
浮点型数据

分类及范围

Java 中浮点数分为单精度浮点数和双精度浮点数，分别是 float 和 double。

浮点数就是小数点在逻辑上是不固定的；单精度和双精度的取值范围和精度也是不同的。

float 的取值范围是 4 字节也就是 32 位，精度是 7~8 位；

double 的取值范围是 8 字节也就是 64 位，精度是 16~17 位。

因为浮点型的特殊存储结构，所以在进行数据的计算，或者数据的转化时，会有精度的损失。

那如何对浮点型进行精确的计算呢：

精确计算

通过 BigDecimal 类，先将浮点型数据转换为 String 类型的数据，然后再进行相应的计算。

以下为利用 BigDecimal 类进行加法和减法运算的测试代码：

```

public class Test {

    public static void main(String[] args) {
        double a = 1.9;
        double b = 1.7;
        System.out.println("直接进行减法运算，运算结果：\n" + (a - b));
        System.out.println("利用BigDecimal类的进行减法运算结果：\n" + sub(a, b));
        System.out.println();
        System.out.println("直接进行加法运算，运算结果：\n" + (a + b));
        System.out.println("利用BigDecimal类的进行加法运算结果：\n" + add(a, b));
    }

    //利用BigDecimal类的进行减法运算
    public static double sub(double v1, double v2) {
        // 将浮点数转为String类型
        BigDecimal b1 = new BigDecimal(Double.toString(v1));
        BigDecimal b2 = new BigDecimal(Double.toString(v2));
        // 通过BigDecimal类中subtract方法完成减法运算，
        // 其中b1是被减数，b2是减数
        double b = b1.subtract(b2).doubleValue();
        return b;
    }

    //利用BigDecimal类的进行加法运算
    public static double add(double v1, double v2) {
        BigDecimal b1 = new BigDecimal(Double.toString(v1));
        BigDecimal b2 = new BigDecimal(Double.toString(v2));
        return b1.add(b2).doubleValue();
    }
}

```

运行结果参考下图：

```

直接进行减法运算，运算结果：
0.19999999999999996
利用BigDecimal类的进行减法运算结果：
0.2

直接进行加法运算，运算结果：
3.5999999999999996
利用BigDecimal类的进行加法运算结果：
3.6

```

如果要运算的数据类型是单精度浮点型 float 那就应该将参数和返回值都改为 float 类型。

关于使用 BigDecimal 类，对浮点型数据进行其他运算的代码，可以查阅 BigDecimal 的 API，地址为：<http://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html>

浮点数的应用场景：

要求有高精度的，医学，化学，军事；数量级特别大的，天文；游戏，图形图像处理等方面。