

学号	姓名	提交时间	总分	成绩
20225102050218	吕思琦	2024-07-02 18:21:15	100	69

1.【单选题】阅读以下配置文件代码，以下说法错误的是（）

得分：2/2



- .....
- A.这不是对SQL语句执行结果和实体数据的映射关系的映射文件，只是一个普通的.xml文件
  - B.它的根元素是元素
  - C.namespace属性是用来管理SQL映射语句的名称，在同一个命名空间中，映射语句不能重名
  - D.这里namespace属性的取值是接口的全限定名

学生答案：A

参考答案：A

2.【单选题】在Mybatis中，配置结果映射时，使用（）标签实现一对多的关联：（）

得分：2/2

- A.many-one
- B.one-many
- C.association
- D.collection

学生答案：D

参考答案：D

3.【单选题】根据存储模型划分，下列有关数据库类型说法正确的是：（）

得分：2/2

- A.Object-oriented Database 面向对象数据库
- B.Hierarchical Database 网状数据库
- C.Network Database 关系数据库
- D.商业应用中主要是面向对象数据库

学生答案：A

参考答案：A

4.【单选题】在MyBatis关于事务管理和数据源的配置代码中，说法错误的是：（）

得分：2/2

- A.environments配置的开发环境
- B.transactionManager配置事务管理，它有两种类型
- C.dataSource采用POOLED类型是，说明不采用数据库连接池
- D.username和password可以更改

学生答案：C

参考答案：C

5.【单选题】在MyBatis中内置了许多别名，下面有误的是：（）

得分：0/2

- A.\_integer（别名），int（全限定名）--基本类型
- B.integer（别名），java.lang.Integer（全限定名）--包装类型yBatis提供、元素类配置别名
- C.\_integer[]（别名），int[]（全限定名）--基本类型数组
- D.\_integer[]（别名），int[]（全限定名）--基本类型数组

学生答案：A

参考答案：D

6.【单选题】MyBatis编程除了映射文件外，还需要掌握几个关键接口或类的使用，下列选项不是MyBatis的接口或类是：（）

得分：2/2

- A. SqlSession
- B. SqlSessionFactory
- C. MySqlSession
- D. Resources

学生答案：C  
参考答案：C

7. 【单选题】 dataSource的类型是POOLED时，下列哪个选项是它的常用属性：（）

得分：2/2

- A. configuration
- B. environment
- C. poolMaximumActiveConnections
- D. method

学生答案：C  
参考答案：C

8. 【单选题】 下面是更新数据mapper映射文件的部分代码,有关说法错误的是：（）

得分：2/2

```
update docs set dtitle=#{dtitle,jdbcType=VARCHAR}, ddesc=#{ddesc,jdbcType=VARCHAR}, ..... where dc  
ode=#{dcode}
```

- A. 在更新语句中需要的参数比较多，因此需要把数据封装在一个实体对象中
- B. 这串代码是写在mybatis-config.xml配置文件中
- C. 参数类型使用parameterType属性定义，这里它的值是一个实体类
- D. 占位符名称与实体属性名称保持一致，MyBatis能够根据名称进行绑定和赋值

学生答案：B  
参考答案：B

9. 【单选题】 下列对于缓存的理解正确的是：（）

得分：2/2

- A. MyBatis本地缓存分为一级缓存和二级缓存。一级缓存（PerpetualCache）是与SqlSession对象绑定的
- B. 二级缓存是跨多个会话的，MyBatis内置了一个二级缓存（PerpetualCache），可以直接使用
- C. : MyBatis默认是开启二级缓存的，可以通过全局参数cacheEnabled进行开关
- D. 以上描述都对

学生答案：D  
参考答案：D

10. 【单选题】 数据源的核心是数据库连接池，dataSource的类型有三种，下面有关说法错误的是：（）

得分：2/2

- A. UNPOOLED类型说明数据库连接对象不采用池化技术，每次数据库会话操作都需要新建和关闭数据库连接
- B. POOLED类型采用数据库连接池
- C. JNDI类型说明数据库连接池不由MyBatis来控制管理，交给中间件服务器（如Tomcat）管理
- D. 数据库连接对象池化并不能提高系统效率，只是对连接对象的初始化、新增、释放等操作进行统一管理

