JAVA.MATH.BIGINTEGER.MODINVERSE METHOD

Description

The **java.math.BigInteger.modInverse*BigIntegerm*** returns a BigInteger whose value is ($this^{-1}$ mod m).

Declaration

Following is the declaration for **java.math.BigInteger.modInverse** method

```
public BigInteger modInverse(BigInteger m)
```

Parameters

- **m** - the modulus

Return Value

This method returns a BigInteger object whose value is $this^{-1}$ mod m.

Exception

- **ArithmeticException** - if m ≤ 0, or this BigInteger has no multiplicative inverse mod m *thatis, thisBigIntegerisnotrelativelyprimetom*.

Example

The following example shows the usage of math.BigInteger.modInverse method

```
package com.tutorialspoint;

import java.math.*;

public class BigIntegerDemo {

public static void main(String[] args) {

        // create 3 BigInteger objects
        BigInteger bi1, bi2, bi3;

 // Two numbers are relatively prime if they have no
 // common factors, other than 1.
        bi1 = new BigInteger("7");
        bi2 = new BigInteger("20");

        // perform modInverse operation on bi1 using bi2
 bi3 = bi1.modInverse(bi2);

        String str = bi1 + "^-1 mod " + bi2 + " is " +bi3;

 // print bi3 value
 System.out.println( str );
    }
}
```

Let us compile and run the above program, this will produce the following result:

```
7^-1 mod 20 is 3
```