Which Statement method allows us to execute dynamic sql queries, and returns boolean value.  
A. executeUpdate();  
B. executeQuery();  
C. execute();  
A) A  
B) B  
C) C  
  
ANS) C  
  
  
1.how many aspects can developers concentrate at a time ?  
a) two  
b) three  
c) one  
d) more than one  
  
1-a  
  
2.Does AOP and OOP follow same dependency ?  
a) YES  
b) NO   
,2-b  
  
3.AOP stands for ?  
a) Angular oriented project  
b) Aspect oriented programing  
c) Aspect oriented project  
d) Any Object Programing  
,3-b  
  
4.AOP is the replacement of OOP ?  
a) TRUE  
b) FALSE  
,4-b  
  
  
5.What is the key component of AOP ?  
a) Aspect  
b) Cross-Cutting  
c) Advice  
d) Point cut  
,5-a  
  
6.What are the unique correct answers about Spring AOP support ?  
a) An Advice could be proxied a constructor class  
b) A point cut could select methods that have a custom annotation   
c) Static initialization code could be target by a point cut  
d) Combinations of pointcuts by &&, || and the ! operators is not supported  
,6-b  
  
7.Which one is the correct statement about AOP proxy  
a) AOP proxies are created by Spring in order to implement the aspect contracts  
b) AOP proxies are always created with a JDK dynamic proxy  
c) Only classes that implements a least one interface could be proxied  
d) All methods could be proxied  
,7-a  
   
  
8.Aspects cannot advise other Aspects?  
a) TRUE  
b) FALSE  
,8-a  
  
9.Methods with which visibility be adviced?  
a) public  
b) private  
c) protected  
d) default  
,9-a  
  
10.Weavig occurs at which point in target object's lifetime?  
a) Compile Time  
b) Classload Time  
c) Runtime  
d) All  
,10-d  
 Around Advice,Throws Advice,After Advice,Before Advice   
  
  
  
11.Which Configuration is used for Around Advice in SpringAop?  
a)<aop:around/>  
b)<Spring aop:aroound/>  
c)<AOP:Around>  
d)<SpringAOP:around/>  
11-a  
  
12.From given,which is Correct for Around Advice?  
a)<AOP:Around method="logaroundAllmenthods" pointcut-ref="LoggingPointCuts"/>  
b)<aop:around method="logAroundAllMenthods" pointcut-ref="loggingPointcuts"/>  
c)<AOP:Around Method="logAroundAllMenthods" Before-ref="LoggingPointCuts"/>  
d)<AOP:Around method="logaroundAllmenthods" After-ref="LoggingPointCuts"/>  
,12-b  
  
13.What interface is implemented for creating Throws Advice  
a)Advice Throws  
b)advice:Throws  
c)Throws Advice  
d)throws:advice  
,13-c  
  
14.Syntax of ThrowsAdvice  
a)public void afterThrowing(Exception e)  
b)public void method(afterThrowing)  
c)public static void main(after Throwing Exception e)  
d)after Throwing(Exception e)  
,14-a  
  
15.@Before is an advice type ensures that an advice run before the method execution  
a)TRUE  
B)FALSE  
,15-a  
  
16.For After Advice,declared by using @After Annotation  
a)method executed after a matched methods executed   
b)Executed before declaring variables  
c)run directly after declaring matched annotations  
d)Compilation fails without method declaration  
,16-a  
  
17.Before advice will run  
a)after method execution  
b)before method execution  
,17-b  
  
18.Advices are actions taken for a particular jointpoint  
a)TRUE  
b)FALSE  
,18-a  
  
19.After Advice also termed as  
a)After(finally) Advice  
b)After(Static) Advice  
c)After(dynamic) Advice  
d)After(jointpoint)Advice  
,19-a  
  
20.Which advice is used to decide the JointPoint method to execute or not?  
a)Before Advice  
b)After Advice   
c)Throws Advice  
d)Around Advice  
,20-d  
  
