链滴

浮点数在虚拟机中的表示

作者：zwxbest
原文链接：https://ld246.com/article/1546593324003
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

综述

在Java虚拟机中，浮点数有float和double两种，分别是32位和64位浮点数，目前，使用最为广泛的由IEEE754定义的浮点数格式。目前Java虚拟机中对于浮点数的处理参考IEEE754的规范。

float

本节将以float为例，简要介绍浮点数的表示方法。

在IEEE754的定义中，一个浮点数由3部分组成，分别是：符号位、指数位和尾数位。以32位float为例，符号位占1位，用s表示表示正负数，指数位占8位，用e表示。尾数位占剩余的23位，用m表示

sflag*m*2^(e-127)

其中，sflag表示符号，当s为0时，sflag为1，当s为1时，sflag为-1。m为尾数值，实际占用空间为23位，但是根据e的取值，有24位精度。当e全为0时，尾数位附加为0,否则，尾数位附加为1。e为指数位，用8位表示。

举例

以浮点数-5为例，其内部表示为：

11000000101000000000000000000000

内部表示可以通过以下代码打印

System.out.println(Integer.toBinaryString(Float.floatToRawIntBits(a)));

符号位为1表示负数，

指数位为10000001,表示129。

尾数位为：01000000000000000000000。因为e不全为0，故实际的尾数位为：10100000000000000000000,最前面的1是根据e添加的，参与运算，但不参与展示。

尾数位表示2的指数次幂的和，每一位表示求和数列中的对应项是否为0,这里表示:

1*2^0+0*2^1+1*2^2+0*2^3+0*2^4+0*2^5

故11000000101000000000000000000000的值为：

-1*2^(129-127)*(1*2^0+0*2^1+1*2^2+0*2^3+0*2^4+0*2^5)=-1*4*1.25=-5

特殊数字

浮点数float还可以表示一些特殊的数字

正无穷01111111100000000000000000000000
负无穷11111111100000000000000000000000
NaN01111111110000000000000000000000
最大浮点数(3.4028235E38)01111111011111111111111111111111
最小规范化正浮点数(1.17549435E-38)00000000100000000000000000000000
最小正浮点数(1.4E45)00000000000000000000000000000001

原文链接：浮点数在虚拟机中的表示

00000000000000000000000000000000

其中，指数位全为1的表示无穷大和NaN等特殊数字。指数位全为0的为非规范化的浮点数。

System.out.println(工nteger.toBinaryString(Float.floatToRawIntBits(a)));

原文链接：浮点数在虚拟机中的表示