

BigDecimal 的加减乘除及比较大小

作者: [hazanr123](#)

原文链接: <https://ld246.com/article/1545111736009>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

import java.math.BigDecimal;
import static java.lang.System.out;
public class BaseClass {
    public static void main(String[] args) {
        BigDecimal num1 = new BigDecimal("100");
        BigDecimal num2 = new BigDecimal("50");

        //加法
        BigDecimal result1 = num1.add(num2);
        out.println("和 是 : " + result1);

        //减法
        BigDecimal result2 = num1.subtract(num2);
        out.println("差 是 : " + result2);

        //乘法
        BigDecimal result3 = num1.multiply(num2);
        out.println("积 是 : " + result3);

        //除法
        BigDecimal result4 = num1.divide(num2);
        out.println("商 是 : " + result4);

        //比较
        BigDecimal num = new BigDecimal("100"); //用做比较的值
        out.println(num1.compareTo(num2)); //大于时, 返回 1
        out.println(num.compareTo(num1)); //等于时, 返回 0
        out.println(num2.compareTo(num1)); //小于时, 返回 -1

        out.println(num1.min(num2)); //求两个数的最小值,返回小的数
        out.println(num1.max(num2)); //求两个数的最大值,返回大的数

        // compareTo方法比较时,不同于equals方法,需要两边都不为null,否则会报空指针异常,
        // 源码如下
        // public int compareTo(BigDecimal val) {
        //     // Quick path for equal scale and non-inflated case.
        //     if (scale == val.scale) {
        //         long xs = intCompact;
        //         long ys = val.intCompact;
        //         if (xs != INFLATED && ys != INFLATED)
        //             return xs != ys ? ((xs > ys) ? 1 : -1) : 0;
        //     }
        //     int xsign = this.signum();
        //     int ysign = val.signum();
        //     if (xsign != ysign)
        //         return (xsign > ysign) ? 1 : -1;
        //     if (xsign == 0)
        //         return 0;
        //     int cmp = compareMagnitude(val);
        //     return (xsign > 0) ? cmp : -cmp;
        // }

        // max/min方法源代码如下

```

```
// public BigDecimal max(BigDecimal val) {  
//     return (compareTo(val) >= 0 ? this : val);  
// }  
// public BigDecimal min(BigDecimal val) {  
//     return (compareTo(val) <= 0 ? this : val);  
// }  
}  
}
```