



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

第三版：Java 55 道

列出 5 个应该遵循的 JDBC 最佳实践

有很多的最佳实践，你可以根据你的喜好来例举。下面是一些更通用的原则：

- 1、使用批量的操作来插入和更新数据
- 2、使用 PreparedStatement 来避免 SQL 异常，并提高性能。
- 3、使用数据库连接池
- 4、通过列名来获取结果集，不要使用列的下标来获取。

说出几条 Java 中方法重载的最佳实践？

下面有几条可以遵循的方法重载的最佳实践来避免造成自动装箱的混乱。

- a) 不要重载这样的方法：一个方法接收 int 参数，而另一个方法接收 Integer 参数。
- b) 不要重载参数数量一致，而只是参数顺序不同的方法。
- c) 如果重载的方法参数个数多于 5 个，采用可变参数。

在多线程环境下，SimpleDateFormat 是线程安全的吗？

不是，非常不幸，DateFormat 的所有实现，包括 SimpleDateFormat 都不是线程安全的，因此你不应该在多线程程序中使用，除非是在对外线程安全的环境中使用，如将 SimpleDateFormat 限制在 ThreadLocal 中。如果你不这么做，

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



在解析或者格式化日期的时候，可能会获取到一个不正确的结果。因此，从日期、时间处理的所有实践来说，我强烈推荐 `joda-time` 库。

Java 中如何格式化一个日期？如格式化为 `ddMMyyyy` 的形式？

<http://javarevisited.blogspot.com/2011/09/convert-date-to-string-simple-dateformat.html>

Java 中，可以使用 `SimpleDateFormat` 类或者 `joda-time` 库来格式日期。`DateFormat` 类允许你使用多种流行的格式来格式化日期。参见中的示例代码，代码中演示了将日期格式化成不同的格式，如 `dd-MM-yyyy` 或 `ddMMyyyy`。

Java 中，怎么在格式化的日期中显示时区？

<http://java67.blogspot.sg/2013/01/how-to-format-date-in-java-simpledateformat-example.html>

Java 中 `java.util.Date` 与 `java.sql.Date` 有什么区别？

<http://java67.blogspot.sg/2014/02/how-to-convert-javautildate-to-javasqldate-example.html>

Java 中，如何计算两个日期之间的差距？

程序

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

<http://javarevisited.blogspot.sg/2015/07/how-to-find-number-of-days-between-two-dates-in-java.html>

Java 中，如何将字符串 YYYYMMDD 转换为日期？

<http://java67.blogspot.sg/2014/12/string-to-date-example-in-java-multithreading.html>

如何测试静态方法？ ()

可以使用 PowerMock 库来测试静态方法。

怎么利用 JUnit 来测试一个方法的异常？

<http://javarevisited.blogspot.sg/2013/04/JUnit-tutorial-example-test-exception-thrown-by-java-method.html>

@Before 和 @BeforeClass 有什么区别？

<http://javarevisited.blogspot.sg/2013/04/JUnit-tutorial-example-test-exception-thrown-by-java-method.html>

怎么检查一个字符串只包含数字？ 解决方案

<http://java67.blogspot.com/2014/01/java-regular-expression-to-check-numbers-in-String.html>

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

在不使用 StringBuffer 的前提下，怎么反转一个字符串？

解决方案

<http://java67.blogspot.com/2012/12/how-to-reverse-string-in-java-string-buffer-stringbuilder.htm>

Java 中，怎么获取一个文件中单词出现的最高频率？

解决方案

<http://java67.blogspot.com/2015/10/java-program-to-find-repeated-words-and-count.html>

如何检查出两个给定的字符串是反序的？

解决方案

<http://javarevisited.blogspot.sg/2013/03/Anagram-how-to-check-if-two-strings-are-anagrams-example-tutorial.html>

Java 中，怎么打印出一个字符串的所有排列？

解决方案

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

<http://javarevisited.blogspot.com/2015/08/how-to-find-all-permutations-of-string-java-example.html>

Java 中，怎样才能打印出数组中的重复元素？

解决方案

<http://javarevisited.blogspot.com/2015/06/3-ways-to-find-duplicate-elements-in-array-java.html>

Java 中如何将字符串转换为整数？

```
String s="123";
```

```
int i;
```

第一种方法：i=Integer.parseInt(s);