 Transaction Management and Declarative Transaction Management  
  
  
  
21.which of the following defines the transaction strategy ?  
  
a)org.springframework.transaction.PlatformTransactionManager  
b)org.springframework.transaction.transactionsystemexception  
c)org.springframework.transaction.annotation.transactional  
d)org.springframework.transaction.annotation.transactional jar  
21-a  
  
22.Transactions can be described with key properties ?  
a) Atomicity  
b) Consistency  
c) Isolation  
d) All of the mentioned  
,22-d  
  
23.To access a database running on the Derby server, you have to add ?  
a) Derby client library  
b) Tomcat client library  
c) All of the mentioned  
d) None of the mentioned  
,23-a  
  
24.Spring’s transaction support offers a set of technology-independent facilities, including transaction managers ?  
a) org.springframework.transaction.PlatformTransactionManager  
b) org.springframework.transaction.support.TransactionTemplate  
c) all of the mentioned  
d) none of the mentioned  
,24-c  
  
25.Spring has several built-in implementations of PlatformTransactionManager interface for use with different transaction management APIs.  
a) True  
b) False  
,25-a  
  
26.Method that allows you to start a new transaction (or obtain the currently active transaction).  
a) getTransaction()  
b) commit()  
c) rollback()  
d) all of the mentioned  
,26-a  
  
27. A TransactionTemplate can accept a transaction callback object that implements ?  
a) TransactionCallback  
b) TransactionCallbackWithoutResult class  
c) All of the mentioned  
d) None of the mentioned  
,27-c  
  
28.You can omit the transaction-manager attribute in the element if your transaction manager has the name transactionManager ?  
a) True  
b) False  
,28-a  
  
29.To help you control the overall transaction management process and transaction exception handling.  
a) SpringTransactionTemplate  
b) TransactionTemplate  
c) Transaction  
d) None of the mentioned  
,29-b  
  
30.A transaction manager is declared in the Spring IoC container as a normal bean ?  
a) True  
b) False  
,30-a  
  
  
 Annotation Driven Aspectj AOP  
  
  
  
31.Which interface is implemented to specify precedence of aspects?  
a) Ordered  
b) ApplicationAspect  
c) AspectPointcut  
d) None of the mentioned  
31-a  
  
  
32. An advice is an action which comes into play at pointcuts.  
a) True  
b) False  
,32-b  
  
  
33.Which tag informs the spring container about the use of AspectJ annotation?  
a) aop:aspectj-autowire  
b) aop:aspectj-name  
c) aop:aspectj-autoproxy  
d) none of the mentioned  
,33-c  
  
  
34.An advice can access the current join point information by declaring an argument of type org.aspectj.lang.AdvicePoint in the advice method signature.  
a) True  
b) False  
,34-b  
  
35.Alternative annotative way to specify precedence of aspects?  
a) @Order  
b) @Aspect  
c) @PointCut  
d) None of the mentioned  
,35-a  
  
36.what will be the output of the code snippet  
  
 import org.aspectj.lang.annotation.Aspect;  
 import org.aspectj.lang.annotation.AfterThrowing;  
   
 @Aspect  
 public class AfterThrowingExample   
 {  
   
 @AfterThrowing(  
 pointcut="com.xyz.myapp.SystemArchitecture.dataAccessOperation()",  
 throwing="ex")  
 public void doRecoveryActions(DataAccessException e)  
   
 {  
 throw new IllegalArgumentException();  
 // [/expand]  
 }  
   
 }  
  
 a) Runtime Error  
 b) IllegalArgumentException  
 c) BeanCreation Exception  
 d) None of the mentioned  
 ,36-c  
37.Which instantiation model is supported by AspectJ?  
a) perthis  
b) pertarget  
c) none of the mentioned  
d) all of the mentioned  
,37-d  
  
  
38.An event in an executing program where the advice associated with an aspect may be executed is known as  
a) aspect  
b) join point  
c) join point model  
d) pointcut  
,38-b  
  
  
39.The incorporation of advice code at the specified join points by an aspect weaver is called”.  
a) aspect  
b) join point  
c) join point model  
d) weaving  
,39-d  
  
  
40.Annotation used to refer poincuts?  
a) @Pointcut  
b) @PointcutExecution  
c) @PointcutBefore  
d) None of the mentioned  
,40-a  
  
  
  
  
 Programatic spring AOP,point cut  
  
  
41.Which advice is executed only when joint point returns or throws an exception?  
  
