

认识p-unit:一款开源的性能测试工 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/257/2021_2022__E8_AE_A4_E8_AF_86p-un_c67_257598.htm

p-unit 是一款开放源码的性能测试框架，和 JUnit 不同，JUnit 关注的是测试案例的正确性，而 p-unit 不仅关注测试案例的正确性，还收集测试案例的性能参数，默认情况下，p-unit 收集测试案例的时间和内存消耗情况，可以产生文件，图片，和 PDF 格式的报表。此外，p-unit 还支持参数化测试，多线程测试以及不同 Java 虚拟机性能之间的比较。p-unit 简介 或许我们已经习惯了使用 JUnit 来写单元测试来保证代码质量（我也一直这么做），但可能经常碰到这样的问题：程序多线程下正确性如何？如何测试程序的性能？当有多个方案可以选择时，技术上如何比较不同方案的性能？对于问题 1，我们或许听天由命？或是凭借人工分析，或是根据用户反馈？很多软件单线程下的单元测试覆盖率相当高，从而保证了代码的健壮性。然而多线程测试时常被忽略，这并不代表多线程测试不重要，相反，修正一个用户报告的多线程 BUG 往往比单线程的要高出很多，因为测试案例经常不是 100% 可重现的。这更要求程序员在开发阶段充分的重视。目前多线程单元测试力度不够的一个重要原因是没有一个像 JUnit 那样易用的测试工具，另外重复写测试案例往往不被程序员接受。对于问题 2，一个成熟的关心性能的产品往往有一个性能测试平台。这个测试平台应该关注的是测试业务逻辑本身，而无需关心如何运行测试案例。你是否为写这样的测试平台痛苦过？以及花费时间在产生一些直观的报表上面？对于问题 3，我们往往写一个原型

来比较不同产品之间的性能，如何比较执行速度和内存消耗？或是选择最适合你的虚拟机？p-unit就是这么一款开源的性能测试软件，它能帮助你很好的解决上述问题。p-unit可以：

- 多线程支持：同一个测试案例可以单线程执行，也可以多线程执行，测试案例开发者只需写一套测试案例。
- 参数化测试案例：很多测试案例，需要测试同一功能在不同数量级上的性能表现。
- 不同虚拟机性能测试：只需指定虚拟机路径，即可测试同一个测试案例在不同虚拟机上的表现，报表上可以非常直观显示性能差别。
- 事件机制构架：punit是基于事件机制构架的，如果用户想定制报表，只需实现事件响应器，并注册该响应器到punit核心即可。

多线程执行测试案例

在了解如何多线程执行测试案例之前，我们先了解一下如何利用p-unit单线程执行测试案例。不同于JUnit，p-unit测试用例无需继承任何测试类或是实现接口，即可执行test开始的方法。尽管JUnit 4中加入了注释（Annotation）的特性，但测试方法前缀为"test"仍然是测试者们的首选。因此如果你的JUnit测试案例遵循的是test命名规则，那么p-unit可以兼容运行JUnit测试案例。下面的代码清单1是一个最为普通的测试案例：

清单 1. 测试案例 1

```
public class SimpleTestClass {
    public void setUp() { SampleUtil.doSomething(); }
    public void tearDown() { SampleUtil.doSomething(); }
    public void testA() {
        System.out.println("testA");
        SampleUtil.doSomething();
    }
    public void testB() { SampleUtil.doSomething(); }
    public void testC() {
        SampleUtil.doSomething();
    }
}

public class SampleUtil {
    private static Random _random = new Random();
    public static void consumeMemory(int length) {
        byte[] data = new byte[length];
        for(int
```

```
i = 0, j = 0. i j.}}public static void consumeTime(int time)
{ThreadUtil.sleepIgnoreInterruption(time).}public static void
doSomething() {consumeTime(Math.abs(_random.nextInt()) %
500).consumeMemory(Math.abs(_random.nextInt()) % 100000).}
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
```