling, collections. But enterprise java involves writing scalable, secure and per-formant applications which can have large user base.

View All Answers

Question - 4:

Explain me what do you mean by batch processing in JDBC?

Ans:

Batch processing helps you to group related SQL statements into a batch and execute them instead of executing a single query. By using batch processing technique in JDBC, you can execute multiple queries which makes the performance faster.

View All Answers

Question - 5:

Tell us how to disable caching on back button of the browser?

Ans:

```
<%
response.setHeader("Cache-Control","no-store");
response.setHeader("Pragma","no-cache");
response.setHeader ("Expires", "0");          //prevents caching at the proxy server
%>
```

View All Answers

Question - 6:

Explain me what is the purpose of JDBC ResultSet interface?

Ans:

The ResultSet object represents a row of a table. It can be used to change the cursor pointer and get the information from the database.

View All Answers

Question - 7:

What is public StackTraceElement[] getStackTrace()?

Ans:

This method returns an array containing each element on the stack trace. The element at index 0 represents the top of the call stack whereas the last element in the array represents the method at the bottom of the call stack.

View All Answers

Question - 8:

Explain me what is JDBC Connection interface?

Ans:

The Connection interface maintains a session with the database. It can be used for transaction management. It provides factory methods that returns the instance of Statement, PreparedStatement, CallableStatement and DatabaseMetaData.

View All Answers

Question - 9:

Tell us what is a finally block? Is there a case when finally will not execute?

Ans:

Finally block is a block which always executes a set of statements. It is always associated with a try block regardless of any exception that occurs or not.
Yes, finally will not be executed if the program exits either by calling System.exit() or by causing a fatal error that causes the process to abort.

View All Answers

Question - 10:

Do you know what is the difference between frameworks like Jquery/DOJO and AJAX?

Ans:

Jquery and DOJO are java script frameworks for writing rich web applications but AJAX is a server communication mechanism which can issue requests without page reload.

View All Answers

Question - 11:

Tell us how is JSP better than Servlet technology?

Ans:

JSP is a technology on the server's side to make content generation simple. They are document centric, whereas servlets are programs. A Java server page can contain fragments of Java program, which execute and instantiate Java classes. However, they occur inside HTML template file. It provides the framework for development of a Web Application.

View All Answers

Question - 12:

Explain me what is the role of JDBC DriverManager class?

Ans:

The DriverManager class manages the registered drivers. It can be used to register and unregister drivers. It provides factory method that returns the instance of Connection.

View All Answers

Question - 13:

Tell us what is JDBC DatabaseMetaData interface?

Ans:

The DatabaseMetaData interface returns the information of the database such as username, driver name, driver version, number of tables, number of views etc.

View All Answers

Question - 14:

Explain me what do you generally do after you have resolved a problem?

Ans:

Perform the Root Cause Analysis and make sure the changes done have not effected any other module.

View All Answers

Question - 15:

Do you know what is synchronization?

Ans:

Synchronization refers to multi-threading. A synchronized block of code can be executed by only one thread at a time. As Java supports execution of multiple threads, two or more threads may access the same fields or objects. Synchronization is a process which keeps all concurrent threads in execution to be in sync. Synchronization avoids memory consistency errors caused due to inconsistent view of shared memory. When a method is declared as synchronized the thread holds the monitor for that method's object. If another thread is executing the synchronized method the thread is blocked until that thread releases the monitor.

Question - 16:

Please explain why should we not configure JSP standard tags in web.xml?

Ans:

We don't need to configure JSP standard tags in web.xml because when container loads the web application and find TLD files, it automatically configures them to be used directly in the application JSP pages. We just need to include it in the JSP page using taglib directive.
In case you are facing any challenges with these java interview questions, please comment your problems in the section below. Apart from this Java Interview Questions Blog.

Question - 17:

Do you know the role of DispatcherServlet and ContextLoaderListener?

