

考试剩余时间

30:00

我要交卷

请完成下列1~50道选择题，针对每一道题目，所有答案都选对，则该题得分，所选答案错误或不能选出所有答案，则该题不得分。每道题均为2分。

1

以下关于Spring中依赖注入的说法中错误的是（ ）。

(选择一项)

- A: 依赖注入使Bean与Bean之间以配置文件的形式组织在一起，而不是以硬编码的方式耦合在一起
- B: Spring提供了一个依赖注入的容器
- C: Spring的配置文件是完成组件组装的主要场所，常用节点包括<bean>及其子节点<bean-property>
- D: 通过依赖注入，Spring可以与Struts 2等框架实现集成

2

下面关于Spring说法不正确的是（ ）。

(选择一项)

- A: Spring是一个轻量级Java EE框架
- B: 开发中，Spring可以作为提供依赖注入的容器
- C: Spring的配置文件必须通过struts.xml通知应用服务器其所在的路径
- D: Spring实现了AOP思想

3

下面关于Spring的说法错误的是（ ）。

(选择一项)

- A: Spring核心机制是依赖注入
- B: Spring可以和Struts、Hibernate无缝地集成
- C: Spring是一个轻量级的框架，可以减少代码的污染
- D: Spring取代了如Struts、Hibernate等技术

4

关于Spring说法错误的是（ ）。

(选择二项)

- A: Spring是一个重量级的框架
- B: Spring是一个轻量级的框架

- C: Spring是一个低耦合的框架
 - D: Spring是一个入侵式的框架
-

5

Spring的核心机制是“依赖注入”，以下选项说法正确的是（ ）。

(选择二项)

- A: Spring提倡面向接口编程，因此依赖注入就是面向接口编程，它将最终替代面向对象编程
 - B: 依赖注入有时也被称作控制反转
 - C: 依赖注入可以让Bean与Bean之间以配置文件组织在一起，而不是以硬编码的方式耦合在一起
 - D: 在Spring中，一个角色需要另一个角色，需要先由程序自身来创建被调用者角色，然后由Spring容器注入到调用者角色中
-

6

以下关于依赖注入的说法正确的是（ ）。

(选择一项)

- A: 依赖注入提倡“面向接口”编程，所以开发中必须使用接口
 - B: 依赖注入的目标是在代码之外管理程序组件间的依赖关系
 - C: 依赖注入是面向对象技术的替代品，代表着更先进的思想
 - D: 依赖注入增大了项目的复杂度，不适合用于大型软件的开发
-

7

使用Spring可以整合JDBC、MyBatis、Hibernate,这三种数据库访问方式中，（ ）方式最适合跨不同类型的数据库。

(选择一项)

- A: JDBC
 - B: MyBatis
 - C: Hibernate
 - D: 三种数据库访问方式都适合
-

8

以下选项对Spring的核心机制说法错误的是（ ）。

(选择一项)

- A: Spring的核心机制是“依赖注入”
- B: “依赖注入”又称为“控制反转”

- C: Spring对Bean的管理是以配置文件的形式组织在一起，而不是以硬编码的方式耦合在一起
- D: 在Spring 里，创建被调用者实例的工作并不是由Spring 容器来完成，而是在调用者中使用new的方式实现
-

9

以下选项中，对于依赖注入的说法错误的是（ ）。

(选择一项)

- A: 依赖注入也称控制反转
- B: 依赖注入是面向对象技术的替代品
- C: 依赖注入让Bean与Bean之间以配置文件的形式组织在一起，而不是以硬编码的方式耦合在一起
- D: 在Spring里，创建被调用者实例的工作通常由Spring容器来完成，然后注入给调用者
-

10

关于Spring容器中的bean，下列说法错误的是（ ）。

(选择一项)