A. @Before   
B. @After   
C. @AfterReturning   
D. @AfterThrowing  
41-c  
  
42. --- bean is required to identify the methods to be advised?  
  
 A.JoinPoint  
 B.PointCut  
 C.Advisor  
,42-b  
  
  
  
43. Which object is used in the around advice to get the intercepted method invoked?  
  
 A.MethodInterceptor   
 B.InvocationHandler   
 C.MethodInvocation   
  
,43-c  
  
  
44. How to declare aspect in Spring AOP?  
  
 <bean id="myAspect" class="com.concretepage.MyAspect">  
 <!-- configure properties of aspect here -->  
 </bean>  
  
  
  
  
45. Is the following pointcut expression correct?  
  
execution(\* ArithmeticCalculator.\*(..))  
  
A. Yes   
B. No   
C. If every target class is in same package   
D. Depends where target class is located  
  
45-c  
  
46. How to declare aspect in Spring AOP?  
  
ans: in XML  
 ------  
  
 <bean class="com.doj.aop.LoggingAspect" id="loggingAspect">  
 <!-- configure properties of aspect here --!>  
 </bean>  
  
 in Java  
 -------  
 @Aspect  
 @Component  
 class LoggingAspect{  
 //advice  
 //pointcut  
 }  
  
  
  
47. Select method's signature that match with the following pointcut:  
execution(\* com.test.service..\*.\*(\*))  
  
A. void com.test.service.MyServiceImpl#transfert(Money amount)  
B. void com.test.service.MyServiceImpl#transfert(Account account, Money amount)  
C. void com.test.service.account.MyServiceImpl#transfert(account account, Money amount)  
D. none  
47-a  
  
  
48. Using the Spring AOP framework, what are the joinpoints methods of the following pointcut expression?  
execution(public \*\*(..))  
  
A. The execution of all public method  
B. The execution of all public method returning a value  
C. The execution of all public method having atleast one parameter  
D. The execution of all public method in class belonging to the default package  
48-a  
  
  
49. Using the Spring AOP framework, what is the visibility of the methods by the following joinpoint?  
@Pointcut("expression(\*\*(..))")  
private void nyOperation(){};  
  
A. All methods, whereas there visiblity  
B. All methods, except private method  
C. Protected and public methods  
D. Public methods  
49-d  
  
  
50.Considering 2 classes AccountServiceImpl and ClientServiceImpl.Any of these classes inherits from eachother.  
What is the result of the pointcut expressions?  
execution(\*\*..AccountServiceImpl.update(..))  
&& execution(\*\*..ClientServiceImpl.update(..))  
  
A. Matches public update methods of the 2 classes,whatever the arguments.  
B. Matches any update methods of the 2 classes,whatever the arguments and method visiblity.  
C. Matches any update methods of the 2 classes,with one more arguments and whatever method visiblity.  
D. No joint point defined  
50-d  
  
 AspectJAOP  
  
  
51. Which of the following is advice supported by Aspect Annotation?  
A. @Before  
B. @After  
C. @AfterReturning  
D. All of the mentioned   
 51-d   
  
52. Which advice is executed only when joint point returns or throws an exception?  
A. @Before  
B. @After  
C. @AfterReturning  
D. @AfterThrowing  
,52-c  
  
53. Special compiler used during weaving  
A. jvm  
B. gcc  
C. ajc  
D. none of the mentioned  
,53-c  
  
54. How to inject Spring bean into domain objcts  
A. AOP  
B. XML  
C. AspectJ  
D. Java Based  
,54-a  
  
55. Lightweight java scripting language executing java fragments  
A. JRuby  
B. Groovy  
C. BeanShell  
D. All of the mentioned  
,55-c  
  