Ans:

* DispatcherServlet is basically the front controller in the Spring MVC application as it loads the spring bean configuration file and initializes all the beans that have been configured. If annotations are enabled, it also scans the packages to configure any bean annotated with @Component, @Controller, @Repository or @Service annotations.
* ContextLoaderListener, on the other hand, is the listener to start up and shut down the WebApplicationContext in Spring root. Some of its important functions includes tying up the lifecycle of Application Context to the lifecycle of the ServletContext and automating the creation of ApplicationContext.

Question - 18:

Tell me what are the steps to connect to a database in java?

Ans:

* Registering the driver class
* Creating connection
* Creating statement
* Executing queries
* Closing connection

Question - 19:

Do you know what is a Spring?

Ans:

Wikipedia defines the Spring framework as "an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform." Spring is essentially a lightweight, integrated framework that can be used for developing enterprise applications in java.

Question - 20:

Tell me how to delete a Cookie in a JSP?

Ans:

The following code explain how to delete a Cookie in a JSP :
Cookie mycook = new Cookie("name1","value1");
response.addCookie(mycook1);
Cookie killmycook = new Cookie("mycook1","value1");
killmycook . set MaxAge (0);
killmycook . set Path ("/");
killmycook . addCookie (killmycook 1);

Question - 21:

Tell me what are the reasons for a page not found error and how will you sort it out?

Ans:

* 1) The URL being sent is wrong
* 2) The web.xml mapping is wrong
* 3) The web server is down
* 4) The application has not been deployed

Question - 22:

Tell me how does cookies work in Servlets?

Ans:

* Cookies are text data sent by server to the client and it gets saved at the client local machine.
* Servlet API provides cookies support through javax.servlet.http.Cookie class that implements Serializable and Cloneable interfaces.
* HttpServletRequest getCookies() method is provided to get the array of Cookies from request, since there is no point of adding Cookie to request, there are no

methods to set or add cookie to request.
* Similarly HttpServletResponse addCookie(Cookie c) method is provided to attach cookie in response header, there are no getter methods for cookie.
View All Answers

Question - 23:

Tell us what kind of HTTP request does the text generate?

Ans:

It will generate HTTP GET request
View All Answers

Question - 24:

What is void printStackTrace()?

Ans:

This method prints the stack trace information to the standard error stream.
View All Answers

Question - 25:

Tell me what steps will you take for ensuring the proper security of an web application?

Ans:

Stuff like Encryption, Authentication and Authorization
View All Answers

Question - 26:

What is synchronized Throwable getCause()?

Ans:

This method returns the cause of the exception or null id as represented by a Throwable object.
View All Answers

Question - 27:

Tell us what are some of the important Spring annotations which you have used?

Ans:

Some of the Spring annotations that I have used in my project are:
* @Controller - for controller classes in Spring MVC project.
* @RequestMapping - for configuring URI mapping in controller handler methods. This is a very important annotation, so you should go through Spring MVC RequestMapping Annotation Examples
* @ResponseBody - for sending Object as response, usually for sending XML or JSON data as response.
* @PathVariable - for mapping dynamic values from the URI to handler method arguments.
* @Autowired - for autowiring dependencies in spring beans.
* @Qualifier - with @Autowired annotation to avoid confusion when multiple instances of bean type is present.
* @Service - for service classes.
* @Scope - for configuring scope of the spring bean.
* @Configuration, @ComponentScan and @Bean - for java based configurations.
AspectJ annotations for configuring aspects and advices, @Aspect, @Before, @After, @Around, @Pointcut etc.
View All Answers

Question - 28:

Explain me what is JDBC ResultSetMetaData interface?

Ans:

The ResultSetMetaData interface returns the information of table such as total number of columns, column name, column type etc.
View All Answers

Question - 29:

Do you know what are the different methods of session management in servlets?

Ans:

Session is a conversational state between client and server and it can consists of multiple request and response between client and server. Since HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.
Some of the common ways of session management in servlets are:
* User Authentication
* HTML Hidden Field
* Cookies
* URL Rewriting
* Session Management API
View All Answers

Question - 30:

Can you name the different modules of the Spring framework?

Ans:

Some of the important Spring Framework modules are:
* Spring Context - for dependency injection.
* Spring AOP - for aspect oriented programming.
* Spring DAO - for database operations using DAO pattern
* Spring JDBC - for JDBC and DataSource support.
* Spring ORM - for ORM tools support such as Hibernate
* Spring Web Module - for creating web applications.
* Spring MVC - Model-View-Controller implementation for creating web applications, web services etc.