- A: Spring中的属性注入，需要在被注入的类中有相应的setter方法
- B: bean的初始化工作由Spring的容器来完成
- C: 在Spring容器中，bean的实例化不需要使用new关键字来实现
- D: <bean>节点的class属性，不需要指明类所在包名
-

11

下面关于Spring说法不正确的是（ ）。

(选择一项)

- A: 开发中，Spring可以作为提供依赖注入的容器
- B: Spring的配置文件必须通过struts.xml通知应用服务器其所在的路径
- C: Spring是一个轻量级JavaEE框架
- D: Spring实现了AOP思想
-

12

在Spring中，下列关于Spring配置文件的说法不正确的是（ ）。

(选择一项)

- A: Spring默认是读取/WEB-INF/applicationContext.xml配置文件
- B: Spring的配置文件可以配置在类路径下，并可以重命名，但是需要在web.xml文件 中指定
- C: 把applicationContext.xml文件放到src目录下，Spring也可以读到
- D: 可以通过在web.xml中的<context-param>、<param-name>和<param-value>进行指定，<context-param>

D: 可以通过在web.xml中的<context-param>param-name/和<param-value>进行指定 Spring配置文件

13

在Spring中，在Spring配置文件di.xml中包含如下的代码 `<bean id="test" class="Test"> <property name="i" value="100" /> </bean>` 由此可以推断出（ ）。

(选择一项)

A: 可以通过如下代码获取Test的实例 `ApplicationContext context=new ClassPathXmlApplicationContext("di.xml"); Test test=(Test)context.getBean("test");`

B: 可以通过如下代码获取i的值 `ApplicationContext context=new ClassPathXmlApplicationContext("di.xml"); int i = (int)context.getBean("i");`

C: Test肯定实现了一个接口

D: Test中一定存在getI()方法

14

在Spring的配置文件中，下列关于<property>标签的value属性和ref属性的作用描述正确的是（ ）。

(选择一项)

A: value属性和ref属性都是用来注入基本数据类型或字符串的值

B: value属性用来注入基本数据类型或字符串的值，ref属性用来注入已经定义好的bean

C: value属性用来注入已经定义好的bean，ref属性用来注入基本数据类型或字符串的值

D: value属性和ref属性都是用来注入已经定义好的bean

15

Spring框架的主要两大核心是（ ）。

(选择二项)

A: IoC

B: DAO

C: ORM

D: AOP

16

下面哪些不是MyBatis的核心对象（ ）。

(选择一项)

A: SqlSessionFactoryBuilder

B: SqlSessionFactory

C: SqlSession

C: sqlSession

D: SqlServer

17

在常用ORM框架技术中，以下描述正确的是（ ）。

(选择一项)

A: Hibernate框架封装过于臃肿，很多功能都用处不大，被大多数开发者摒弃

B: MyBatis框架非常灵巧，适用于任何场景的开发

C: MyBatis中所有sql语句与具体使用的数据库无关，移植性好

D: Hibernate和MyBatis各有千秋，在不同的使用场景，需要合理选择

18

在处理数据库中的数据时，Mybatis、Hibernate和JDBC那种技术的效率最高

(选择一项)

A: JDBC

B: Hibernate

C: Mybatis

D: 效率相同

19

在常用ORM框架技术中，以下描述错误的是（ ）。

(选择一项)

A: Hibernate和MyBatis都是orm对象关系映射框架，都是用于将数据持久化的框架技术

B: 相比于MyBatis，Hibernate封装更彻底，可以节省相对更多的开发工作量

C: MyBatis更灵巧，学习成本要低于Hibernate

D: 在对性能要求更高的项目中Hibernate比MyBatis更有优势

20

在Java中，为了实现面向接口编程，一般需要编写工厂类来提供实现接口类的具体对象，下列（ ）的功能类似于工厂类的功能。

(选择一项)

A: Hibernate的二级缓存

B: Struts2的拦截器

C: Spring的IoC容器

D: Hibernate的HibernateTemplate类

21

关于Hibernate与MyBatis说法错误的是（ ）。

(选择一项)