56. XML element to specify the property values for scripting bean  
A. lang:property  
B. bean:property  
C. script:property  
D. None of the mentioned  
,56-a  
  
57. Annotation used to declare an introduction  
A. Before  
B. After  
C. @DeclareParents  
D. None of the mentioned  
,57-c  
  
  
58. Is the following pointcut expression correct?  
  
execution(\* ArithmeticCalculator.\*(..))  
A. Yes  
B. No  
C. If every target class is in same package  
D. Depends where target class is located  
,58-c  
  
59. Dynamic-language-backed bean with some configurations  
A. Refreshable bean  
B. Beanshell  
C. Scripting Beans  
D. Inline Script  
,59-a  
  
  
60.Which of the following aspect implementation spring supports?  
A.XML Schema based aspect implementation  
B.@AspectJ based aspect implementation  
C.Both of above.  
D.None of above.  
,60-c  
  
  
 Annotations driven Transaction Management and 100% java code based transaction management  
  
  
  
61.Transactions can be described with key properties:-  
a) Atomicity  
b) Consistency  
c) Isolation  
d) All of the mentioned  
61-d  
  
62.To access a database running on the Derby server, you have to add:-  
a) Derby client library  
b) Tomcat client library  
c) All of the mentioned  
d) None of the mentioned  
,62-a  
  
63 Spring’s transaction support offers a set of technology-independent facilities, including transaction managers.  
a) org.springframework.transaction.PlatformTransactionManager  
b) org.springframework.transaction.support.TransactionTemplate  
c) all of the mentioned  
d) none of the mentioned  
,63-c  
  
64.Spring’s core transaction management abstraction is based on the interface:-  
a) PlatformTransaction  
b) PlatformTransactionManager  
c) TransactionManager  
d) PlatformManager  
,64-b  
  
65.The PlatformTransactionManager interface provides methods for working with transactions:  
a) getTransaction(TransactionDefinition definition)  
b) commit(TransactionStatus status)  
c) rollback(TransactionStatus status)  
d) all of the mentioned  
,65-d  
  
66.Spring has several built-in implementations of PlatformTransactionManager interface for use with different transaction management APIs.  
a) True  
b) False  
,66-a  
  
67.Method that allows you to start a new transaction (or obtain the currently active transaction).  
a) getTransaction()  
b) commit()  
c) rollback()  
d) all of the mentioned  
,67-a  
  
68.To help you control the overall transaction management process and transaction exception handling.  
a) SpringTransactionTemplate  
b) TransactionTemplate  
c) Transaction  
d) None of the mentioned  
,68-b  
  
69. A TransactionTemplate can accept a transaction callback object that implements:-  
a) TransactionCallback  
b) TransactionCallbackWithoutResult class  
c) All of the mentioned  
d) None of the mentioned  
,69-c  
  
  
70.Spring (since version 2.0) offers a transaction advice that can be easily configured via the:-  
a) rx:advice  
b) bx:advice  
c) tx:advice  
d) none of the mentioned  
,70-c  
  
  
  
  
  
  
  
   
1-a,2-b,3-b,4-b,5-a,6-b,7-a,8-a,9-a,10-d  
11-a,12-b,13-c,14-a,15-a,16-a,17-b,18-a,19-a,20-d  
21-a,22-d,23-a,24-c,25-a,26-a,27-c,28-a,29-b,30-a  
31-a,32-b,33-c,34-b,35-a,36-c,37-d,38-b,39-d,40-a  
41-c,42-b,43-c,44-explaination,45-c,46-explaination,47-a,48-a,49-d,50-d  
51-d,52-c,53-c,54-a,55-c,56-a,57-c,58-c,59-a,60-c  
61-d,62-b,63-c,64-b,65-d,66-a,67-a,68-b,69-c,70-c