View All Answers

Question - 31:

Explain me how to integrate Spring and Hibernate Frameworks?

Ans:

We can use Spring ORM module to integrate Spring and Hibernate frameworks, if you are using Hibernate 3+ where SessionFactory provides current session, then you should avoid using HibernateTemplate or HibernateDaoSupport classes and better to use DAO pattern with dependency injection for the integration.
Also Spring ORM provides support for using Spring declarative transaction management, so you should utilize that rather than going for hibernate boiler-plate code for transaction management.

View All Answers

Question - 32:

Explain me what are the differences between Checked Exception and Unchecked Exception?

Ans:

Checked Exception:
* The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions.
* Checked exceptions are checked at compile-time.
* Example: IOException, SQLException etc.
Unchecked Exception:
* The classes that extend RuntimeException are known as unchecked exceptions.
* Unchecked exceptions are not checked at compile-time.
* Example: ArithmeticException, NullPointerException etc.

View All Answers

Question - 33:

Tell us what are the common issues you have faced in web applications and how did you resolve them?

Ans:

* 1) Server not starting up. Proper heap size not set
* 2) Migrating from JBoss to Weblogic. Wrote a number of XML configurations

View All Answers

Question - 34:

Explain me the Server and Database are working fine at your end but not on customer machine. What will you do?

Ans:

1) Check if the customer has not done any customizations
2) Provide a test build same as running at my end (ask customer to take a backup of their app)

View All Answers

Question - 35:

What is string getMessage()?

Ans:

This method returns the message String about the exception . The message can be provided through its constructor.

View All Answers

Question - 36:

Tell me how can you handle Java exceptions?

Ans:

There are five keywords used to handle exceptions in java:
* try
* catch
* finally
* throw
* throws

View All Answers

Question - 37:

Tell me how to disable session in JSP?

Ans:

<%@ page session="false" %>

View All Answers

Question - 38:

Do you know bean in Spring and List the different Scopes of Spring bean?

Ans:

Beans are objects that form the backbone of a Spring application. They are managed by the Spring IoC container. In other words, a bean is an object that is instantiated, assembled, and managed by a Spring IoC container.
There are five Scopes defined in Spring beans.
* Singleton: Only one instance of the bean will be created for each container. This is the default scope for the spring beans. While using this scope, make sure spring bean doesn't have shared instance variables otherwise it might lead to data inconsistency issues because it's not thread-safe.
* Prototype: A new instance will be created every time the bean is requested.
* Request: This is same as prototype scope, however it's meant to be used for web applications. A new instance of the bean will be created for each HTTP request.
* Session: A new bean will be created for each HTTP session by the container.
* Global-session: This is used to create global session beans for Portlet applications.

View All Answers

Question - 39:

Tell me how will you know whether a Java file is a servlet or not?

Ans:

It will extend from HttpServlet class

View All Answers

Question - 40:

Please explain what purpose does the keywords final, finally, and finalize fulfill?

Ans:

Final:
Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed. Let's take a look at the example below to understand it better.
class FinalVarExample {
public static void main(String args[])
{
final int a=10;   // Final variable
a=50;   //Error as value can't be changed
}
Finally
Finally is used to place important code, it will be executed whether exception is handled or not. Let's take a look at the example below to understand it better.
class FinallyExample {
public static void main(String args[]){
try {
int x=100;
}
catch(Exception e) {
System.out.println(e);
}
finally {
System.out.println("finally block is executing");}
}}
}
Finalize
Finalize is used to perform clean up processing just before object is garbage collected. Let's take a look at the example below to understand it better.
class FinalizeExample {
public void finalize() {
System.out.println("Finalize is called");
}
public static void main(String args[])
{
FinalizeExample f1=new FinalizeExample();
FinalizeExample f2=new FinalizeExample();
f1= NULL;
f2=NULL;
System.gc();
}
}

View All Answers

Question - 41:

Explain me what are the important benefits of using Hibernate Framework?