- A: 相对MyBatis, Hibernate对ORM的实现更加完善
- B: Hibernate不需要关注Sql的生成与结果的映射, MyBatis不需要写Sql, 但是需要关注结果的映射
- C: MyBatis直接使用sql, 移植性差
- D: MyBatis相对于Hibernate, 使用灵活性更高

22

下面关于Spring的核心概念说法错误的是（ ）。

(选择一项)

- A: Spring包括控制反转和面向切面编程两个核心概念
- B: 控制反转是把组件内部对象的创建控制权转移到了组件外部
- C: AOP是用于替代OOP的一种设计思想
- D: Spring AOP是针对方法级别的连接点模型

23

在Spring中提供的通过Web容器来启动Spring框架的类有（ ）。

(选择一项)

- A: ContextLoaderListener
- B: ServletLoaderListener
- C: ServletContextListener
- D: ActionServlet

24

下面对Spring中Bean的注入说法正确的是（ ）。

(选择一项)

- A: Bean根据注入方式不同可以分为构造方法注入和自动装载
- B: 使用构造方法注入构造对象的同时, 完成依赖关系的建立
- C: 在关系的对象很多时, 使用构造方法注入更适合
- D: 构造方法通过constructor-index属性来指定, 在该标签下要指定索引的位置

25

在动态SQL中，以下说法正确的是（ ）。

(选择一项)

- A: 利用if实现简单的查询判断
 - B: where不能简化SQL语句中条件判断
 - C: choose相当于Java中switch语句，通常与when和otherwise搭配
 - D: foreach迭代一个集合，通常用于where条件
-

26

以下（ ）不是Spring容器的作用。

(选择一项)

- A: 实例化类，创建类的对象
 - B: 建立各个对象间的关系
 - C: 管理对象的生命周期
 - D: 把容器中的对象全部实现单例模式
-

27

在Spring配置文件中有如下代码段，下面的说法正确的是（ ）。

```
<bean id="student" class="com.jbit.Student">
    <property name="name" value="tom"/>
    <property name="age" value="21"/>
</bean>
```

(选择一项)

- A: 在类Student中需要有public void setname(String name)方法
 - B: 在类Student中需要有private void setage(int age)方法
 - C: 在类Student中需要有public void setName(String name)方法
 - D: 在类Student中需要有private void setAge(int age)方法
-

28

下列关于Spring的配置文件中<property>标签及其属性描述正确的是（ ）。

(选择二项)

- A: 通过<property>标签可以指定调用相应的setter访问器实现依赖注入
- B: 与<property name="who">对应的是，类中必须有一个名为who的属性
- C: value属性用来注入已经定义好的bean，ref属性用来注入基本类型或字符串的值
- D: value属性用来注入基本类型或字符串的值，ref属性用来注入已经定义好的bean

29

关于Spring的依赖注入，下列说法中正确的是（ ）。

(选择二项)

- A: 使用Spring依赖注入的各组件之间没有依赖关系
- B: 依赖注入提倡面向接口编程
- C: 通过依赖注入，开发中可以先定义组件接口，然后独立开发各个组件，最后再根据组件间的依赖关系进行组装
- D: 组件的组装通过硬编码的方式实现

30

关于MyBatis和Hibernate的区别，以下选项描述错误的是（ ）。

(选择一项)

- A: MyBatis可以进行更加细致的SQL优化，可以减少查询字段
- B: MyBatis入门相对简单并且延续了SQL的使用经验
- C: HiBernate功能强大，数据无关性更有优势，程序员可以更专心进行业务处理
- D: Hibernate支持二级缓存而MyBatis不支持

31

下列有关MyBatis中动态SQL的说法中错误的是（ ）。

(选择二项)

