旧版本可以从[www.MindView.net](http://www.MindView.net)中下载

# 第一章 对象导论

继承和组合：在建立新类时，应该首先考虑组合，因为它更加简单灵活。如果采用这种方式，设计会变得更加清晰。一旦有了一些经验之后，便能够看出必须使用继承的场合了。

使用父类引用保存子类对象，在调用方法时，在编译时是不知道要调用的代码是哪一段的，在运行时才绑定到指定代码的地址去执行，这就是多态。

# 第二章 一切都是对象

## 2.2.2 特例：基本类型

Java中的基本类型所占存储空间大小并不像其他大多数语言那样随机器硬件架构的变化而变化，这是Java程序比其他 大多数语言编写的程序更具可移植性的原因之一。

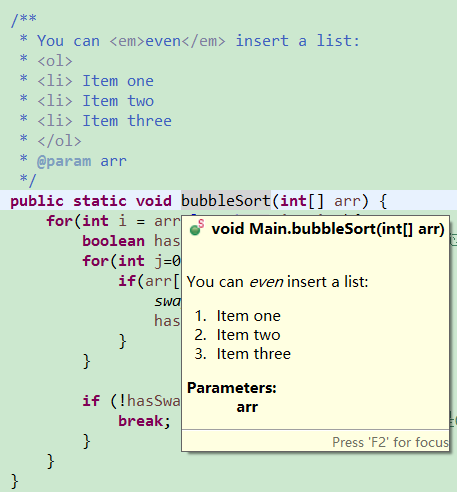
**高精度数字**

Java提供了两个用于高精度计算的类：BigInteger和BigDecimal。虽然它们大体上属于“包装器类”的范畴，但二者都没有对应的基本类型。它们像基本数据类型一样可以进行各种运算，但是这些运算只能通过其提供的方法了来完成，由于这么做复杂了许多，所以运算速度会比较慢，但是它却换来了精度。

BigInteger可以表示任何大小的整数值，而不会丢失任何信息。BigDecimal则可以表示任何精度的浮点数，可以用它进行精确的货币计算。

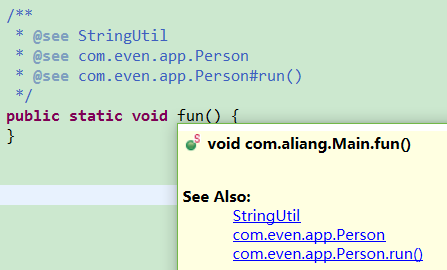
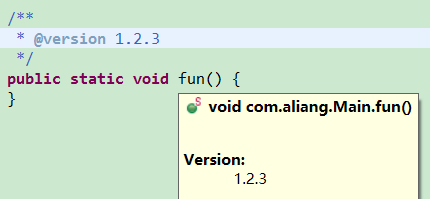
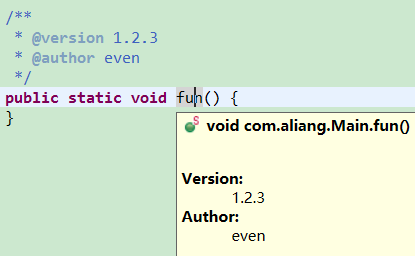
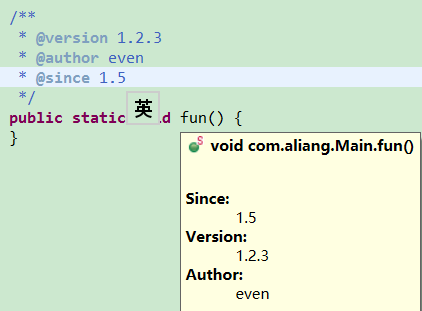
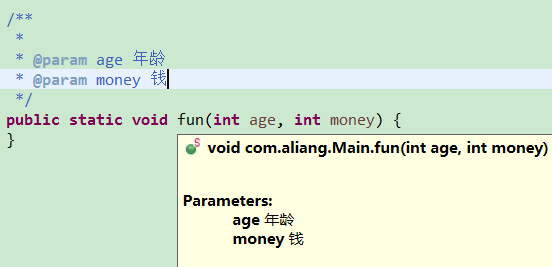
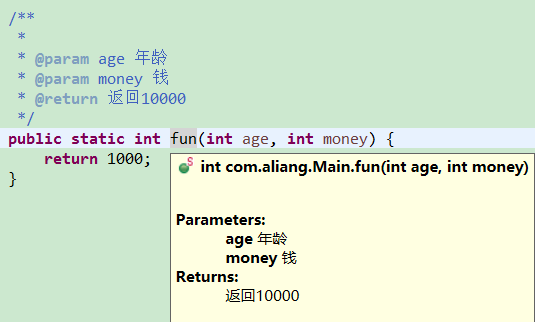
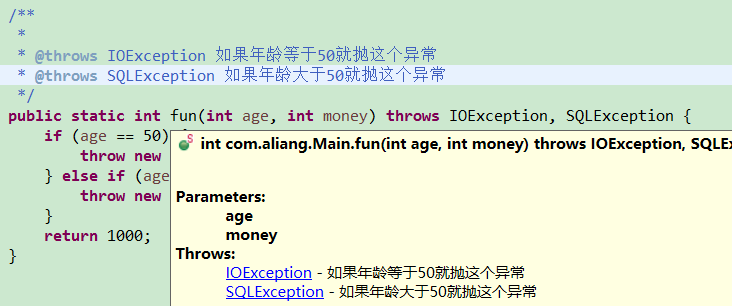
## 2.8 注释和嵌入式文档

可以像编写Html那样编写文档注释，如下：



注意：在文档注释中，位于每一行开头的星号和前导空格都会被javadoc丢弃。javadoc会对所有内容重新格式化，使其与标准的文档一致。不要使用标题标签，如<h1>或<hr>，因为javadoc会插入自己的标题，而你的标题可能同它们发生冲突。

### 2.8.4 一些标签示例

1. @see 标签允许用户引用其它类的文档：  
   @see classname  
   @see full-qualified-classname  
   @see full-qualified-classname#method-name  
     
   如果是同一个包名下的类可以不加包名。
2. {@link package.class#member label}  
     
   从这个例子可以看到，可以给see或者link的内容起一个别名，也可以不起。
3. {@docRoot} 该标签产生到文档根目录的相对路径，用于文档树页面的显示超链接。  
   
4. {@inheritDoc} 该标签从当前这个类的最直接的基类中继承相关文档到当前的文档注释中。
5. @version 如果javadoc命令使用了“-version”标记，那么就从生成的HTML文档别提取出版本信息。  
   
6. @author 可以是你的姓名或电子邮件或者其他任何适宜信息，如果javadoc命令使用了-author标记，那么就从生成的HTML文档中特别提取作者信息。  
   
7. @since 该标签允许你指定程序代码最早使用的版本，可以在HTML Java文档中看到它被用来指定此类是在哪个JDK版本时出现的  
   
8. @param parameter-name description  
   
9. @return  
   
10. @throws full-qualified-class-name description  
    
11. @deprecated 该标签用于指出一些旧特性已由改进的新特性所取代，建议用户不要再使用这些旧特性，因为在不久的将来它们很可能会被删除。如果使用该标记，会引起编译器发布警告。在Java SE5中，Javadoc标签@deprecated已经被@Deprecated注解所替代。  
    