第二种方法：i=Integer.valueOf(s).intValue();

在没有使用临时变量的情况如何交换两个整数变量的值？

解决方案

https://blog.csdn.net/zidane_2014/article/details/34180223

关于 OOP 和设计模式的面试题

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

这部分包含 Java 面试过程中关于 SOLID 的设计原则，OOP 基础，如类，对象，接口，继承，多态，封装，抽象以及更高级的一些概念，如组合、聚合及关联。也包含了 GOF 设计模式的问题。

接口是什么？为什么要使用接口而不是直接使用具体类？

接口用于定义 API。它定义了类必须得遵循的规则。同时，它提供了一种抽象，因为客户端只使用接口，这样可以有多重实现，如 List 接口，你可以使用可随机访问的 ArrayList，也可以使用方便插入和删除的 LinkedList。接口中不允许写代码，以此来保证抽象，但是 Java 8 中你可以在接口声明静态的默认方法，这种方法是具体的。

Java 中，抽象类与接口之间有什么不同？

Java 中，抽象类和接口有很多不同之处，但是最重要的一个是 Java 中限制一个类只能继承一个类，但是可以实现多个接口。抽象类可以很好的定义一个家族类的默认行为，而接口能更好的定义类型，有助于后面实现多态机制。

除了单例模式，你在生产环境中还用过什么设计模式？

这需要根据你的经验来回答。一般情况下，你可以说依赖注入，工厂模式，装饰模式或者观察者模式，随意选择你使用过的一种即可。不过你要准备回答接下的基于你选择的模式的问题。

你能解释一下里氏替换原则吗？

https://blog.csdn.net/pu_xubo565599455/article/details/51488323

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

什么情况下会违反迪米特法则？为什么会有这个问题？

迪米特法则建议“只和朋友说话，不要陌生人说话”，以此来减少类之间的耦合。

适配器模式是什么？什么时候使用？

适配器模式提供对接口的转换。如果你的客户端使用某些接口，但是你有另外一些接口，你就可以写一个适配去来连接这些接口。

什么是“依赖注入”和“控制反转”？为什么有人使用？

控制反转（IOC）是 Spring 框架的核心思想，用我自己的话说，就是你要做一件事，别自己可劲 new 了，你就说你要干啥，然后外包出去就好~

依赖注入（DI）在我浅薄的想法中，就是通过接口的引用和构造方法的表达，将一些事情整好了反过来传给需要用到的地方~

抽象类是什么？它与接口有什么区别？你为什么使用过抽象

类？

接口用于规范，抽象类用于共性、

声明方法的存在而不去实现它的类被叫做抽象类

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。

构造器注入和 setter 依赖注入，那种方式更好？

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

每种方式都有它的缺点和优点。构造器注入保证所有的注入都被初始化，但是 setter 注入提供更好的灵活性来设置可选依赖。如果使用 XML 来描述依赖，Setter 注入的可读写会更强。经验法则是强制依赖使用构造器注入，可选依赖使用 setter 注入。

依赖注入和工程模式之间有什么不同？

虽然两种模式都是将对象的创建从应用的逻辑中分离，但是依赖注入比工程模式更清晰。通过依赖注入，你的类就是 POJO，它只知道依赖而不关心它们怎么获取。使用工厂模式，你的类需要通过工厂来获取依赖。因此，使用 DI 会比使用工厂模式更容易测试。

适配器模式和装饰器模式有什么区别？

虽然适配器模式和装饰器模式的结构类似，但是每种模式的出现意图不同。适配器模式被用于桥接两个接口，而装饰模式的目的是在不修改类的情况下给类增加新的功能。

适配器模式和代理模式之前有什么不同？

这个问题与前面的类似，适配器模式和代理模式的区别在于他们的意图不同。由于适配器模式和代理模式都是封装真正执行动作的类，因此结构是一致的，但是适配器模式用于接口之间的转换，而代理模式则是增加一个额外的中间层，以便支持分配、控制或智能访问。

什么是模板方法模式？

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

模板方法提供算法的框架，你可以自己去配置或定义步骤。例如，你可以将排序算法看做是一个模板。它定义了排序的步骤，但是具体的比较，可以使用 Comparable 或者其语言中类似东西，具体策略由你去配置。列出算法概要的方法就是众所周知的模板方法。

