

This关键字在Java语言中的应用Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/629/2021\\_2022\\_This\\_E5\\_85\\_B3\\_E9\\_94\\_AE\\_c104\\_629768.htm](https://www.100test.com/kao_ti2020/629/2021_2022_This_E5_85_B3_E9_94_AE_c104_629768.htm) 应用一：引用成员变量。

```
Public Class Student{ String name. //定义一个成员变量name  
private void SetName(String name){ //定义一个参数(局部变量)name this.name=name. //将局部变量的值传递给成员变量 } }
```

如上面这个代码中，有一个成员变量name。同时在方法中有一个形式参数，名字也是name。然后再方法中将形式参数name的值传递给成员变量name。虽然我们可以看明白这个代码的含义，但是作为Java编译器它是怎么判断的呢？到底是将形式参数name的值传递给成员变量name，还是反过来讲成员变量name的值传递给形式参数name呢？也就是说，两个变量名字如果相同的话，那么Java如何判断使用哪个变量？此时this这个关键字就起到作用了。This这个关键字其代表的就是对象中的成员变量或者方法。也就是说，如果在某个变量前面加上一个this关键字，其指的就是这个对象的成员变量或者方法，而不是指成员方法的形式参数或者局部变量。为此在上面这个代码中，this.name代表的就是对象中的成员变量，又叫做对象的属性。而后面的name则是方法的形式参数。代码this.name=name就是将形式参数的值传递给成员变量。这就是上面这个代码的具体含义。一般情况下，在Java语言中引用成员变量或者成员方法都是以对象名.成员变量或者对象名.成员方法的形式。不过有些程序员即使在没有相同变量的时候，也喜欢使用this.成员变量的形式来引用变量。这主要是从便于java认证更多详细资料代码的阅读考虑的。一看到这

个this关键字就知道现在引用的变量是成员变量或者成员方法，而不是局部变量。这无形中就提高了代码的阅读性。不过话说回来，这是this关键字在Java语言中的最简单的应用。从这个应用中，我们可以看出this关键字其代表的就是对象的名字。掌握这个基本的原理之后，再了解后续的一些高级应用，也就是水到渠成的事情了。笔者这里只是以一个形式参数为例。其实如果是局部变量的话，也是相同的道理。如在上面的代码中，name不是形式参数，而是一个局部变量。此时Java也会遇到相同的疑惑，即变量名name代表的到底是局部变量还是形式参数？name=name到底代表的是什么含义？根据局部变量的作用域，在方法内部，如果局部变量与成员变量同名的话，那么是以局部变量为准。可是在name=name这个赋值语句中，将局部变量的值赋值给自己，显然并不是很合适。根据代码的含义，本来的意思应该是将局部变量赋值给成员变量。为了更清晰的表达这个含义，为此最好采用如下的书写格式：this.name=name。这里的this关键字含义就是对象名student。为此this.name就表示student.name。这就转换为正规的成员变量应用了。

应用二：调用类的构造方法。在一个Java类中，其方法可以分为成员方法和构造方法两种。构造方法是一个与类同名的方法，在Java类中必须存在一个构造方法。如果在代码中没有显示的体现构造方法的话，那么编译器在编译的时候会自动添加一个没有形式参数的构造方法。这个构造方法跟普通的成员方法还是有很多不同的地方。如构造方法的话一律是没有返回值的。而且也不用void关键字来说明这个构造方法没有返回值。而普通的方法可以有返回值、也可以没有返回值，程序员可以根据自己的需要来定义

。不过如果普通的方法没有返回值的话，那么一定要在方法定义的时候采用void关键字来进行说明。其次构造方法的名字有严格的要求，即必须与类的名字相同。也就是说，Java编译器发现有个方法与类的名字相同才将其当作构造方法来对待。而对于普通方法的话，则要求不能够与类的名字相同，而且多个成员方法不能够采用相同的名字。在一个类中可以存在多个构造方法，这些构造方法都采用相同的名字。只是形式参数不同。Java语言就凭用户参数的参数来判断调用那个构造方法。this关键字除了可以调用成员变量之外，还可以调用构造方法。如下面的代码：

```
public class Student{ //定义一个类，类的名字为student。
public Student(){ //定义一个方法，名字与类相同故为构造方法
this(“luobo”) }
public Student(String name){ //定义一个带形式参数的构造方法 } }
```

在上面这个代码中，笔者定义了两个构造方法，一个带参数，另一个没有带参数。构造方法都不会有返回值，不过由于构造方法的特殊性，为此不必要在构造方法定义时带上void关键字来说明这个问题。在第一个没有带参数的构造方法中，笔者使用了this(“luobo”)这个代码。这个代码表示什么含义呢？在构造方法中使this关键字表示调用类中的构造方法。如果一个类中有多个构造方法，因为其名字都相同，跟类名一致，那么这个this到底是调用哪个构造方法呢？其实，这跟采用其他方法引用构造方法一样，都是通过形式参数来调用构造方法的。如上例中，this关键字后面加上了一个参数，那么就表示其引用的是带参数的构造方法。如果现在有三个构造方法，分别为不带参数、带一个参数、带两个参数。那么Java编译器会根据所传递的参数数量的不同，来判断该调用哪个

构造方法。从上面这个示例中可以看出，this关键字不仅可以用来引用成员变量，而且还可以用来引用构造方法。不过如果要使用这种方式来调用构造方法的话，有一个语法上的限制。一般来说，利用this关键字来调用构造方法，只有在无参数构造方法中第一句使用this调用有参数的构造方法。否则的话，翻译的时候，就会有错误信息。这跟引用成员变量不同。如果引用成员变量的话，this关键字是没有位置上的限制的。如果不熟悉这个限制的话，那么还是老老实实的采用传统的构造方法调用方式为好。虽然比较麻烦，但是至少不会出错。

应用三：返回对象的值。this关键字除了可以引用变量或者成员方法之外，还有一个重大的作用就是返回类的引用。如在代码中，可以使用return this，来返回某个类的引用。此时这个this关键字就代表类的名称。如这个代码在上面这个student类中，那么这个代码其代表的含义就是return student。可见，这个this关键字除了可以引用变量或者成员方法之外，还可以做为类的返回值。这才是this关键字最引人注意的地方。在使用this关键字的时候，需要注意一个细节问题。在笔者谈到的第一个案例中，利用this关键字来引用成员变量。即this.name=name。这个语句是将局部变量或则形式参数赋值给成员变量。其实，此时这个this关键字不加也可以起到类似的效果。即上面这个语句中，与name=name这个语句时相同的。即Java编译器会自动将第一个name当作成员变量来对待。而将等号右边的name变量当作形式参数。虽然这么书写代码，可以节省this这几个字符的输入。但是对于代码的阅读是不利的。为此，从代码的阅读性考虑，如果成员变量与形式参数或者局部变量相同的话，那么最好是使用this.成员变量(成

员方法)的方式来引用成员变量。虽然语法上没有错误，但是作为一个高级的JAVA程序员，在代码书写的规范上，也不能够忽视。 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](http://www.100test.com)