Ans:

Some of the important benefits of using hibernate framework are:
* Hibernate eliminates all the boiler-plate code that comes with JDBC and takes care of managing resources, so we can focus on business logic.
* Hibernate framework provides support for XML as well as JPA annotations, that makes our code implementation independent.
* Hibernate provides a powerful query language (HQL) that is similar to SQL. However, HQL is fully object-oriented and understands concepts like inheritance, polymorphism and association.
* Hibernate is an open source project from Red Hat Community and used worldwide. This makes it a better choice than others because learning curve is small and there are tons of online documentations and help is easily available in forums.
* Hibernate is easy to integrate with other Java EE frameworks, it's so popular that Spring Framework provides built-in support for integrating hibernate with Spring applications.
* Hibernate supports lazy initialization using proxy objects and perform actual database queries only when it's required.
* Hibernate cache helps us in getting better performance.
* For database vendor specific feature, hibernate is suitable because we can also execute native sql queries.
Overall hibernate is the best choice in current market for ORM tool, it contains all the features that you will ever need in an ORM tool.

View All Answers

Question - 42:

Explain me how to handle exceptions in Spring MVC Framework?

Ans:

Spring MVC Framework provides the following ways to help us achieving robust exception handling.
* Controller Based:
We can define exception handler methods in our controller classes. All we need is to annotate these methods with @ExceptionHandler annotation.
* Global Exception Handler:
Exception Handling is a cross-cutting concern and Spring provides @ControllerAdvice annotation that we can use with any class to define our global exception handler.
* HandlerExceptionResolver implementation:
For generic exceptions, most of the times we serve static pages. Spring Framework provides HandlerExceptionResolver interface that we can implement to create global exception handler. The reason behind this additional way to define global exception handler is that Spring framework also provides default implementation classes that we can define in our spring bean configuration file to get spring framework exception handling benefits.

View All Answers

Question - 43:

What is executeUpdate(String query)?

Ans:

Statement executeUpdate(String query) is used to execute Insert/Update/Delete (DML) statements or DDL statements that returns nothing. The output is int and equals to the row count for SQL Data Manipulation Language (DML) statements. For DDL statements, the output is 0.

View All Answers

Question - 44:

Tell me what are the JDBC API components?

Ans:

The java.sql package contains interfaces and classes for JDBC API.
Interfaces:
* Connection
* Statement
* PreparedStatement
* ResultSet
* ResultSetMetaData
* DatabaseMetaData
* CallableStatement etc.
Classes:
* DriverManager
* Blob
* Clob
* Types
* SQLException etc.

View All Answers

Question - 45:

Explain me what is the life-cycle of a servlet?

Ans:

There are 5 stages in the lifecycle of a servlet:
* Servlet is loaded
* Servlet is instantiated
* Servlet is initialized
* Service the request
* Servlet is destroyed

View All Answers

Question - 46:

Tell us how do you keep yourself updated about the latest web technologies?

Ans:

Whatever websites/blogs/forums/authors you follow.

View All Answers

Question - 47:

Explain me what are the common browser issues you will keep in mind while creating a web application?

Ans:

* 1) User pressing back/refresh button
* 2) Browser crashing
* 3) Session issues
* 4) Compatibility across web browsers

View All Answers

Question - 48:

Explain me what are the important methods of Java Exception Class?

Ans:

Methods are defined in the base class Throwable. Some of the important methods of Java exception class are stated below.
* String getMessage() - This method returns the message String about the exception . The message can be provided through its constructor.
* public StackTraceElement[] getStackTrace() - This method returns an array containing each element on the stack trace. The element at index 0 represents the top of the call stack whereas the last element in the array represents the method at the bottom of the call stack.
* Synchronized Throwable getCause() - This method returns the cause of the exception or null id as represented by a Throwable object.
* String toString() - This method returns the information in String format. The returned String contains the name of Throwable class and localized message.
* void printStackTrace() - This method prints the stack trace information to the standard error stream.

View All Answers

Question - 49:

Please explain what is exception hierarchy in java?

