

JavaME下的单元测试开发之JUnit篇 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/138/2021\\_2022\\_JavaME\\_E4\\_B8\\_8B\\_E7\\_c97\\_138725.htm](https://www.100test.com/kao_ti2020/138/2021_2022_JavaME_E4_B8_8B_E7_c97_138725.htm) 摘要 不能因为Java Micro Edition缺乏反射能力就说Java Micro Edition开发者无法利用JUnit风格测试的优点。其实，借助于具有JUnit风格的其它一些框架和工具，Java ME开发人员仍然能够改进Java ME应用程序的开发质量。本系列文章（两篇）正是想详细探讨J2MEUnit和JUnit这两个开源框架在Java ME单元测试开发中的应用。

一、引言

如今，JUnit测试正在逐渐成为大多数Java标准版(SE)和企业版(EE)应用程序开发中的基本组成部分-对于那些积极拥护测试驱动开发者尤其如此。Kent Beck和Eric Gamma的最初的Smalltalk框架变得如此流行和成功，以至于它被移植到很多编程语言中，包括Ada(AUnit)、C#(NUnit)、Python(PyUnit)，甚至还有Fortran(fUnit)。实践证明，Java的JUnit是所有的单元测试框架中最成功的并且已经派生出许多JUnit"变种"（以各种"扩展"的形式），这些框架最终帮助在从多线程Java应用程序到高级企业Java应用程序的主流开发中实现单元测试。

然而，使用JUnit或找到一种JUnit扩展用于Java Micro Edition开发一直以来却是很难的事情。须知，JUnit框架依赖于Java反射API。由于Java ME环境中还不支持反射API，所以，典型的很大程度上依赖于反射的JUnit工具还不能帮助进行Java ME开发。尽管如此，已经出现两个专门针对设备应用程序开发者构建的Java ME JUnit扩展。值得注意的是，随着NetBeans和NetBeans Mobility 5.5的发行，NetBeans和NetBeans Mobility Pack正在计划合并一个Java ME JUnit风格的框架。这种新版本

的IDE将以一种更为利索的方式在你的Java ME应用程序中加入单元测试。 本文将通过使用Java ME JUnit框架向你介绍JUnit测试；通过本文，你会发现，如何获取这些工具，如何使用它们进行测试以及如何使用它们来构建质量更好的软件。

## 二、 获取Java ME单元测试框架

如今，市场上存在两个可用的JavaME JUnit测试框架，它们是J2MEUnit和JMUnit。这两个工程都是可自由下载的开源框架，你可以从SourceForge.net上下载一个打包文件。然而，这两个开源工程的工程主管(Elmar Sonnenschein和Brunno Silva，分别维护J2MEUnit和JMUnit)，正在计划把这两个框架合并为一个。新的工程将在J2MEUnit工程基础上得到进一步巩固。根据Sonnenschein本人的说法，"因为更多的现有用户的工程都是基于SourceForge上的J2MEUnit框架；所以，我们计划基于Brunno的JMUnit 2.0创建一个J2MEUnit 2.0发行版本。"Silva在一次最近的新闻发布会上声称在今年年底可能无法完成产品的合并和一个2.0版本的发行。Silva进一步建议说，新的工程"不想破坏这两个框架的当前用户的原有代码，因此，原始代码会继续存在，只是不再建议使用。新的单元框架应该展现出两个框架-JMUnit和J2MEUnit-各自的特色。"

## 三、 一个简单的示例应用程序

在分析各种单元测试框架之前，你需要一些简单的代码以备测试之用。在这个例子中，下面这个简单的Conversion类能够用于创建并测试Java ME单元测试。

```
public class DistanceConversion {
    public static int feetToMeters(int ft){
        return (ft * 3048)/10000. }
    public static int metersToFeet(int meters){
        return (meters*3281)/1000. }
    public static int milesToKM(int miles){
        return (miles*1609)/1000. }
    public static int kmToMiles(int km){
```

```
return (km*6214)/10000. }}public class TemperatureConversion {  
public static float fahrenheitToCelsius (float degrees){ return  
((degrees-32)/9)*5. } public static float celsiusToFahrenheit (float  
degrees){ return ((degrees * 9)/5) 32. } public static boolean isHotter  
(float degFaren , float degCel){ return  
((fahrenheitToCelsius(degFaren))-degCel) > 0. } public static  
boolean isCooler (float degFaren , float degCel){ return  
((fahrenheitToCelsius(degFaren))-degCel) } } 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
```