学生答案：D  
参考答案：D

11. 【单选题】 MyBatis trim标签使用中，对下面属性说明有误的是：（）

得分：2/2

- A. prefix:前缀要替换的内容
- B. suffix:后缀要替换的内容
- C. prefixOverride:前缀要替换的内容
- D. suffixOverride:后缀要判断的内容

学生答案：C  
参考答案：C

12. 【单选题】 阅读以下代码，理解正确的是：（）

得分：0/2

```
a.name=#{name}
```

- A. 上述代码缺少'prefixOverrides="and|or"'代码
- B. 这段代码的条件成立时，会产生where子句，并且会把紧跟where之后的and或者or删除
- C. name不为空时，会往SQL语句后面追加"group by a.name=#{name}"
- D. 代码编写有两处错误

学生答案 : B

参考答案 : C

13. 【单选题】 以下关于二级缓存补充说明有误的是 : ( ) 得分 : 2/2

- A. 映射语句文件中的所有select语句将会被缓存
- B. 映射语句文件中的所有insert, update和delete语句都会刷新缓存
- C. 缓存会使用Least Recently Used (LRU, 平常很少使用的) 算法来收回
- D. 上述都是错的

学生答案 : D

参考答案 : D

14. 【单选题】 有关属性文件的说法错误的是 : ( ) 得分 : 2/2

- A. 采用xml文件相比属性文件的好处是配置信息独立一份, 但是属性文件加密比xml文件更简单
- B. 属性文件的内容是以"="号进行分割的"名值对"
- C. 把数据源信息独立出来之后, 可以在mybatis-config.xml文件中利用配置读取属性文件
- D. 利用读取属性时, 可以通过\${\*\*\*}表达式语法读取属性内容

学生答案 : A

参考答案 : A

15. 【单选题】 关于标签说法有误的是 : ( ) 得分 : 2/2

- A. 主要用于动态SQL查询条件添加
- B. 元素会自动将其后第一个条件的and或or给忽略掉
- C. 元素主要用来简化SQL语句中where条件判断的书写
- D. 元素在内部标签没有返回内容时也会在动态语句上插入where条件语句

学生答案 : D

参考答案 : D

16. 【单选题】 Spring的环绕通知必须实现的接口是 ( ) 。 得分 : 2/2

- A. InvocationHandler
- B. MethodInterceptor
- C. MethodBeforeAdvice
- D. AfterReturningAdvice

学生答案 : B

参考答案 : B

17. 【单选题】 Spring的后置通知必须实现的接口是 ( ) 。 得分 : 0/2

- A. InvocationHandler
- B. MethodInterceptor
- C. MethodBeforeAdvice
- D. AfterReturningAdvice

学生答案 : D

参考答案 : A

18. 【单选题】 在Spring框架中, 面向方面编程 (AOP) 的目标在于 ( ) 。 得分 : 0/2

- A. 编写程序时无须关注其依赖组件的实现
- B. 封装JDBC访问数据库的代码, 简化数据访问层的重复性代码
- C. 将程序中涉及的公共问题集中解决
- D. 可以通过Web服务调用

学生答案 : C

参考答案 : A

19. 【单选题】 在Spring中, 下面的代码定义了一个前置通知类, 则下列选项中, 说法错误的是 ( ) 。 public class LogAdvice implements MethodBeforeAdvice{public void before(Method m, Object[] args, Object target) throws Throwable{System.out.println(m.getName()+"(" + Arrays.toString(args) + ")"; 得分 : 2/2

- A. 方法before是MethodBeforeAdvice接口中定义的方法
- B. 参数m是被通知的目标方法
- C. 参数args是调用方法的参数
- D. 参数target是代理类

学生答案 : D

参考答案 : D

20. 【单选题】在Spring中, 下列说法正确的是 ( ) 。 得分 : 0/2

- A. Spring DAO需要注入DataSource
- B. Spring DAO需要注入SessionFactory
- C. DataSource需要注入SessionFactory
- D. Spring DAO无须注入SessionFactory

