优秀程序员之路——Java开发经验及技巧大汇总

开发经验、技巧总结

Java 的开发总要经过立项——设计——编码——测试等诸多过程，下面先来介绍一下 Java 开发中的这些开发经验。

1. 立项

这个时候需要做的是需求的收集，开发人员需要重视需求。机会总是倾向有准备的人，成功自然也是这样。开发人员不但要做好需求分析，还要做出精确的估计。因为编码工作保质保量的按时完成需要多方的准备工作，技术难点需要进行充分的技术预言，不熟悉的依赖平台或类库要进行熟悉。

2. 设计

设计的时候要善用图，因为通过图可以将模块结构以及流程等很难用用文字描述的内容清晰的表达出来。

此外，设计的时候要耐得住详细设计过程中的繁琐、痛苦，要以快乐的心态面对。

3. 编码

编码工作需要保持细致、认真、严谨的工作的工作态度，做出的编码不能仓促、草率，要考虑周到。在逻辑、风格、简单有效等方面拿出你的最好，既能更好为公司实现价值，同时更有利自己在技能，岗位的进步。

4. 测试

编写的代码不能因为一时的疏漏而造成后期工作麻烦，开发人员都要将自己编写的代码详细的测试一下，验证无误再提交，这样才能成为一个优秀的开发人员。

5. 总结

在项目的整个流程过程中，要善于及时的沟通。遇到问题要及时找到解决的办法，这个时候可以和同事一起解决问题，学习他们的长处和优点，将其变为己有。

开发经验及技巧问题汇总

接下来我们来总结一些 Java 开发技巧的问题以供大家参考。

1. 什么是 JRE/J2RE？

答：J2RE 是 Java2RuntimeEnvironment，即 Java 运行环境，有时简称 JRE。如果你只需要运行 Java 程序或 Applet，下载并安装它即可。如果你要自行开发 Java 软件，请下载 JDK。在 JDK 中附带有 JRE。注意由于 Microsoft 对 Java 的支持不完全，请不要使用 IE 自带的虚拟机来运行 Applet，务必安装一个 JRE 或 JDK。

2. 什么是 J2SE/J2EE/J2ME？

答：J2SE 就是一般的 Java。J2ME 是针对嵌入式设备的，比如支持 Java 的手机，它有自己的 JRE 和 SDK。J2EE 是一组用于企业级程序开发的规范和类库，它使用 J2SE 的 JRE。

3. 怎么做可以把 java 编译成 exe 文件？

答：JDK 只能将 java 源文件编译为 class 文件。class 文件是一种跨平台的字节码，必须依赖平台相关的 JRE 来运行。Java 以此来实现跨平台性。有些开发工具可以将 java 文件编译为 exe 文件。作者反对这种做法，因为这样就取消了跨平台性。如果你确信你的软件只在 Windows 平台上运行，你可以考虑使用 C++/C#来编程。

4. 类名首字母应该大写。

字段、方法以及对象（句柄）的首字母应小写。对于所有标识符，其中包含的所有单词都应紧靠在一起，而且大写中间单词的首字母。

例如：

```
ThisIsAClassName
thisIsMethodOrFieldName
```

若在定义中出现了常数初始化字符，则大写 staticfinal 基本类型标识符中的所有字母。这样便可标志出它们属于编译期的常数。

Java 包（Package）属于一种特殊情况：它们全都是小写字母，即便中间的单词亦是如此。对于域名扩展名称，如 com，org，net 或者 edu 等，全部都应小写（这也是 Java1.1 和 Java1.2 的区别之一）。

5. package 是什么意思？怎么用？

答：为了唯一标识每个类并分组，java 使用了 package 的概念。每个类都有一个全名，例如 String 的全名是 java.lang.String，其中 java.lang 是包名，String 是短名。按照 java 命名惯例，包名是全部小写的，而类名的第一个字母是大写的。这样，如果你自行定义了同样名字的类 String，你可以把它放在 mypackage 中，通过使用全名 mypackage.String 和 java.lang.String 来区分这两个类。同时，将逻辑上相关的类放在同一个包中，可以使程序结构更为清楚。为了定义包，你要做的就是在 java 文件开头加一行“packagemypackage;”。注意包没有嵌套或包含关系，mypackage 包和 mypackage.mysubpackage 包对 JRE 来说是并列的两个包（虽然开发者可能暗示包含关系）。

