说明：所谓魔法数值，是指在代码中直接出现的数值，只有在这个数值记述的那部分代码中才能明确了解其含义。

　　魔法数值使代码的可读性大大下降。而且，如果同样的数值多次出现时，到底这些数值是不是带有同样的含义呢，谁也说不清楚。另一方面，如果本来应该使用相同数值的地方，一旦用错了，也很难发现。因此，需要注意以下几点，极力避免使用魔法数值。

　　① 不使用魔法数值，代之以有名字的Static final或者enum值

　　在[**Java**](http://lib.csdn.net/base/java)语言中，对有名字的值的定义，可以使用Static final或者enum来声明的值

　　注意（命名方法）：在取名时一定要注意增加名字的信息量。比如，为16命名为SIXTEEN是没有意义的。

　　如果16表示的是价格表的一个元素，则应该命名为PRICE\_TABLE\_MAX.关于命名规则，还要参照本小册子的第6条。

　　② 原则上，数值就是魔法数值

　　但是，0，作为数组的最小下标是经常使用的。

　　例：for（ int i = 0； I < table.leERRORth（）； i++ ）

　　读到这里，读者都会认为这里的0是数组的最小下标，所以在这里不认为0是魔法数值。

　　另外，0经常被作为变量初始化的缺省值，这时候一般也不认为0是魔法数值。

　　通过使用有名字的值，一方面可以提高代码的可读性，另一方面，也可以把变更的地方局部化，从而提高可维护性。

　　例子：

　　魔法数字的例子

　　int priceTable[] = new int[16]； //ERROR：这个16究竟有何含义呢？

　　使用了带名字的数值的例子

　　static final int PRICE\_TABLE\_MAX = 16； //OK：带名字

　　int price Table[] = new int [PRICE\_TABLE\_MAX]； //OK：名字的含义是很清楚的