

介绍Java创建成员内部类的注意事项计算机等级考试 PDF转换
可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E4_BB_8B_E7_BB_8DJava_c97_645223.htm

注意事项一：在内部类中可以随意使用外部类的成员方法以及成员变量。众所周知，在定义成员方法或者成员变量的时候，可以为其加上一些权限的修饰词，以防止其他类的访问。如在成员变量或者成员方法前面，加上Private关键字，则其他类就无法调用这个类中的成员方法或则和成员变量。但是，如果这个类有成员内部类，则不受这方面的限制。也就是说，在成员内部类中可以随意引用外部类的成员方法以及成员变量，即使这些类成员方法或者成员变量被修饰了private。如在成员外部类中定义了一个i变量，并且利用private关键字来修饰。此时在其他类中是不能够引用这个私有成员变量的。但是这个类的成员内部类的方法中，则不仅可以引用这个外部类的私有变量，而且还可以对其进行赋值等操作。这个赋值操作对于外部类也是有效的。即成员内部类可以更改外部类中私有变量的值。

注意事项二：要在外部类中实例化内部类对象的引用。如果一个类定义在另外一个类中，成为成员内部类，此时一定要注意，内部类的实例一定要绑定在外部类的实例上。也就是说，要从外部类中初始化一个内部类的对象，此时内部类的对象就会绑定在外部类的对象上。这跟普通的类有所不同。普通的类，创建完之后，不一定马上需要实例化。在需要用到这个对象的时候，再进行实例化即可。但是，如果一个类成为另外一个类的成员内部类，则就不同了。必须要在外部类中实例化内部类对象的引用，以实现将内部类的实例绑定在

外部类的实例上。简单的说，就是在定义外部类的时候，如果有成员内部类，那么就不要再望了在外类中利用new关键字来实例化内部类对象的引用。而对于外部类来说，则可以在需要的时候再进行实例化。如此就可以保证，利用外部类创建对象的同时创建了内部类的对象。从而可以保证内部类的实例绑定在外类类的实例上。 注意事项三：成员内部类中成员方法与成员变量的私有性。作为成员内部类，可以随意引用外部类中的成员变量与成员方法。那么在成员内部类中定义的成员变量，外部类是否也可以随意访问呢?答案是否定的。内部类可以访问它外部类的成员，但是内部类的成员(如成员变量或者成员方法)只有在内部类的范围之内是可知的，不能够被外部类直接引用。如现在在外类中定义了一个变量i，在内部类中定义了另一个变量ii。此时在成员内部类中，可以直接引用这个外部类中的变量i，也可以对其直接进行赋值，如i=5等等。但是在外类中，则不能够直接引用内部类中的成员变量。如在外类中，利用ii=5的赋值语句改变这个变量的值，就属于语法错误，在编译的时候就会出现错误。如果外部类真的要引用内部类的成员，那也不是不可以。只是不能够进行直接的引用，而是要使用内部类对象引用的方法才能够调用内部类的成员变量。这一点程序开发人员需要切记。成员内部类与外部类相互访问彼此的成员方法限制是不同的。特别需要注意的是，如果在外类和非静态方法之外实例化内部对象，则需要使用外部类.内部类的形式来制定这个对象的类型。这非常的麻烦。为此要尽量避免在外类和非静态方法之外实例化内部类对象。再者，内部类对象会依赖于外部类对象，除非已经存在一个外部类对象，否则类中

不会出现内部类对象。简单的说，内部类对象与外部类对象之间的关系非常的紧密。有时候即像一对父子(成员内部类可以随意使用外部类的成员)，有时候又像是陌生人(外部类不能够直接使用内部类中的成员)。作为一个出现的程序开发人员，必须要深入了解外部类对象与内部类对象的关系。因为在实际工作中，在外部类中定义成员内部类还是很常见的。只有了解他们彼此之间的关系，在编写应用程序中，才可以把控好他们。注意实现四：使用this关键字获取内部类与外部类对象的引用。在外部类和成员内部类中，都可以定义变量。成员内部类可以随意访问外部类中的变量，而外部类不能够直接访问内部类中的变量，只有通过使用内部类对象来引用内部类的成员变量。不过需要注意的是，在外部类中定义的变量与内部类中定义的变量名字可以相同。也就是说，在外部类中可以定义一个变量i，在内部类中也可以定义一个变量i。此时新的问题就出来了。由于在内部类中可以随意访问外部类中成员方法与成员变量。但是此时成员内部类与外部类的变量名相同，那么如果要在内部类使用这个变量i，编译器怎么知道现在需要调用外部类的变量i，还是内部类的变量i呢？在实际工作中，应该尽量避免这种情况。即在定义变量名字的时候，内部类中的变量名字尽量不要与外部类中的变量名字相同。但是如果真的发生这种情况的话，Java编译器还是有手段可以解决这个变量名字的冲突问题。如果在类中的成员内部类中遇到成员变量名字与外部类中的成员变量名字相同，则可以通过使用this关键字来加以区别。如现在有一个类名字叫做student。而在这个类中又创建了一个成员内部类，名字叫做age。现在在这两个类中各自定义了一个成员变量i，

用来做循环之用。此时如果在成员内部类中调用这个变量，该如何进行区分呢?通常情况下，如果使用this.i的方式来调用变量，则表示这个变量是成员内部类中定义的变量。而通过使用student.this.i的形式调用变量时表示此时引用的是外部类的成员变量。也就是说，如果内部类中引用外部类中同名的成员，需要通过外部类名字.this.外部类成员名字的方式加以引用。而且，在引用成员内部类自己的定义的成员时也需要使用this关键字，以加以区别。显然这非常的麻烦。为此笔者仍然要强调一遍，除非有非常特殊的必要，否则的话要尽量避两外部类与成员内部类中成员变量或者成员方法采用相同的名字。否则的话，会给后续的引用带来很大的麻烦。从以上的总结中可以看出，成员内部类与外部类之间的关系确实很复杂。作为一个Java程序开发人员，必须要了解内部类与外部类成员相互之间引用的准则、内部类对象与外部类对象的关系。在编写代码的时候，尽量避免内部类与外部类中成员变量与成员方法的命名冲突。如果真的发生重名的话，要学会利用this关键字来消除这个冲突。笔者上面谈的一些注意实现也是一些比较抽象的内容，需要开发人员在实际工作中去领悟。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com