Ans:

The hierarchy is as follows:
Throwable is a parent class of all Exception classes. There are two types of Exceptions: Checked exceptions and UncheckedExceptions or RunTimeExceptions. Both type of exceptions extends Exception class whereas errors are further classified into Virtual Machine error and Assertion error.

View All Answers

Question - 50:

Do you know what is the jspDestroy() method?

Ans:

jspDestry() method is invoked from javax.servlet.jsp.JspPage interface whenever a JSP page is about to be destroyed. Servlets destroy methods can be easily overridden to perform cleanup, like when closing a database connection.

View All Answers

Question - 51:

Explain me what are the advantages of Hibernate over JDBC?

Ans:

Some of the important advantages of Hibernate framework over JDBC are:
* Hibernate removes a lot of boiler-plate code that comes with JDBC API, the code looks more cleaner and readable.
* Hibernate supports inheritance, associations and collections. These features are not present with JDBC API.
* Hibernate implicitly provides transaction management, in fact most of the queries can't be executed outside transaction. In JDBC API, we need to write code for transaction management using commit and rollback.
* JDBC API throws SQLException that is a checked exception, so we need to write a lot of try-catch block code. Most of the times it's redundant in every JDBC call and used for transaction management. Hibernate wraps JDBC exceptions and throw JDBCException or HibernateException un-checked exception, so we don't need to write code to handle it. Hibernate built-in transaction management removes the usage of try-catch blocks.
* Hibernate Query Language (HQL) is more object oriented and close to java programming language. For JDBC, we need to write native sql queries.
* Hibernate supports caching that is better for performance, JDBC queries are not cached hence performance is low.
* Hibernate provide option through which we can create database tables too, for JDBC tables must exist in the database.
* Hibernate configuration helps us in using JDBC like connection as well as JNDI DataSource for connection pool. This is very important feature in enterprise application and completely missing in JDBC API.
* Hibernate supports JPA annotations, so code is independent of implementation and easily replaceable with other ORM tools. JDBC code is very tightly coupled with the application.

View All Answers

Question - 52:

Can you list some of the important annotations in annotation-based Spring configuration?

Ans:

The important annotations are:
* @Required
* @Autowired
* @Qualifier
* @Resource
* @PostConstruct
* @PreDestroy

Question - 53:

What is execute(String query)?

Ans:

Statement execute(String query) is used to execute any SQL query and it returns TRUE if the result is an ResultSet such as running Select queries. The output is FALSE when there is no ResultSet object such as running Insert or Update queries. We can use getResultSet() to get the ResultSet and getUpdateCount() method to retrieve the update count.

Question - 54:

Tell me what is JDBC Driver?

Ans:

JDBC Driver is a software component that enables java application to interact with the database. There are 4 types of JDBC drivers:
* JDBC-ODBC bridge driver
* Native-API driver (partially java driver)
* Network Protocol driver (fully java driver)
* Thin driver (fully java driver)

Question - 55:

Explain me what is a servlet?

Ans:

* Java Servlet is server side technologies to extend the capability of web servers by providing support for dynamic response and data persistence.
* The javax.servlet and javax.servlet.http packages provide interfaces and classes for writing our own servlets.
* All servlets must implement the javax.servlet.Servlet interface, which defines servlet lifecycle methods. When implementing a generic service, we can extend the GenericServlet class provided with the Java Servlet API. The HttpServlet class provides methods, such as doGet() and doPost(), for handling HTTP-specific services.
* Most of the times, web applications are accessed using HTTP protocol and thats why we mostly extend HttpServlet class. Servlet API hierarchy is shown in below image.

Question - 56:

Tell me when will you use Servlet and JSP or MVC framework?

Ans:

While framework provides a number of components and allows one to concentrate more on the business logic but Servlets and JSP are used for controller and view layer respectively.

Question - 57:

What is String toString()?

Ans:

This method returns the information in String format. The returned String contains the name of Throwable class and localized message.

Question - 58:

Please explain can we write multiple catch blocks under single try block?