什么时候使用访问者模式？

访问者模式用于解决在类的继承层次上增加操作，但是不直接与之关联。这种模式采用双派发的形式来增加中间层。

什么时候使用组合模式？

组合模式使用树结构来展示部分与整体继承关系。它允许客户端采用统一的形式来对待单个对象和对象容器。当你想要展示对象这种部分与整体的继承关系时采用组合模式。

继承和组合之间有什么不同？

虽然两种都可以实现代码复用，但是组合比继承更灵活，因为组合允许你在运行时选择不同的实现。用组合实现的代码也比继承测试起来更加简单。

描述 Java 中的重载和重写？

重载和重写都允许你用相同的名称来实现不同的功能，但是重载是编译时活动，而重写是运行时活动。你可以在同一个类中重载方法，但是只能在子类中重写方法。重写必须要有继承。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

Java 中，嵌套公共静态类与顶级类有什么不同？

类的内部可以有多个嵌套公共静态类，但是一个 Java 源文件只能有一个顶级公共类，并且顶级公共类的名称与源文件名称必须一致。

OOP 中的 组合、聚合和关联有什么区别？

如果两个对象彼此有关系，就说他们是彼此相关联的。组合和聚合是面向对象中的两种形式的关联。组合是一种比聚合更强大的关联。组合中，一个对象是另一个的拥有者，而聚合则是指一个对象使用另一个对象。如果对象 A 是由对象 B 组合的，则 A 不存在的话，B 一定不存在，但是如果 A 对象聚合了一个对象 B，则即使 A 不存在了，B 也可以单独存在。

给我一个符合开闭原则的设计模式的例子？

开闭原则要求你的代码对扩展开放，对修改关闭。这个意思就是说，如果你想增加一个新的功能，你可以很容易的在不改变已测试过的代码的前提下增加新的代码。有好几个设计模式是基于开闭原则的，如策略模式，如果你需要一个新的策略，只需要实现接口，增加配置，不需要改变核心逻辑。一个正在工作的例子是 Collections.sort() 方法，这就是基于策略模式，遵循开闭原则的，你不需为新的对象修改 sort() 方法，你需要做的仅仅是实现你自己的 Comparator 接口。

抽象工厂模式和原型模式之间的区别？

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

抽象工厂模式：通常由工厂方法模式来实现。但一个工厂中往往含有多个工厂方法生成一系列的产品。这个模式强调的是客户代码一次保证只使用一个系列的产品。当要切换为另一个系列的产品，换一个工厂类即可。

原型模式：工厂方法的最大缺点就是，对应一个继承体系的产品类，要有一个同样复杂的工厂类的继承体系。我们可以把工厂类中的工厂方法放到产品类自身之中吗？如果这样的话，就可以将两个继承体系为一个。这也就是原型模式的思想，原型模式中的工厂方法为 clone，它会返回一个拷贝（可以是浅拷贝，也可以是深拷贝，由设计者决定）。为了保证用户代码中到时可以通过指针调用 clone 来动态绑定地生成所需的具体的类。这些原型对象必须事先构造好。

原型模式想对工厂方法模式的另一个好处是，拷贝的效率一般对构造的效率要高。

什么时候使用享元模式？

享元模式通过共享对象来避免创建太多的对象。为了使用享元模式，你需要确保你的对象是不可变的，这样你才能安全的共享。JDK 中 String 池、Integer 池以及 Long 池都是很好的使用了享元模式的例子。

Java 面试中其他各式各样的问题

这部分包含 Java 中关于 XML 的面试题，正则表达式面试题，Java 错误和异常及序列化面试题

嵌套静态类与顶级类有什么区别？

一个公共的顶级类的源文件名称与类名相同，而嵌套静态类没有这个要求。一个嵌套类位于顶级类内部，需要使用顶级类的名称来引用嵌套静态类，如

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

HashMap.Entry 是一个嵌套静态类，HashMap 是一个顶级类，Entry 是一个嵌套静态类。

你能写出一个正则表达式来判断一个字符串是否是一个数字吗？

一个数字字符串，只能包含数字，如 0 到 9 以及 +、- 开头，通过这个消息，你可以下一个如下的正则表达式来判断给定的字符串是不是数字。

首先要 import java.util.regex.Pattern 和 java.util.regex.Matcher

```
public boolean isNumeric(String str){
    Pattern pattern = Pattern.compile("[0-9]*");
    Matcher isNum = pattern.matcher(str);
    if( !isNum.matches() ){
        return false;
    }
    return true;
}
```

