

# 变量的作用域

## 目 录

1. 概述.....	2
2. 作用域 .....	2



## 1. 概述

Java 中的变量可以在代码块中的任何地方声明。代码块以左大括号开始，以右大括号结束。块用来定义作用域，每次创建一个新块后，就会创建一个新的作用域。变量的作用域能确定可在程序中按变量名访问该变量的区域，还确定变量的生命周期。


## 2. 作用域

变量在作用域内为局部变量，所以在其作用域外不能使用。因此，在某个作用域内声明一个变量后，该变量就成为局部变量，以防擅自访问。

作用域可以嵌套。每次创建一个代码块后，就会创建一个新的嵌套作用域。外层作用域包含内层作用域。也就是内层作用域中的代码可使用外层作用域中声明的变量。反过来却不行，即外层作用域无法使用在内层作用域中声明的变量。

示例 1 讲解变量及其作用域。

### 示例 1:



```
/*
 * 北大青鸟
 * 版权所有
 */

/**
 * 这个类用于演示变量的作用域
 * @author jb
 */
public class ScopeVariable {

    /**
     * 这是 main 方法
     * @param args 传递至 main 方法的参数
     */
    public static void main(String[] args) {

        /* num 在内层作用域中可用 */
        int num = 2;

        /* 测试变量 num */
        if (num == 2) {
            int num1 = num * num;

            System.out.println("num 和 num1 的值分别为 " + num + "    " + num1);
        }
    }
}
```

```
        /* num1 = 2; 错误! num1 未知 */  
        System.out.println("num 的值为 " + num);  
    }  
}
```

在示例 1 中，变量 num 是在 main() 方法中声明的，因此在 main() 方法内的代码中都可以访问该变量。另一个变量 num1 是在 if 块内部声明的。因此只有 if 块中出现的代码才可使用 num1。为此，给 main() 方法块中的代码行 num1 = 2; 添加了注释。否则，编译器会生成错误。但是变量 num 可以在 if 块中使用，因为已经在 if 块外部声明了这个变量。

只要作用域中的代码开始执行，变量就存在于内存中。意思是变量超出作用域的范围后就会释放它的值。变量的生存期受到其作用域的限制。

如果在作用域中初始化一个变量，则每次调用块时系统会重新初始化该变量。

