```
package bd;

import java.math.BigDecimal;
import java.math.RoundingMode;

public class BigDecimalExample {

        public static void main (String args[]) {

                // There is a constructors that allows you to create a BigDecimal
using a String
                BigDecimal aNumber = new BigDecimal("4.32");

                // ... and another, using a numerical literal, quite flexible!
                BigDecimal anotherNumber = new BigDecimal(7.19);

                // BigDecimals are not mutable
                // What I'm trying to do: 4.32 - 7.19
                BigDecimal result = aNumber.subtract(anotherNumber);
                result = result.setScale(2, RoundingMode.HALF_UP);

                // What I'm trying to do: 4.32 + 7.19
                BigDecimal result2 = aNumber.add(anotherNumber);
                result2 = result2.setScale(2, RoundingMode.HALF_UP);

                System.out.println(result);
                System.out.println(result2);

                int check = aNumber.compareTo(anotherNumber);
                // is aNumber greater than anotherNumber? NO -> return -1
                System.out.println(check);

                int check2 = anotherNumber.compareTo(aNumber);
                // is anotherNumber greater than aNumber? YES -> return 1
                System.out.println(check2);

                int check3 = anotherNumber.compareTo(new BigDecimal(7.19));
                // if both values are equal return zero.
                System.out.println(check3);

        }
}
```