学生答案 : A

参考答案 : B

21. 【多选题】在Spring中, 下列关于setter注入和构造注入的说法, 正确的有 ( ) 。 得分 : 0/4

- A. 执行效果完全相同
- B. 注入依赖关系注入时机不同
- C. setter注入方式依赖对象先注入
- D. 构造注入方式依赖对象先注入

学生答案 : ABD

参考答案 : CD

22. 【多选题】在Spring中, AOP将软件系统分为两个部分, 分别是( )。 得分 : 4/4

- A. 切面
- B. 业务处理
- C. 核心关注点
- D. 横切关注点

学生答案 : CD

参考答案 : CD

23. 【多选题】在Spring中, 实现AOP代理时, 下列说法正确的有 ( ) 。 得分 : 4/4

- A. Spring默认使用用于接口的JDK动态代理
- B. Spring只能使用JDK动态代理
- C. Spring也可以使用CGLIB代理
- D. Spring主要使用JDK动态代理

学生答案 : ACD

参考答案 : ACD

24. 【多选题】在Spring中, 使用Java实现代理方式需实现InvocationHandler接口, 且必须实现invoke(Object proxy, Method method, Object[] args)方法, 则下列说法正确的有 ( ) 。 得分 : 4/4

- A. 第一个参数是目标类
- B. 第二个参数Method由被代理接口的方法调用
- C. 第三个参数是方法调用的参数
- D. 当程序调用代理的目标方法时, 会自动变为调用invoke方法

学生答案 : BCD

参考答案 : BCD

25. 【多选题】在Spring中, 下列关于AOP的理解, 正确的有 ( ) 。 得分 : 4/4

- A. 面向纵向的
- B. 面向横向的
- C. AOP关注的是面
- D. AOP关注的是点

学生答案 : BC

参考答案 : BC

26. 【多选题】通过Spring构建Hibernate, 存在的优点有 ( ) 。 得分 : 0/4

- A. 测试简单
- B. 异常封装
- C. 通用的代码管理
- D. 综合的事务管理

学生答案 : ABCD

参考答案：ABD

27. 【多选题】在Spring包装Hibernate后，能够用于保存数据的方法有（）。

得分：0/4

- A.save
- B.load
- C.saveOrUpdate
- D.merge

学生答案：ACD

参考答案：ABC

28. 【多选题】Spring IOC容器管理Action的方式有（）。

得分：0/4

- A.使用org.springframework.web.struts.DelegatingRequestProcessor
- B.使用org.springframework.web.struts.DelegatingActionProxy
- C.使用org.springframework.web.context.ContextLoaderListener
- D.使用org.springframework.web.context.ContextLoaderServlet

学生答案：CD

参考答案：AB

29. 【多选题】在Spring中，关于IOC的理解，下列说法正确的有（）。

得分：4/4

- A.控制反转
- B.对象被动地接受依赖类
- C.对象主动地寻找依赖类
- D.一定要用接口

学生答案：AB

参考答案：AB

30. 【多选题】在Spring MVC中，以下实现拦截器的方式正确的有（）

得分：4/4

- A.实现 HandlerInterceptor 接口
- B.继承 HandlerInterceptorAdapter 方式定义拦截器
- C.实现Filter接口
- D.使用注解@Interceptor

学生答案：AB

参考答案：AB

31. 【简答题】什么是spring？

得分：0/5

学生答案：Spring 是一个广泛应用于 Java 开发的开源框架，旨在简化企业级应用的开发过程。

它提供了一系列的特性和功能，帮助开发者构建结构良好、易于维护和扩展的应用程序。

Spring 的核心特点包括：

控制反转 (IOC) / 依赖注入 (DI)：管理对象之间的依赖关系，将对象的创建和配置控制权交给框架，降低了组件之间的耦合度。

面向切面编程 (AOP)：能够将横切关注点（如日志、事务、安全等）从业务逻辑中分离出来，提高代码的模块化和可维护性。

方便的事务管理：提供了统一且灵活的事务处理机制，确保数据的一致性和可靠性。

丰富的模块和整合能力：涵盖了 Web 开发、数据访问、消息处理、安全等多个领域，并能与其他流行的框架和技术进行良好的整合。

总的来说，Spring 为 Java 开发者提供了一个强大而灵活的基础架构，加速了应用程序的开发和部署。

参考答案：Spring是一个轻量级Java开发框架，最早有Rod Johnson创建，是为了解决企业级应用开发的业务逻辑层和其他各层的耦合问题。它是一个分层的JavaSE/JavaEE full-stack（一站式）轻量级开源框架，为开发Java应用程序提供全面的基础架构支持。Spring负责基础架构，因此Java开发者可以专注于应用程序的开发。