- A: where元素可以动态创建where子句，且编写时无需考虑以and开头的情况
- B: 可以使用if元素和else元素实现条件选择
- C: when元素的test属性实现条件判断，其语法格式是test="\${表达式}"
- D: set元素可以为update语句动态创建set子句

32

在动态SQL中，以下说法错误的是（ ）。

(选择一项)

- A: 利用if实现简单的条件选择
- B: where简化SQL语句中where的条件判断
- C: trim可以灵活地去除多余的关键字
- D: set解决动态插入语句

33

在Spring配置文件中有如下代码段，下面的说法正确的是（ ）。

```
<bean id="Employee" class="com.jbit.entity.Employee">
    <property name="name" value="tom"/>
    <property name="age" value="21"/>
</bean>
```

(选择一项)

- A: 在类Employee中需要有代码public void setName()方法
- B: 在类Employee中需要有代码private void setage()方法
- C: 在类Employee中需要有代码public void setName(String name)方法
- D: 在类Employee中需要有代码private void setAge(int age)方法

34

对于src源文件目录下的applicationContext.xml,初始化Spring容器错误的是（ ）。

(选择一项)

- A: `ApplicationContext ctx = new ClassPathXmlApplicationContext(new String[]{"applicationContext.xml"});`
- B: `ApplicationContext ctx = new ClassPathXmlApplicationContext("applicationContext.xml");`
- C: `ApplicationContext ctx = new ClassPathXmlApplicationContext("file:applicationContext.xml");`
- D: `ApplicationContext ctx = new FileSystemXmlApplicationContext("src/applicationContext.xml");`

35

在web.xml文件中实现Spring容器初始化任务的类是（ ）。

(选择一项)

- A: Bean
- B: ContextLoaderListener
- C: ApplicationContext
- D: SessionFactory

36

在一个运行结果正确的测试类中，有如下代码：

```
ApplicationContext context = new
    ClassPathXmlApplicationContext("applicationContext.xml");
UserService userService = (UserService)context.getBean("userService");
```

下列选项由关于Spring的配置文件理解错误的是（ ）。

17. 以下关于Spring的配置项注册错误的是（ ）。

(选择一项)

- A: Spring的配置文件名字为applicationContext.xml
- B: 在Spring配置文件中一般使用bean元素中的id="userService"来声明程序中调用的userService
- C: ClassPathXmlApplicationContext可以直接替换为FileSystemXmlApplicationContext
- D: 在配置文件中UserService可以没有属性注入

37

在工程的classpath根路径下有Spring配置文件applicationContext.xml，其部分内容如下：

```
<bean id="user1" class="accp.User">
    <property name="username" value="Rod Johnson" />
</bean>
```

则以下说法中错误的是（ ）。

(选择二项)

- A: 在程序中可以通过代码：ApplicationContext ctx = new ClassPathXmlApplicationContext("applicationContext.xml"); 加载该配置文件
- B: 在程序中得到id为用户1的Bean的代码可以是
accp.User user1 = ctx.getBean("user1");
- C: 创建ClassPathXmlApplicationContext类的实例时，Spring中配置的Bean就会被实例化
- D: 在程序中每次调用ApplicationContext的getBean()方法获取id为用户1的Bean时，都会创建一个新的accp.User类型的对象

38

在MyBatis应用中，当单表数据量很小的情况下，可以使用Mybatis的RowBounds实现内存分页，以下选项说法正确的是（ ）。

(选择一项)

- A: 方法调用getSqlSession().selectList("selectUser", null, new RowBounds(offset, limit));
其中“selectUser”是Mapper文件中的SQL key，offset表示当前页数，limit表示当前页显示数据条数，除此之外，还需要在SQL语句最后中增加：
LIMIT #{offset},#{limit}
- B: 方法调用getSqlSession().selectList("selectUser", null, new RowBounds(offset, limit));
其中“selectUser”是Mapper文件中的SQL key，offset表示当前页数，limit表示当前页显示数据条数，由于是在内存分页，因此SQL语句不用任何分页代码
- C: 只需要在SQL语句末尾增加LIMIT #{offset},#{limit}即可
- D: 以上说法都不对

