

Java 初中级面试题

一. 请说出作用域 `public`, `private`, `protected`, 以及不写时的区别

回答: 这四个作用域的可见范围如下表所示。

说明: 如果在修饰的元素上面没有写任何访问修饰符, 则表示 `friendly`。

作用域 当前类 同一包 (`package`) 子孙类 其他包 (`package`)

<code>public</code>	✓	✓	✓	✓
<code>protected</code>	✓	✓	✓	×
<code>friendly</code>	✓	✓	×	×
<code>private</code>	✓	×	×	×

二. 接口和抽象类

接口 (`interface`) 可以说成是抽象类的一种特例, 接口中的所有方法都必须是抽象的。接口中的方法定义默认为 `public abstract` 类型, **接口中的成员变量类型默认为 `public static final`** (这里需要说一点, 既然一个变量被 `final` 修饰了, 那么这个变量就是一个常量!!!!!! 变量必须初始化成常量!!!!!!)

下面比较一下两者的语法区别:

1. 抽象类可以有构造方法, 接口中不能有构造方法。
2. 抽象类中可以有普通成员变量, 接口中没有普通成员变量!!!!!! (注意重点在 普通 即 非静态 和 变量!!!!)
3. 抽象类中可以包含非抽象的普通方法, 接口中的所有方法必须都是抽象的, 不能有非抽象的普通方法。

4. 抽象类中的抽象方法的访问类型可以是 `public`, `protected` 和 (默认类型, 虽然 `eclipse` 下不报错, 但应该也不行), 但接口中的抽象方法只能是 `public` 类型的, 并且默认即为 `public abstract` 类型。

三. `String`、`StringBuffer` 与 `StringBuilder` 的区别

`String` 字符串常量

`StringBuffer` 字符串变量 (线程安全)

`StringBuilder` 字符串变量 (非线程安全)

四. 数组有没有 `length()` 这个方法? `String` 有没有 `length()` 这个方法?

答: 数组没有 `length()` 这个方法, 有 `length` 的属性。`String` 有 `length()` 这个方法。

五. 多线程部分

1. 什么是多线程?

在一个应用程序中, 同时, 有多个不同的执行路径。

2. 说一下多线程的好处?

提供程序效率。

3. 线程和进程有什么区别?

线程是进程的一条执行路径, 而进程是线程的集合。

4. 什么是线程同步、异步?

线程同步表示, 当前线程执行完后下一个线程接着执行。

线程异步表示, 在一个应用程序中, 同时, 有多个不同的执行路径。例如

`javaweb ajax android handler`

5. 线程之间如何同步

线程之间同步使用 **synchronized**、wait 与 notify

6. 什么是线程不安全？如何解决？（重点）

就是在多个线程共享同一个数据会受到其他线程的干扰。如何解决：使用线程同步技术，用上锁(synchronized)。让一个线程执行完了，在让另一个线程执行。

7. 如何创建一个线程？有几种方法？

继承 thread 类，重写 run 方法、实现 Runnable 接口，重新 run 方法，启动一个线程用 start()；

8. 是使用 Runnable 接口好？还是继承 Thread 类好？

是实现 Runnable 接口好，因为实现的接口还可以继续继承。如果继承了 Thread 类不能在继承。

9. sleep() 和 wait() 有什么区别？

a、sleep 是让当前线程指定休眠时间，然后继续工作 不释放锁

b、让当前线程 wait 则是等待，直到有线程通知 notify() 唤醒他才会重新工作。释放锁

六. 集合部分

1. List 和 Set、Map 区别？

Java 中的集合包括三大类，它们是 Set、List 和 Map，它们都处于 java.util 包中，Set、List 和 Map 都是接口，它们有各自的实现类。Set 的实现类主要有 HashSet 和 TreeSet，List 的实现类主要有 ArrayList，Map 的实现类主要有 HashMap 和 TreeMap。

Set 中的对象不按特定方式排序，并且没有重复对象。但它的有些实现类能对集合中的对象按特定方式排序，例如 TreeSet 类，它可以按照默认排序，也可以通过实现 java.util.Comparator 接口来自定义排序方式。

List 中的对象按照索引位置排序，可以有重复对象，允许按照对象在集合中的索引位置检索对象，如通过 list.get(i) 方式来获得 List 集合中的元素。

Map 中的每一个元素包含一个键对象和值对象，它们成对出现。键对象不能重复，值对象可以重复。

2. List、Map、Set 三个接口，存取元素时，各有什么特点？

list：存储： 有序的 可重复的

访问：可以 for 循环，foreach 循环，iterator 迭代器 迭代。

set：存储：无序的 不重复的

访问：可以 foreach 循环，iterator 迭代器 迭代

map：存储：存储的是一对一对的映射 ” key=value “，key 值 是无序，不重复的。value 值可重复

访问：可以map 中 key 值转为为 set 存储，然后迭代这个 set，用 map.get(key) 获取 value

也可以 转换为 entry 对象 用迭代器迭代

3. 说出 ArrayList, Vector, LinkedList 的存储性能和特性

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差，而

LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

4. hashmap 和 hashtable

1. hashMap 去掉了 Hashtable 的 contains 方法，但是加上了 containsValue () 和 containsKey () 方法。
2. hashtable 同步的，而 HashMap 是非同步的，效率上逼 hashtable 要高。
3. hashMap 允许空键值，而 hashtable 不允许。