32. 【简答题】使用 Spring 有哪些方式？

得分：5/5

学生答案：依赖注入 (DI)：通过配置文件（如 XML 配置或注解），让 Spring 容器自动为对象注入所需的依赖，减少对象之间的耦合。

面向切面编程 (AOP)：利用 Spring 的 AOP 功能，将横切关注点（如日志记录、事务处理、权限检查等）从业务逻辑中分离出来，提高代码的可维护性和可扩展性。

事务管理：Spring 提供了强大的事务管理机制，能够方便地对数据库操作等事务进行控制，确保数据的一致性和完整性。

数据访问：通过整合各种数据访问技术（如 JDBC、MyBatis、Hibernate 等），简化数据的读取和写入操作。

Web 应用开发：使用 Spring MVC 框架构建 Web 应用，处理请求、响应、视图渲染等。

配置管理：可以使用 XML 配置文件、Java 注解或者属性文件来定义 Spring 应用的各种配置信息，包括组件的创建、依赖关系、Bean 的属性等。

消息处理：借助 Spring 的消息机制，实现异步通信和系统集成。

测试支持：Spring 提供了丰富的测试工具和框架，便于对 Spring 应用进行单元测试和集成测试。

参考答案：作为一个成熟的 Spring Web 应用程序。作为第三方 Web 框架，使用 Spring Frameworks 中间层。作为企业级 Java Bean，它可以包装现有的 POJO (Plain Old Java Objects)。用于远程使用。

33. 【简答题】什么是Spring IOC 容器？

得分：5/5

学生答案：Spring IOC（控制反转）容器是 Spring 框架的核心组件之一。

它负责管理应用程序中的对象（称为 Beans）的创建、配置和生命周期。

在传统的编程方式中，对象的创建和依赖关系的管理通常由开发者手动完成。而在 Spring 中，这些工作由 IOC 容器接管。

开发者只需定义 Beans 的配置信息（如 Bean 的类、属性、依赖等），IOC 容器会根据这些配置自动创建对象，并将依赖对象注入到需要的地方。

IOC 容器使得对象之间的耦合度降低，提高了代码的可维护性和可测试性。同时，它还提供了一些生命周期管理的功能，如对象的初始化、销毁等方法的调用时机控制。常见的 IOC 容器实现有 BeanFactory 和 ApplicationContext 等。

参考答案：控制反转即IoC (Inversion of Control)，它把传统上由程序代码直接操控的对象的调用权交给容器，通过容器来实现对象组件的装配和管理。所谓的“控制反转”概念就是对组件对象控制权的转移，从程序代码本身转移到了外部容器。

Spring IOC 负责创建对象，管理对象（通过依赖注入（DI），装配对象，配置对象，并且管理这些对象的整个生命周期。

34. 【简答题】什么是bean的自动装配？

得分：5/5

学生答案：Bean 的自动装配是 Spring 框架中的一个特性。

它指的是 Spring 容器能够根据某些规则，自动将一个 Bean 所需的依赖 Bean 注入到该 Bean 中，而无需开发者显式地在配置文件或注解中指定每个依赖的注入。

自动装配可以基于类型、名称等方式进行。例如，如果一个 Bean 需要另一个特定类型的 Bean，Spring 会在容器中查找该类型的 Bean 并自动注入。

自动装配能够减少配置的工作量，提高开发效率，但在一些复杂的场景中，可能会导致依赖关系不够清晰，因此需要谨慎使用。

参考答案：在Spring框架中，在配置文件中设定bean的依赖关系是一个很好的机制，Spring 容器能够自动装配相互合作的bean，这意味着容器不需要和配置，能通过Bean工厂自动处理bean之间的协作。这意味着 Spring可以通过向Bean Factory中注入的方式自动搞定bean之间的依赖关系。自动装配可以设置在每个bean上，也可以设定在特定的bean上。