6. 没有声明任何 package 会怎么样？

答：你的类被认为放在默认包中，这时全名和短名是一致的。

7. 对于自己创建的每一个类，都考虑置入一个 main()，其中包含了用于测试那个类的代码。

为使用一个项目中的类，我们没必要删除测试代码。若进行了任何形式的改动，可方便地返回测试。这些代码也可作为如何使用类的一个示例使用。

8. 使类尽可能短小精悍，而且只解决一个特定的问题。

下面是对类设计的一些建议。

⑴ 一个复杂的开关语句：考虑采用"多形"机制。

⑵ 数量众多的方法涉及到类型差别极大的操作：考虑用几个类来分别实现。

⑶ 许多成员变量在特征上有很大的差别：考虑使用几个类。

9. 在 javaxxx 的时候显示“Exceptioninthread"main"，java.lang.NoSuchMethodError:main”。

答：首先，在你的程序中每个 java 文件有且只能有一个 public 类，这个类的类名必须和文件名的大小写完全一样。其次，在你要运行的类中有且只能有一个 publicstaticvoidmain (String[]args)方法，这个方法就是你的主程序。

10. java 里面怎么定义宏？

答：java 不支持宏，因为宏代换不能保证类型安全。如果你需要定义常量，可以将它定义为某个类的 staticfinal 成员。

11. 怎么判断要读的文件已经到了尽头？

答：在 Reader 的 read 方法中明确说明返回-1 表示流的结尾。

12. 用继承及方法覆盖来表示行为间的差异，而用字段表示状态间的区别。

一个非常极端的例子是通过对不同类的继承来表示颜色，这是绝对应该避免的：应直接使用一个"颜色"字段。

13. 阅读代码的时间比写代码的时间多得多，请多阅读代码。

思路清晰的设计可获得易于理解的程序，但注释、细致的解释以及一些示例往往具有不可估量的价值。无论对你自己，还是对后来的人，它们都是相当重要的。如对此仍有怀疑，那么请试想自己试图从联机 Java 文档里找出有用信息时碰到的挫折，这样或许能将你说服。

14. throw 和 throws 有什么不同？

答：throws 用于方法声明中，声明一个方法会抛出哪些异常。而 throw 是在方法体中实际执行抛出异常的动作。如果你在方法中 throw 一个异常，却没有在方法声明中声明之，编译器会报错。注意 Error 和 RuntimeException 的子类是例外，无需特别声明。

15. extends 和 implements 有什么不同？

答：对于 class 而言，extends 用于（单）继承一个类（class），而 implements 用于实现一个接口（interface）。interface 的引入是为了部分地提供多继承的功能。在 interface 中只需声明方法头，而将方法体留给实现的 class 来做。这些实现的 class 的实例完全可以当作 interface 的实例来对待。在 interface 之间也可以声明为 extends（多继承）的关系。注意一个 interface 可以 extends 多个其他 interface。

16. 良好的设计能带来最大的回报。

简言之，对于一个特定的问题，通常会花较长的时间才能找到一种最恰当的解决方案。但一旦找到了正确的方法，以后的工作就轻松多了，再也不用经历数小时、数天或者数月的

痛苦挣扎。我们的努力工作会带来最大的回报（甚至无可估量）。而且由于自己倾注了大量心血，最终获得一个出色的设计方案，成功的快感也是令人心动的。坚持抵制草草完工的诱惑——那样做往往得不偿失。

17. 应将方法设计成简要的、功能性单元，用它描述和实现一个不连续的类接口部分。

理想情况下，方法应简明扼要。若长度很大，可考虑通过某种方式将其分割成较短的几个方法。这样做也便于类内代码的重复使用（有些时候，方法必须非常大，但它们仍应只做同样的一件事情）。

18. java 里面能不能重载操作符？

答：不能。String 的+号是唯一一个内置的重载操作符。你可以通过定义接口和方法来实现类似功能。

19. 在子类的构造方法中如何调用父类的构造方法？

答：在子类构造方法的第一行调用 super(...)即可。

20. 尽量使用 interfaces，不要使用 abstract 类。

若已知某样东西准备成为一个基础类，那么第一个选择应是将其变成一个 interface（接口）。只有在不得不使用方法定义或者成员变量的时候，才需要将其变成一个 abstract（抽象）类。接口主要描述了客户希望做什么事情，而一个类则致力于（或允许）具体的实施细节。