39

```

<select id="findDept" parameterType="hashmap" resultType="com.Dept">
  Select * from(
    Select d.*,rownum rn from(
      Select * from dept d order by d.deptname desc
    )d where rownum <#{endRow}
  )where rn>= #{startRow}
</select>

```

在MyBatis的SQL映射文件中的分页SQL映射的使用，下面说法正确的是（ ）。

(选择二项)

- A: 查询方法的入参是一个Map类型的参数
- B: 传入参数中至少包含endRow和startRow两个参数
- C: parameterType不能使用hashmap
- D: 该SQL查询返回的结果是集合，所以resultType应该写List

40

在MyBatis中，需要大量使用映射文件的标签，以下关于MyBatis常用标签的说法中错误的选项是（ ）。

(选择一项)

- A: where标签用来简化sql语句中where的条件判断
- B: set标签使用在更新语句中
- C: if标签用来实现条件判断
- D: choose(if,else)标签类似于java的switch语句，通常与if和else搭配

41

若类的成员变量为who，但是setter方法名为setSomebody(),在<property>标记里面name应写成（ ）。

(选择一项)

- A: who
- B: Who
- C: somebody
- D: Somebody

42

在Spring的配置文件di.xml中，包含如下代码：

```

<bean id="testBean" class="test.TestBean">
  <property name="dp" value="10"/>
  <property name="sp" value="mm"/>
</bean>

```

则下列说法正确的是（ ）。

(选择二项)

- A: TestBean中一定有属性: `private String sp=""`;
- B: TestBean中需要有方法: `public void setSp(String value)`
- C: TestBean中如果有属性dp, 其类型可能是int也可能是String
- D: 可以通过下列代码获取对象:
- ```
ApplicationContext context=new ClassPathXmlApplicationContext("di.xml");
TestBean t=(TestBean)context.getBean("test.TestBean");
```
- 

43

在Spring配置文件applicationContext.xml有如下内容:

```
<bean id="user1" class="accp.User">

</bean>
<bean id="group1" class="accp.Group">

</bean>
```

Test.java类中有代码如下:

```
ApplicationContext ctx
= new ClassPathXmlApplicationContext("applicationContext*.xml");
```

以下说法错误的是（ ）。

(选择二项)

- A: 该代码无法成功加载applicationContext.xml文件, 只能加载applicationContext\*.xml这个文件
- B: ClassPathXmlApplicationContext创建时, 配置的user1和group1都会被实例化
- C: 当前配置所示的所有Bean都将只生成一个实例(即单例模式)
- D: 取得id为用户1的bean的代码可以是
- ```
User user1 = ctx.getBean("user1");
```
-

44

使用MyBatis时, 有如下代码:

```
<select id="findActiveBlogWithTitleLike"
parameterType="Blog" resultType="Blog">
SELECT * FROM BLOG
WHERE state = 'ACTIVE'
-----
AND title like 'MyBatis'
</if>
</select>
```

这段配置文件需要判断“title”字段是否为空, 则可以填写在横线处的代码是（ ）

(选择一项)

- A: `<if test="title" != null>`

```
B: <if test="title != null">

C: <if title != null>

D: <if title ="test" != null">
```

45

在Spring配置文件applicationContext.xml有如下内容：

```
<bean id="user1" class="accp.User">
    ... ..
</bean>

<bean id="add1" class="accp.Address">
    ... ..
</bean>
```

Test.java类中有代码如下：

```
ApplicationContext ctx
= new ClassPathXmlApplicationContext("applicationContext.xml");
```

以下说法错误的是（ ）。

(选择一项)