5. currenthashmap

七. 说一下 java 反射机制以及常用场景。

JAVA 反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意方法和属性；这种动态获取信息以及动态调用对象方法的功能称为 java 语言的反射机制。反射机制的应用，比如 spring 框架，逆向代码，例如反编译

八. 框架部分

1. 说说 Spring?

Spring 是一个轻量级的控制反转 (IoC) 和面向切面 (AOP) 的容器框架。

IOC (Inversion of Control) 控制反转：将类的创建和依赖关系写在 spring 的配置文件中，由配置文件在程序运行的时候动态注入，这样就实现轻耦合。

AOP (Aspect Oriented Programming) 面向切面编程：将安全、事务、log 等公共的服务对于程序逻辑相对独立的功能抽取出来，利用 spring 的配置文件将这些功能在程序运行的时候动态的插进去，让程序员更专注于业务逻辑的实现，实现了按照方面编程，提高了可维护性和复用性

2. 说 SpringIOC、SpringAOP 原理?

SpringIOC，控制反转是一种编程思想，而依赖注入是控制反转在 spring 在具体实现。利用了工厂模式 将对象交给容器管理，你只需要在 spring 配置文件总配置相应的 bean，以及设置相关的属性，让 spring 容器来生成类的实例对象以及管理对象。在 spring 容器启动的时候，spring 会把你在配置文件中配置的 bean 都初始化好，然后在你需要调用的时候，就把它已经初始化好的那些 bean 分配给你需要调用这些 bean 的类（假设这个类名是 A），分配的方法就是调用

A 的 setter 方法来注入，而不需要你在 A 里面 new 这些 bean 了。主要涉及到 java 的工厂涉及模式和反射机制。

SpringAOP 面向切面的编程，或 AOP，是一种编程技术，允许程序模块化横向切割关注点，或横切典型的责任划分，如日志和事务管理。SpringAop 就是用 Java 的动态代理。主要用 java 的代理模式和反射机制。

3. Spring 的底层实现机制是什么？

使用 Demo4j（解析 XML）+Java 反射机制

Demo4j 其实就是解析 XML。使用反射机制实例化 bean。

4. Spring 注入有那些方式？

Set 注入

构造器注入

静态工厂的方法注入

实例工厂的方法注入

Spring 有那些注解？

@Autowired(按类型注入)

@Service(标示为注入为服务层)

@Transactional 事物注解

@Controller(标识控制器 bean id)

@RequestMapping（表示映射 URL 路径）

简述 Spring 的优缺点？

Spring 的优点？？

1. 降低了组件之间的耦合性，实现了软件各层之间的解耦
2. 可以使用容易提供的众多服务，如事务管理，消息服务等
3. 容器提供单例模式支持
4. 容器提供了 AOP 技术，利用它很容易实现如权限拦截，运行期监控等

功能

5. 容器提供了众多的辅助类，能加快应用的开发
6. spring 对于主流的应用框架提供了集成支持，如 hibernate, JPA,

Struts 等

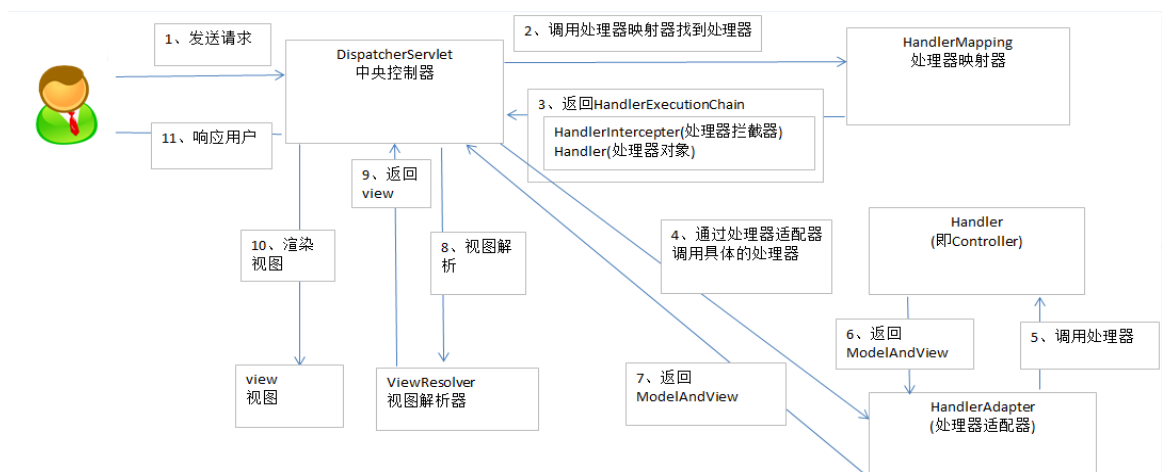
7. spring 属于低侵入式设计，代码的污染极低
8. 独立于各种应用服务器
9. spring 的 DI 机制降低了业务对象替换的复杂性
10. Spring 的高度开放性，并不强制应用完全依赖于 Spring，开发者可

