

Java编程中的断言和时态逻辑计算机等级考试 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\\_Java\\_E7\\_BC\\_96\\_E7\\_A8\\_8B\\_c97\\_645402.htm](https://www.100test.com/kao_ti2020/645/2021_2022_Java_E7_BC_96_E7_A8_8B_c97_645402.htm) 虽然传统断言可以增加对 Java 代码执行的检查次数，但有许多检查不能用它们来执行。弥补这一缺陷的方法是使用“时态逻辑”，它是一种用于描述程序状态如何随时间而更改的形式体系。在本文中，Eric Allen 将讨论断言，介绍时态逻辑并描述用于处理程序中时态逻辑断言的工具。我们大家同意对 Java 代码检查得越多就越好，我们检查了断言在测试新的和改进的编程中的用法。虽然传统断言可以增加执行的检查次数，但有许多检查不能用它们来执行。然而，有一个方法可以弥补断言留下的检查缺口。那就是使用时态逻辑。时态逻辑是用于描述程序状态如何随时间而更改的形式体系。让我们讨论一下断言及其特性，以及时态逻辑是如何适合检查的。然后，我们将研究用于处理时态逻辑断言的工具。断言及其特性除了类型检查和单元测试外，断言还提供了一种确定各种特性是否在程序中得到维护的极好方法。让我们快速浏览三种类型常见的断言特性（虽然是常见的，但它们没有提供我们所需的完整范围），将它们与可以用传统断言语言表示的程序特性的类型进行比较，并检查多线程上下文所必需的，但不可能表示成常规断言的断言特性。我们还将提供一些代码示例。常见的断言特性传统上，断言特性分成下面三种类型：?代码块执行之前特性所持有的条件前断言。?代码块执行之后特性所持有的条件后断言。?代码块执行之前和之后特性所持有的不变断言。与这些典型形式的断言一样有用，它们不太会有我们

希望能在程序中持有的所有特性范围。让我们看一下典型的用断言表示的程序特性。可表示为断言的程序特性这只是可以用传统断言语言表示的程序特性类型的简短列表——所有程序员都希望在代码中包含的特性：

- 确保任何一次性特性都仅生成一次
- 断言文档决不被未授权的代理程序访问
- 断言向每个线程提供运行机会
- 断言系统将决不会使其本身陷入死锁

安全性协议使用一次性特性（使用过一次的数字）生成器来确保事务未被用过。作为安全性中的简单概念，确保一旦生成特殊一次性特性，就不再生成它，这一点很重要。另一个重要的安全性断言是安全文档决不被未授权的代理程序访问。

在多线程代码中，我们希望断言每个线程最终都会有运行机会。我们还希望确保系统决不会使其本身陷入死锁状态（即在两个或多个线程可以继续处理之前，它们正在彼此等待提供资源）。基本的非常规断言特性下面是我们希望获得的（通常想要多线程代码环境中获得的）两种非常有用的特性类型，不可能仅用常规断言来表示它们：

- 安全断言
- 生存断言

安全断言声明某些不合需要的系统状态将决不在任何环境下起作用。生存断言声明保证最终发生某些事件——例如，给定的线程将最终被唤醒，而不是永远休眠。时态逻辑可以帮助产生这些断言。

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)