- A: ApplicationContext是一个接口，ClassPathXmlApplicationContext是其实现类
 - B: ClassPathXmlApplicationContext创建时，当前配置的两个Bean都会被实例化
 - C: 当前配置所示的所有Bean都将只生成一个实例（即单例模式）
 - D: Test.java中得到id为用户1的bean的代码可以是
User user1 = ctx.getBean("user1");
-

46

下列说法错误的是（ ）。

(选择一项)

- A: 控制反转和依赖注入是同一个概念。
 - B: 依赖注入的作用是避免组件之间以硬码的方式耦合在一起。
 - C: 依赖注入可以增加模块的重用性灵活性。
 - D: Spring依赖注入只有设值注入和构造注入两种方式
-

47

在MyBatis中，以下选项中写法有误的是（ ）。

(选择多项)

```
A: <select id="queryUser" resultType="User" parameterType="User">
    SELECT * FROM tb_users t
    <where>
        <if test="searchFlag != null and searchFlag == 'name'">
            + user.name = #{user.name}
```

```

        t.user_name = #{userName}
    </if>
    <else>
        t.login_name = #{userName}
    </else>
</where>
</select>
B: <select id="queryUser" resultType="User" parameterType="User">
    SELECT * FROM tb_users t
    <where>
        <choose>
            <when test="searchFlag != null and searchFlag == 'name'" >
                t.user_name = #{userName}
            </when>
            <otherwise>
                t.login_name = #{userName}
            </otherwise>
        </choose>
    </where>
</select>
C: <select id="queryUser" resultType="User" parameterType="User">
    SELECT * FROM tb_users t
    <where>
        <if test="userAge != null and userAge != ''" >
            t.user_age < #{userAge}
        </if>
    </where>
</select>
D: <select id="queryUser" resultType="User" parameterType="string">
    SELECT * FROM tb_users t
    <where>
        <if test="userAge != null and userAge != ''" >
            t.user_id IN (#{ids})
        </if>
    </where>
</select>

```

48

Spring配置文件中有如下配置：

```

<bean id="user" class="User">
    <property name="name" value="James"/>
    <property name="age" value="30"/>
</bean>

```

由此配置信息可以推断必定错误的一项是（ ）。

(选择一项)

A: User是接口

B: User中有public void setAge(int age)方法

C: User中有public String getName()方法

D: User中有public void setName(String name)方法

49

在Spring配置文件applicationContext.xml有如下内容:

```
<bean id="user1" class="accp.User">
    ...
</bean>
```

Test.java类中有代码如下:

```
ApplicationContext ctx
    = new ClassPathXmlApplicationContext("applicationContext.xml");
```

则以下说法中错误的是()。

(选择二项)

A: 创建ClassPathXmlApplicationContext类的对象时, Spring中配置的单例模式的Bean就会被实例化

B: 默认情况下, Spring中配置的Bean都将只产生一个实例

C: 在Test.java中得到id为用户1的Bean的代码可以是

```
accp.User user1 = ctx.getBean("user1");
```

D: 在Test.java中多次调用ctx对象的getBean()方法获取id为用户1的Bean, 每次调用都会创建一个新的accp.User类型的对象

50

存在如下配置:

```
<select id="sqlLog" parameterType="appLog" resultType=" appLog ">
    SELECT * FROM t_applog WHERE
    <if test="title != null">
        title = #{title}
    </if>
    <if test="content != null">
        AND content = #{content}
    </if>
</select>
```

以下选项说明错误的是()。

(选择一项)

A: 这个配置存在问题, 当第一个if语句不成立时WHERE后面紧跟AND是一个错误的SQL将会出现运行时异常

B: 为了解决title为null时的运行时异常我们可以在WHERE后面增加1=1

C: 为了解决title为null时的运行时异常我们可以用<where>标签把所有<if>包含起来

D: 为了解决title为null时的运行时异常我们可以用<trim prefix="where" prefixOverrides="and |or"> </trim>把所有<if>包含起来, 并把WHERE去掉