Java 中，受检查异常 和 不受检查异常的区别？

受检查异常编译器在编译期间检查。对于这种异常，方法强制处理或者通过 throws 子句声明。其中一种情况是 Exception 的子类但不是 RuntimeException 的子类。非受检查是 RuntimeException 的子类，在编译阶段不受编译器的检查。

Java 中，throw 和 throws 有什么区别

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

throw 用于抛出 `java.lang.Throwable` 类的一个实例化对象,意思是说你可以通过关键字 `throw` 抛出一个 `Error` 或者 一个 `Exception`, 如:

```
throw new IllegalArgumentException("size must be multiple of 2")
```

而 `throws` 的作用是作为方法声明和签名的一部分,方法被抛出相应的异常以便调用者能处理。Java 中,任何未处理的受检查异常强制在 `throws` 子句中声明。

Java 中, `Serializable` 与 `Externalizable` 的区别?

`Serializable` 接口是一个序列化 Java 类的接口,以便于它们可以在网络上传输或者可以将它们的状态保存在磁盘上,是 JVM 内嵌的默认序列化方式,成本高、脆弱而且不安全。`Externalizable` 允许你控制整个序列化过程,指定特定的二进制格式,增加安全机制。

Java 中, `DOM` 和 `SAX` 解析器有什么不同?

`DOM` 解析器将整个 XML 文档加载到内存来创建一棵 `DOM` 模型树,这样可以更快的查找节点和修改 XML 结构,而 `SAX` 解析器是一个基于事件的解析器,不会将整个 XML 文档加载到内存。由于这个原因,`DOM` 比 `SAX` 更快,也要求更多的内存,不适合于解析大 XML 文件。

说出 `JDK 1.7` 中的三个新特性?

虽然 `JDK 1.7` 不像 `JDK 5` 和 `8` 一样的大版本,但是,还是有很多新的特性,如 `try-with-resource` 语句,这样你在使用流或者资源的时候,就不需要手动关闭,Java 会自动关闭。`Fork-Join` 池某种程度上实现 Java 版的 `Map-reduce`。允许 `Switch` 中有 `String` 变量和文本。菱形操作符(`<>`)用于类型推断,不再需

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

要在变量声明的右边申明泛型，因此可以写出可读写更强、更简洁的代码。另一个值得一提的特性是改善异常处理，如允许在同一个 catch 块中捕获多个异常。

说出 5 个 JDK 1.8 引入的新特性?

- 1、Java 8 在 Java 历史上是一个开创新的版本，下面 JDK 8 中 5 个主要的特性：
- 2、Lambda 表达式，允许像对象一样传递匿名函数
- 3、Stream API，充分利用现代多核 CPU，可以写出很简洁的代码
- 4、Date 与 Time API，最终，有一个稳定、简单的日期和时间库可供你使用
- 5、扩展方法，现在，接口中可以有静态、默认方法。
- 6、重复注解，现在你可以将相同的注解在同一类型上使用多次。

Java 中，Maven 和 ANT 有什么区别?

虽然两者都是构建工具，都用于创建 Java 应用，但是 Maven 做的事情更多，在基于“约定优于配置”的概念下，提供标准的 Java 项目结构，同时能为应用自动管理依赖（应用中所依赖的 JAR 文件），Maven 与 ANT 工具更多的不同之处请参见。

- 1、这就是所有的面试题，如此之多，是不是？我可以保证，如果你能回答列表中的所有问题，你就可以很轻松的应付任何核心 Java 或者高级 Java 面试。虽然，这里没有涵盖 Servlet、JSP、JSF、JPA、JMS、EJB 及其它 Java EE 技术，也没有包含主流的框架如 Spring MVC、Struts 2.0、Hibernate，也没有包含

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

SOAP 和 RESTful web service, 但是这份列表对做 Java 开发的、准备应聘 Java web 开发职位的人还是同样有用的, 因为所有的 Java 面试, 开始的问题都是 Java 基础和 JDK API 相关的。如果你认为我这里有任何应该在这份列表中而被我遗漏了的 Java 流行的问题, 你可以自由的给我建议。我的目的是从最近的面试中创建一份最新的、最优的 Java 面试问题列表。

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题