1 断言的概念
2 启用和禁用断言
3 使用断言

1 断言的概念

断言机制允许在测试期间向代码中插入一些检查，而在生产代码中会自动删除这些检查。使用`assert`关键字实现断言，这个关键字有两种形式：`assert condition;`和`assert condition : expression;`

这两个语句都会计算条件（condition），如果结果为`false`，则抛出一个`AssertionError`异常。在第二种形式中，表达式（expression）会传入`AssertionError`对象的构造器，并转换成一个消息字符串。

例如，要断言x是一个非负数，可以使用这条语句：

```
assert x >= 0;
```

或者将x的值传递给`AssertionError`对象，以便以后显示：

```
assert x >= 0 : x;
```

2 启用和禁用断言

默认情况下，断言是禁用的。启用或禁用断言是类加载器的功能。禁用断言时，类加载器会去除断言代码，因而不会降低程序运行的速度。

如果使用命令行运行程序，可以用`-ea`选项启用断言，用`-da`禁用断言。

也可以通过编程控制类加载器的断言状态，有关这方面的API展示如下：

```
/* java.lang.ClassLoader */
void setDefaultAssertionStatus(boolean b)
    // 为通过类加载器加载的类启用或禁用断言
void setClassAssertionStatus(String className, boolean b)
    // 为给定的类和它的内部类启用或禁用断言
void setPackageAssertionStatus(String packageName, boolean b)
    // 为给定包及其子包中的所有类启用或禁用断言
void clearAssertionStatus()
    // 删除所有显式的类和包断言状态设置，并禁用通过这个类加载器加载的所有类的断言。
```

3 使用断言

断言失败是致命的、不可恢复的错误，断言检查只在开发和测试阶段打开。因此，不应该使用断言向程序的其他部分通知发生了可恢复性的错误，不应该利用断言与程序用户沟通问题。断言只应该用于在测试阶段确定程序内部错误的位置。