以自由选择 spring 的部分或全部

缺点：

使用到了大量反射机制。反射机制非常占内存，

11. Springmvc 简单的执行流程



12.mybatis #{}和\${}区别, 那种可以防止 sql 注入。

13 说一下事物以及 spring 事物的配置。

九. 什么是单例? 单例有那些写法?

单例分类: 懒汉式单例、饿汉式单例

单例模式有以下特点:

- 1、单例类只能有一个实例。
- 2、单例类必须自己创建自己的唯一实例。
- 3、单例类必须给所有其他对象提供这一实例。

①懒汉式单例

```
1. //懒汉式单例类. 在第一次调用的时候实例化自己
2. public class Singleton {
3.     private Singleton() {}
4.     private static Singleton single=null;
5.     //静态工厂方法
6.     public static Singleton getInstance () {
7.         if (single == null) {
8.             single = new Singleton();
9.         }
10.        return single;
11.    }
12. }
```


②饿汉式单例

13. //饿汉式单例类. 在类初始化时, 已经自行实例化

14. public class Singleton1 {

15. private Singleton1() {}

16. private static final Singleton1 single =
new Singleton1();

17. //静态工厂方法

18. public static Singleton1 getInstance() {

19. return single;

20. }

21. }

懒汉式与饿汉式区别?

从名字上来说, 饿汉和懒汉,

饿汉就是类一旦加载, 就把单例初始化完成, 保证 `getInstance` 的时候, 单例是已经存在的了,

而懒汉比较懒, 只有当调用 `getInstance` 的时候, 才回去初始化这个单例。

另外从以下两点再区分以下这两种方式:

1、线程安全:

饿汉式天生就是线程安全的, 可以直接用于多线程而不会出现问题,

懒汉式本身是非线程安全的，为了实现线程安全有几种写法，分别是上面的

1、2、3，这三种实现在资源加载和性能方面有些区别。

2、资源加载和性能：

饿汉式在类创建的同时就实例化一个静态对象出来，不管之后会不会使用这个单例，都会占据一定的内存，但是相应的，在第一次调用时速度也会更快，因为其资源已经初始化完成，

而懒汉式顾名思义，会延迟加载，在第一次使用该单例的时候才会实例化对象出来，第一次调用时要做初始化，如果要做的工作比较多，性能上会有些延迟，之后就和饿汉式一样了。

十. 数据库面试题

1. 基本 sql 增删改查，左右连接、内连接、order by 、group by

2. 索引。

3. 存储过程。

4. 触发器。

5. 锁：共享锁、排他锁

6. mysql 引擎。

十一. Js

1. ajax 的基本参数

type: 'GET', // 这是请求的方式 可以是 GET 方式也可以是 POST 方式，默认是 GET 。

url: 'xxx', // 这是请求的连接地址 一般情况下这个地址是后台给前端的一个连接，直接写就可以 。

`dataType: 'json',` // 这是后台返回的数据类型 一般情况下都是一个 json 数据, 前端遍历一下就 OK 。

`async: true,` // 默认为 true, 默认为 true 时, 所有请求均为异步请求, 如果需要发送同步请求, 需设置为 false,

```
data: {  
  
    // 要传递的参数  
  
    'xxx' : 'xxx',  
},
```

2. 如何解决跨域问题

Jsonp, 缺点, 只能 get 请求。

以下简单了解

十二. Cookie 和 session 区别

cookie 和 session 的区别:

1、cookie 数据存放在客户的浏览器上, session 数据放在服务器上。

2、cookie 不是很安全, 别人可以分析存放在本地的 COOKIE 并进行 COOKIE 欺骗

考虑到安全应当使用 session。

3、session 会在一定时间内保存在服务器上。当访问增多, 会比较占用你服务器的性能

考虑到减轻服务器性能方面, 应当使用 COOKIE。

4、单个 cookie 保存的数据不能超过 4K, 很多浏览器都限制一个站点最多保存 20 个 cookie。

5、所以个人建议：

将登陆信息等重要信息存放为 SESSION

其他信息如果需要保留，可以放在 COOKIE 中