Ans:

Yes we can have multiple catch blocks under single try block but the approach should be from specific to general. Let's understand this with a programmatic example.
```
public class Example {
public static void main(String args[]) {
try {
int a[]= new int[10];
a[10]= 10/0;
}
catch(ArithmeticException e)
{
System.out.println("Arithmetic exception in first catch block");
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("Array index out of bounds in second catch block");
}
catch(Exception e)
{
System.out.println("Any exception in third catch block");
}
}
}
```

Question - 59:

Do you know how to create a custom Exception?

Ans:

To create you own exception extend the Exception class or any of its subclasses.
* class New1Exception extends Exception { }   // this will create Checked Exception
* class NewException extends IOException { }   // this will create Checked exception
* class NewException extends NullPonterExcpetion { } // this will create UnChecked exception

View All Answers

Question - 60:

Tell us what is difference between Error and Exception?

Ans:

An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error. These JVM errors you can not repair them at runtime.Though error can be caught in catch block but the execution of application will come to a halt and is not recoverable.
While exceptions are conditions that occur because of bad input or human error etc. e.g. FileNotFoundException will be thrown if the specified file does not exist. Or a NullPointerException will take place if you try using a null reference. In most of the cases it is possible to recover from an exception (probably by giving user a feedback for entering proper values etc.

View All Answers

Question - 61:

Tell me what are the different tags provided in JSTL?

Ans:

There are 5 type of JSTL tags.
* core tags
* sql tags
* xml tags
* internationalization tags
* functions tags

View All Answers

Question - 62:

Tell us what is Hibernate Framework?

Ans:

Object-relational mapping or ORM is the programming technique to map application domain model objects to the relational database tables. Hibernate is java based ORM tool that provides framework for mapping application domain objects to the relational database tables and vice versa.
Hibernate provides reference implementation of Java Persistence API, that makes it a great choice as ORM tool with benefits of loose coupling. We can use Hibernate persistence API for CRUD operations. Hibernate framework provide option to map plain old java objects to traditional database tables with the use of JPA annotations as well as XML based configuration.
Similarly hibernate configurations are flexible and can be done from XML configuration file as well as programmatically.

View All Answers

Question - 63:

Please explain what is the difference between execute, executeQuery, executeUpdate?

Ans:

Statement execute(String query) is used to execute any SQL query and it returns TRUE if the result is an ResultSet such as running Select queries. The output is FALSE when there is no ResultSet object such as running Insert or Update queries. We can use getResultSet() to get the ResultSet and getUpdateCount() method to retrieve the update count.
Statement executeQuery(String query) is used to execute Select queries and returns the ResultSet. ResultSet returned is never null even if there are no records matching the query. When executing select queries we should use executeQuery method so that if someone tries to execute insert/update statement it will throw java.sql.SQLException with message "executeQuery method can not be used for update".
Statement executeUpdate(String query) is used to execute Insert/Update/Delete (DML) statements or DDL statements that returns nothing. The output is int and equals to the row count for SQL Data Manipulation Language (DML) statements. For DDL statements, the output is 0.
You should use execute() method only when you are not sure about the type of statement else use executeQuery or executeUpdate method.

View All Answers

Question - 64:

Do you know what is composition in Java?

Ans:

Composition is again specialized form of Aggregation and we can call this as a "death" relationship. It is a strong type of Aggregation. Child object dose not have their lifecycle and if parent object deletes all child object will also be deleted. Let's take again an example of relationship between House and rooms. House can contain multiple rooms there is no independent life of room and any room can not belongs to two different house if we delete the house room will automatically delete.

View All Answers

Java Programing Most Popular & Related Interview Guides

1 : **Core Java Interview Questions and Answers.**

2 : **Hibernate Interview Questions and Answers.**

3 : **IBM WebSphere Interview Questions and Answers.**

4 : **Advanced Java Interview Questions and Answers.**

5 : **Spring Framework Interview Questions and Answers.**

6 : **Full Stack Developer (Java) Interview Questions and Answers.**

7 : **JSF Interview Questions and Answers.**

8 : **JDBC Interview Questions and Answers.**

9 : **Java Swing Programming Interview Questions and Answers.**

10 : **Java JSP Programming Interview Questions and Answers.**

Follow us on FaceBook
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter
https://twitter.com/InterviewQA

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team
https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG