

1. Does Spring provide programmatic transaction management? Select a unique answer.

- a. Yes using the `@Transactional` bean post processor
- b. Yes with the `TransactionTemplate` class
- c. Yes with the `TransactionService` class
- d. Yes using the `@Transactional` bean post processor

ans: b

Explanation

- 1. The `TransactionTemplate` class provides an `execute(TransactionCallback)` method
- 2. The `TransactionService` class does not exist
- 3. The `@Transactional` annotation is for declarative transaction management

2. How could you externalize constants from a Spring configuration file or a Spring annotation into a `.properties` file? Select one or more

- a. By using the `<context:property-placeholder />` tag
- b. By using the `<util:constant />` tag
- c. By declaring the `ConstantPlaceholderConfigurer` bean post processor
- d. By using the `c:` namespace

ans: a

3. To validate Java beans in a web application using annotations.

- a) XML
- b) Java Based
- c) JAR-303 standard
- d) All of the mentioned

ans: d

4. Annotation used to indicate a field has to have a minimum of 2 characters.

- a) `@NotNull`
- b) `@Size`
- c) `@MaxSize`
- d) `@size`

Answer: b

Explanation: `@Size` annotation used to indicate a field has to have a minimum of 2 characters.

5. Spring MVC supports generating Excel files using which of the following libraries.

- a) Apache POI library
- b) JExcelAPI library
- c) All of the mentioned
- d) None of the mentioned

ans: C

6. Interface for Dispatcher Servlet to auto detect view resolver beans.

- a) `localeResolver`
- b) `tiles`
- c) `ViewResolver`
- d) none of the mentioned.

ans: C

Explanation: Spring MVC comes with several `ViewResolver` implementations for you to resolve views using

different strategies.

7. By default, XmlViewResolver loads view beans from /WEB-INF/views.xml which is final.

- a) True
- b) False

ans: b

8. Views that can't be resolved by InternalResourceViewResolver.

- a) redirect
- b) redirect:prefix
- c) redirect:suffix
- d) all of the mentioned

ans: a

9. Annotation which allows a controller's field to be assigned using Spring Expression Language(SpEL)

- a) @Value
- b) @After
- c) @Default
- d) @None

ans: a

10. SpEL statements are recognizable.

- a) True
- b) False

View Answer

Answer: a

Explanation: They use a notation in the form “#{ SpEL statement }”.

11. To publish a REST service with Spring.

- a) publishing an application's data as a REST service
- b) accessing data from third-party REST services
- c) none of the mentioned
- d) all of the mentioned

ans: a b

12. Publishing an application's data as a REST service requires.

- a) @RequestMapping
- b) @PathVariable
- c) All of the mentioned
- d) None of the mentioned

ans: a b

13. Annotation added as an input parameter to the handler method.

- a) @PathVariable
- b) @Path
- c) @PathLocale
- d) None of the mentioned

ans: a

14. Which of the following statements is true regarding the @ResponseStatus annotation?

- a. @ResponseStatus is detected on nested exceptions
- b. TheExceptionHandlerExceptionResolver uses the @ResponseStatus annotation to map exception to HTTP status code
- c. A controller handler is annotated with the @ResponseStatus, the response status set by RedirectView takes precedence over the annotation value.
- d. The @ResponseStatus annotation can go on a @RequestMapping method or a @RestController class or a

business exception class.

ans: a

15. Default localeResolver used by Spring.

- a) AcceptHeaderLocale
- b) AcceptHeader
- c) AcceptHeaderLocaleResolver
- d) AcceptLocaleResolver

ans: c

16. Alternative way to resolve locales.

- a) AcceptHeaderLocale
- b) AcceptHeader
- c) AcceptHeaderLocaleResolver
- d) SessionLocaleResolver

ans: d

17. General-purpose class that allows a response to be rendered using a marshaller.

- a) MarshallingView
- b) Marshalling
- c) View
- d) All of the mentioned

View Answer

Answer: a

Explanation: The member template view is defined as a MarshallingView type, which is a general-purpose class that allows a response to be rendered using a marshaller.

18. Annotation which allows the Jaxb2Marshaller marshaller to detect a class's (i.e., object's) fields.

- a) @XmlRootElement
- b) @XmlRoot
- c) @NotNull
- d) None of the mentioned

ans: a

19. How to auto-inject into a field a Spring bean by its name? Select one or more answer choices.

- a. By using both the @Autowired and the @Qualifier Spring annotations
- b. By using the @Autowired annotation and naming the field with the bean name
- c. With the name attribute of the @Autowired annotation

d. By using the single `@Qualifier` annotation
ans : a

20. Considering 2 classes `AccountServiceImpl` and `ClientServiceImpl`. Any of these 2 classes inherits from each other. What is the result of the following pointcut expression?
`execution(* *..AccountServiceImpl.update(..))`
`&& execution(* *..ClientServiceImpl.update(..))`
a.No joint point is defined
b.Matches public update methods of the 2 classes, whatever the arguments
c.Matches any update methods of the 2 classes, whatever the arguments and method visibility
d.Matches any update methods of the 2 classes, with one more arguments and whatever method visibility
ans: a

21. Select the right statement about referring a Spring configuration file inside the package `com.example.myapp` in the below example?
`ApplicationContext context = new`
`ClassPathXmlApplicationContext("classpath:/com.example.myapp.config.xml");`
a. all of the above
b. The `classpath:` prefix could be omitted
c.Package name using the dot character is not well formatted
d.The slash character preceding `com.example` could be omit
ans: a

22. Using the Spring AOP framework, what is the visibility of the method matches by the following join point?
`@Pointcut("execution(* *(..))")`
`private void anyOperation() {};`
a. Public methods
b. All methods, whereas their visibility
c.All methods, except private method
d. Protected and public methods
ans: a

23. What the name of the bean defined in the following configuration class? Select a single answer.
`@Configuration`
`public class ApplicationConfig {`
`@Autowired`
`private DataSource dataSource;`
`@Bean`
`ClientRepository clientRepository() {`
`ClientRepository accountRepository = new JpaClientRepository();`
`accountRepository.setDataSource(dataSource);`
`return accountRepository;`
`}`
`}`
a. `clientRepository`

- b. JpaClientRepository
 - c. jpaClientRepository
 - d. Two beans are defined: a data source and a repository
- ans: a

24. Which of the following is true regarding the below Spring controller?

```
@RestController
public class OwnerController {
    @RequestMapping(value = "/owner/{ownerId}", method = RequestMethod.POST)
    @ResponseBody
    public Owner findOwner(@PathVariable("ownerId") int ownerId) {
        return new Owner();
    }
}
```

- a. RequestMethod.GET method is more accurate than POST
- b. @ResponseBody could be removed
- c. @PathVariable should be replaced with the @PathParam annotation
- d. Returning the 201 HTTP status code is better

ans: a

25. What is an advice? Select a unique answer.

- a. An action taken by an aspect at a particular join point
- b. A point during the execution of a program
- c. An aspect and a pointcut
- d. A predicate that matches join points

ans: a

26. What is the easiest method to write a unit test?

- a. @RequestMapping("/displayAccount")
String displayAccount(@RequestParam("accountId") int id, Model model)
- b. void displayAccount(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException
- c. void displayAccount(HttpServletRequest req, HttpSession session) throws ServletException, IOException
- d. @RequestMapping("/displayAccount")
String displayAccount(@PathVariable("accountId") int id, Model model)

ans: a

27. Select method's signatures that match with the following pointcut:

execution(* com.test.service..*.*(*))

- a. void com.test.service.MyServiceImpl#transfert(Money amount)
- b. void com.test.service.account.MyServiceImpl#transfert(Money amount)
- c. void com.test.service.MyServiceImpl#transfert(Account account, Money amount)
- d. void com.test.service.account.MyServiceImpl#transfert(Account account, Money amount)
- e. None of the above

ans: a

28. Given the following configuration class, what are the correct affirmations? Select one or more answers.

```

public class ApplicationConfig {
private DataSource dataSource;
@Autowired
public ApplicationConfig(DataSource dataSource) {
this.dataSource = dataSource;
}
@Bean(name="clientRepository")
ClientRepository jpaClientRepository() {
return new JpaClientRepository();
}
}

```

- a. Configuration annotation is missing
- b. Default or no-arg constructor is missing
- c. @Bean name is ambiguous
- d. @Bean scope is prototype

ans: a

29. Using JdbcTemplate, what is the Spring provided class you will use for result set parsing and merging rows into a single object? Select a unique answer.

- a. ResultSetExtractor
- b. RowMapper
- c. RowCallbackHandler
- d. ResultSetMapper

ans: a

30. Spring-WS provides various abstract endpoint classes for you to process the request.

- a) org.springframework.ws.server.endpoint
- b) org.springframework.ws.server
- c) org.springframework.*
- d) none of the mentioned

Ans:a

31. Which method is used to gracefully shutdown all the bean processes after closing the spring container?

- A. shutdownHook
- B. destory method
- C. none of the mentioned
- D. all of the mentioned

Ans:A

32. Declaring bean form object properties can be done using:-

- A. PropertyPathFactoryBean
- B. util:constant
- C. None of the mentioned

D. All of the mentioned

Ans:A

33. What are the ways to create custom Property Editors?

- A. implement PropertyEditor interface
- B. extend PropertyEditorSupport Class
- C. none of the mentioned
- D. all of the mentioned

Ans:A,B

34. Spring Web MVC (and in Spring in general) is the "Open for extension, closed for modification" principle. The reason that this principle is being mentioned here is because a number of methods in the core classes in Spring Web MVC are marked final.

A.true

B.false

Ans:A

35. Which of the among is not a special bean in WebApplicationContext?

- a.Locale resolver
- b.Theme resolver
- c.multipart file
- d.resolver
- e.Handler exception resolver
- f.Handler mappings
- g. Non of the above

Ans:g

36. How the Spring DispatcherServlet supports for returning the last-modification-date, as specified by the Servlet API ?

- a.By implementing the LastModified interface
- b.By implementing the LastSessionModified interface
- c.By implementing the LastDateModified interface
- d.By implementing the LastModifiedDate interface

Ans:A

37. @RequestMapping annotation is used to map a HTTP request method (GET or POST) to a specific class or method in the controller which will handle the respective request?

- 1. false
- 2. true

Ans:true

38. Which of the following exception classes is related to all the exceptions thrown in spring applications?

1. `DataAccessException`
2. `ArrayIndexOutOfBoundsException`.
3. `NullPointerException`.
4. `SpringException`.
5. None of these.

Ans:1

39. Which of the following is true for Controllers in Spring?

1. An abstract class.
2. An interface.
3. Concrete class.
4. Final class.
5. None of these.

Ans: An interface.

40. Annotations work as metadata for classes and Spring uses annotations as an alternative to XML for declarative configuration. Is it true or false?

1. false
2. true

Ans:true 2

41. Which of the following specify the points where weaving can be applied?

1. Compile Time , Classload Time , load time.
2. Compile Time , Classload Time , Runtime
3. Compile Time and Runtime.
4. Compile Time and Classload Time.
5. None of these.

Ans-2

42. Which of the following layers the `@Controller` annotation is used in?

1. Business layer
2. Service layer.
3. Presentation layer.
4. Session layer.
5. None of these.

Ans:3

43. Which Of The Following Is Not A Valid Difference Between `@Inject` And `@Autowired` Annotations?

1. `@Inject` annotation is part of Java EE 7's Context and Dependency Injection framework while `@Autowired` is the Spring Frameworks own implementation
2. `@Autowired` is a legacy annotation whereas `@Inject` is its latest version..

3. There is no difference and both can be used interchangeably.
4. None of these.

Ans:2

44. Which of the following statements is true?

1. BeanFactory extends ApplicationContext.
2. ApplicationContext extends BeanFactory.
3. ApplicationContext implements BeanFactory.
4. BeanFactory implements ApplicationContext
5. None of these.

Ans:2

45. Which of the following is advice supported by Aspect Annotation?

- A. @Before
- B. @After
- C. @AfterReturning
- D. All of the mentioned

Ans:d

46. Which advice is executed only when joint point returns or throws an exception?

- A. @Before
- B. @After
- C. @AfterReturning
- D. @AfterThrowing

Ans:C

47. Special compiler used during weaving

- A. jvm B. gcc
- C. ajc
- D. none of the mentioned

Ans:C

48. How to inject Spring bean into domain objects

- A. AOP
- B. XML C. AspectJ
- D. Java Based

Ans:D

49. Method used to process bean before initialization callback

- A. scope
- B. postProcessAfterInitialization()
- C. postProcessBeforeInitialization()
- D. it's own constructor

Ans:B

50. Which attribute is used to set the scope of the bean?

- A. setScope
- B. scope

C. getScope
D. none of the mentioned
Ans:B

51. Which attribute is used to specify classname of the bean?
A. name
B. id
C. class
D. constructor-args
Ans:B

52. Which tag is also allowed by static field?
A. util:constant
B. list C. set D. Constructor-args
Ans:A

53. How before advice works?
A - Run advice before a class loads.
B - Run advice before a method execution.
C - Run advice before http response is to be returned.
D - Run advice before http request is to be processed.
Answer : B
Explanation
before advice runs before a method execution.

54 - What is Introduction?
A - An introduction represents a point in your application where you can plug-in AOP aspect.
B - This is used to inject values in objects.
C - This is not invoked during program execution by Spring AOP framework.
D - An introduction allows you to add new methods or attributes to existing classes.
Answer : D

55 - How to use ref keyword in beans.
xml?
A - Using setter method only.
B - Using constructor argument only.
C - Using setter method and constructor argument both.
D - None of the above.
Answer : C

Explanation
ref is used to reference already defined bean and it can

56- How do you turn on annotation wiring?
A - Add <annotation-context:config /> to bean configuration.
B - Add <annotation-config /> to bean configuration.
C - Add <annotation-context-config /> to bean configuration.
D - Add <context:annotation-config/> to bean configuration.

Answer : D

Explanation

Add `<context:annotation-config/>` to bean configuration to turn on annotation wiring.

57 - What is Join point?

A - This represents a point in your application which joins two objects.

B - This represents a point in your object where you join values.

C - This represents a point in your object where you join injected values.

D - This represents a point in your application where you can plug-in AOP aspect.

Answer : D

Explanation

Join point represents a point in your application where you can plug-in AOP aspect.

58 - What are the different points where weaving can be applied?

A - Compile time, load time

B - Compile time, run time

C - Run time

D - Compile time, load Time, Run time

Answer : D

Explanation

Weaving can be applied at Compile time, load Time and Run time.

59 - Which of the following database is not supported using jdbcTemplate?

A - MySql

B - PostgreSQL

C - NoSql

D - Oracle

Answer : C

Explanation

NoSql is not supported using jdbcTemplate.

60. What is bean scope?

A - Bean scope forces Spring to produce a new bean instance as per the scope defined.

B - Bean scope defines the accessibility of bean in a java class.

C - Bean scope defines the accessibility of bean in a java package.

D - Bean scope defines the accessibility of bean in a web application.

Answer : A

Explanation

Bean scope instructs Spring Container to produce a new bean instance as per the scope defined.

61. 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.

Answer : C

Explanation

Spring supports both XML Schema based and @AspectJ based aspect implementation.

62. - Core container has AOP as one of its module.

A - True

B - False

Answer : B

Explanation

AOP is not the part of spring core container.

63. - Which of the following stands true for spring beans?

A - Spring beans are managed by the Spring IoC container.

B - Spring beans are instantiated, assembled, and otherwise managed by a Spring IoC container.

C - Spring beans are simple POJOs.

D - All of the above.

Answer : D

Explanation

All of the above statements are correct.

64. - How to get object of a service in spring framework?

A - Using new keyword

B - Using dependency injection

Answer : B

Explanation

In spring, dependency injection is used to get any object.

65. What is Target object?

A - A represents a object in your application where you can plug-in AOP aspect.

B - The object being advised by one or more aspects, this object will always be a proxy object,

also referred to as the advised object.

C - This is used to inject values in objects.

D - This is not invoked during program execution by Spring AOP framework.

Ans:B

66- What is Weaving?

A - Weaving is the process of injecting values in objects to create an advised object.

B - Weaving is the process of linking aspects with other application types or objects to create an

advised object.

C - This is used to inject values in objects.

D - Weaving is used to check object dependencies.

Ans:B

67- What are the different points where weaving can be applied?

A - Compile time, load time

B - Compile time, run time

C - Run time

D - Compile time, load Time, Run time

Ans:D

68- What are the types of advice?

A - then, after, after-returning, after-throwing, around

B - When, after, after-returning, around

C - Where, after, after-returning, after-throwing, around

D - Before, after, after-returning, after-throwing, around

Ans:b

69 - How before advice works?

A - Run advice before a class loads.

B - Run advice before a method execution.

C - Run advice before http response is to be returned.

D - Run advice before http request is to be processed.

Ans:B

70 - How after-returning advice works?

A - Run advice after a class loads only if class loads successfully.

B - Run advice after a method execution only if method completes successfully.

C - Run advice after http response is returned only if http response is success.

D - Run advice after http request is processed with no exception.

Ans:B

71- How after-throwing advice works?

A - Run advice after a method execution only if method exits by throwing an exception.

B - Run advice after a class loads only if class throws exception during load time.

C - Run advice after http response is returned with error status.

D - Run advice after http request is processed and an exception occurred.

Ans:A

72. - How around advice works?

A - Run advice before and after the advised method is invoked.

B - Run advice before and after a class is loaded.

C - Run advice before and after http response is returned.

D - Run advice before and after http request is processed.

Ans:A

73- 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.

Ans:C

74 - What are the types of the transaction management Spring supports?

A - Programmatic transaction management

B - Declarative transaction management

C - Both of above.

D - None of above.

Ans:C

75- What is Spring MVC framework?

A - Spring MVC framework is Model-Value-Class architecture and used to bind model data with values.

B - The Spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications.

C - Spring MVC framework is used for Transaction management for Web Applications.

D - Spring MVC framework is used for AOP for Web Applications.

Ans:B

76 - What is @Controller annotation?

A - The @Controller annotation indicates that a particular class serves the role of a controller.

B - The @Controller annotation indicates how to control the transaction management.

C - The @Controller annotation indicates how to control the dependency injection.

D - The @Controller annotation indicates how to control the aspect programming.

Ans:A

77- What are the ways to access Hibernate by using Spring?

A - Inversion of Control with a Hibernate Template and Callback.

B - Extending HibernateDAOSupport and Applying an AOP Interceptor node.

C - Both of above.

D - None of above.

Ans:C

78 - Which ORM Spring supports ?

A - Hibernate

B - iBatis

C – JPA

D - All of above.

E - None of above.

Ans:D

79 - Which of the following is part of Data Access layer in Spring framework?

A - Beans

B - Aspects

C - JMS

D – Context

Ans:C

80 - How to use ref keyword in beans.

xml?

A - Using setter method only.

B - Using constructor argument only.

C - Using setter method and constructor argument both.

D - None of the above.

Ans:C

81 - Core container has AOP as one of its module.

A - True

B - False

Ans:B

82. - SpEL is part of core container.

A - False

B - True

Ans:B

83. - Which class acts as IoC Container?

A - ServletContext

B - DispatcherServlet

C - ApplicationContext

D - None of the above

Ans:C

84.- What stands true for spring framework?

A - Spring framework is a light weight framework.

B - Spring framework is a heavy weight framework.

Ans:A

85.- Expression Language/ SpEL was introduced in which version of spring framework.

A - 1.0

B - 2.0

C - 3.0

D - 4.0

Ans:C

86- Can we integrate Struts with Spring.

A - Yes

B - No

Ans:A

87 - By default a bean is lazily loaded.

A - True

B - False

Ans:B

88 - If a bean is scoped to HTTP request, scope is

A - session

B - global-session

C - prototype

D - request

Ans:D

89 - Thread scoped bean is introduced in which version of spring framework.

A - 1.0

B - 2.0

C - 3.0
D – 4.0
Ans:C

90. - What is the scope of stateless bean?
A - global-session
B - singleton
C - prototype
D – request
Ans:B

91. - What is the scope of stateful bean?
A - session
B - global-session
C - prototype
D – request
Ans:C

92 - How to handle shut down of IoC container?
A - Using shutdownHook
B - Using shutdownHandler
C - Using registerHook
D - Using registerShutdownHook
Ans:D

93 - How bean life cycle can be controlled?
A - Using init only
B - Using InitializingBean class only
C - Using DisposableBean class only
D - Using All of above
Ans:D

94- What is the scope of bean in portlet context?
A - session
B - global-session
C - prototype
D – request
Ans:B

95 - How after advice works?
A - Run advice after a method execution regardless of its outcome.
B - Run advice after a class loads.
C - Run advice after http response is returned.
D - Run advice after http request is processed.
Ans:A

96 - Which class is used to map a database row to a java object in spring?
A - ResultSet

B - RowMapper
C - RowSetMapper
D – ResultSetMapper
Ans:B

97

A bean must have id attribute in beans configuration file.
A - True
B – False
Ans:B

98 - Which of the following class can be used to execute Sql queries in spring?

B - JDBCHelper
A - JdbcTemplate
C - DBHelper
D - DBTemplate
Ans:A

99- Which of the following class can be used to call Stored Procedures in spring?

A - SPHelper
B - JdbcTemplateCall
C - JdbcTemplate
D – SimpleJdbcCall
Ans:D

100 - What is a DispatcherServlet?

A - DispatcherServlet is used for transaction management.
B - DispatcherServlet is used for AOP.
C - DispatcherServlet handles all the HTTP requests and responses.
D - DispatcherServlet is used for Dependency injection.
Ans:C

101. - What is ACID in transactional management?

A - Accurate, Controlled, Isolation, Durability
B - Atomicity, Consistency, Independent, Done
C - Atomicity, Consistency, Isolation, Durability
D - Accurate, Controlled, Independent, Done
Ans:C

102. - Where do you define DispatcherServlet?

A - In Beans configuration file.
B - Web.xml file
C - Meta-inf/dispatcher.xml
D - Web-inf/ dispatcher.xml
Ans:B

103. - What is true about BeanPostProcessor?

A - It is a concrete class.
B - It is an interface.
C - It is an abstract class.
D - None of the above.

Ans:A

104. - What BeanPostProcessor does?

A - It processes beans once a bean is initialized.

B - It defines callback methods that you can implement to provide your own instantiation logic, dependency-resolution logic etc.

C - It processes beans once a bean is loaded.

D - It processes beans once a bean exits.

Ans:B

105. - Can be bean be configured to have an inner bean?

A - True

B - False

Ans:A

106. - Can we inject value and ref both together in a bean?

A - True

B - False

Ans:A

107. - Following class can be extended to create custom event in spring.

A - SpringEvent

B - Event

C - ApplicationEvent

D - None of above

Ans:C