

JAVA无线开发之定点库MathFP使用入门 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022_JAVA_E6_97_A0_E7_BA_BF_c104_167044.htm 众所周知，CLDC1.0是不支持小数运算的，而CLDC1.1才支持浮点运算。但是目前市面上的手机，绝大部分是采用CLDC1.0这种configuration。那我们需要进行小数运算怎么办呢？比如说要绘制任意角度的飞机运行轨迹。你当然可以自己写一个类，用整数来模拟定点小数运算(模拟浮点小数运算非常困难)，不过你不必重新发明轮子,网上有很多用整数运算来模拟小数运算的代码库，而MathFP就是其中非常优秀的一个，它健壮，稳定，高速，是在J2ME环境中进行小数运算的不二之选，而且最关键的是，它的体积很小。MathFP的下载地址

是:<http://home.rochester.rr.com/ohommes/MathFP>。我下载的版本是基于CLDC的，下载的MathFP版本号是1.1.2.下载回来的全部东西就是一个MathFP.class(该类所在的包名是net.jscience.util)，你可以把该类置于你的classpath中进行开发，发布软件的时候把该class加入到jar文件中。或者你也可以把该class反编译，得到源码，直接放入你的工程的src目录中，我这里采用的是后一种做法。记得同时把MathFP的API文档下载回来。因为MathFP是用整数来模拟定点小数的，所以小数的内部表现形式还是一个整数，但是你一定要记得把表示小数的整数和真正的整数区别开来，否则就会造成很多难于调试的bug(一个小技巧就是表示小数的整形变量名以FP为后缀)。你只需要掌握一个原则，就是首先把要参与小数运算的整数都转换成小数，然后进行小数运算，运算完了以后

，再把结果转换成成整数使用。下面的这个例子，就是用来演示MathFP的基本使用方法的。假定屏幕左下角有一个点，每隔100毫秒，就沿60度的角度向东北方向运动5个像素，绘制出此点的运动轨迹。这个例子涉及到小数和三角运算，因为该点x坐标的增量是 $\cos 60(\text{度})$ ，y坐标上的增量是 $-\sin 60(\text{度})$ 。我们来看代码怎么写：

```
import javax.microedition.lcdui.Canvas.  
import javax.microedition.lcdui.Graphics. import  
net.jsience.util.MathFP. /** * 小数运算演示Canvas * @author  
Jagie */ public class FloatCanvas extends Canvas implements  
Runnable { //用于统计屏幕刷新次数 int paintCount. //屏幕宽度  
，高度。定点数 int w_FP, h_FP. //当前点坐标，前一点坐标,定  
点数 int curX_FP, curY_FP, lastX_FP, lastY_FP. //速率 public static  
final int RATE = 5. public FloatCanvas() { w_FP =  
MathFP.toFP(this.getWidth()). h_FP =  
MathFP.toFP(this.getHeight()). //开始点处于屏幕的左下角  
lastX_FP = MathFP.toFP(0). lastY_FP = h_FP. new  
Thread(this).start(). } protected void paint(Graphics g) { //第一次  
只是清屏 if (paintCount == 0) { g.setColor(0). g.fillRect(0, 0,  
w_FP, h_FP). } else { //画线 g.setColor(0x00ff00). //把定点数转换  
成整数 g.drawLine(MathFP.toInt(lastX_FP),  
MathFP.toInt(lastY_FP), MathFP .toInt(curX_FP),  
MathFP.toInt(curY_FP)). } paintCount . } public void run() { //当  
前点没有超出屏幕时循环 while (curX_FP curY_FP amp.  
MathFP.toInt(curX_FP) >= 0 amp. MathFP.toInt(curY_FP) >= 0) {  
//60度角度转换成弧度 int radians =  
MathFP.div(MathFP.mul(MathFP.toFP(60), MathFP.PI),
```

```
MathFP.toFP(180)). //x方向增量 int deltaX =
MathFP.mul(MathFP.toFP(RATE), MathFP.cos(radians)). //y方向
增量 int deltaY = MathFP.mul(MathFP.toFP(RATE),
MathFP.sin(radians)). //新坐标 , 定点数 curX_FP = lastX_FP
deltaX. curY_FP = lastY_FP - deltaY. System.out.println(curX_FP
"," curY_FP). repaint(). try { Thread.sleep(100). } catch
(InterruptedExcepTion e) { // TODO Auto-generated catch block
e.printStackTrace(). } //新坐标成为旧坐标 lastX_FP = curX_FP.
lastY_FP = curY_FP. } } }该Canvas在设备上绘制效果如下图: 大
家可以看到 , 曲线正沿60度角的方向朝东北方向不停的增长
。有了这个定点库 , 我们就能在游戏中使用小数运算了 , 所
以一些简单的游戏物理算法也可以使用了。 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
```