JAVA.MATH.BIGINTEGER.MODPOW METHOD

Description

The **java.math.BigInteger.modPow*BigIntegerexponent, BigIntegerm*** returns a BigInteger whose value is (this^{exponent} mod m). Unlike pow, this method permits negative exponents.

Declaration

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

```
public BigInteger modPow(BigInteger exponent, BigInteger m)
```

Parameters

- **exponent** - the exponent
- **m** - the modulus

Return Value

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

Exception

- **ArithmeticException** - if m ≤ 0 or the exponent is negative and this BigInteger is not relatively prime to m.

Example

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

```
package com.tutorialspoint;

import java.math.*;

public class BigIntegerDemo {

public static void main(String[] args) {

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

 // create a BigInteger exponent
 BigInteger exponent = new BigInteger("2");

        bi1 = new BigInteger("7");
        bi2 = new BigInteger("20");

        // perform modPow operation on bi1 using bi2 and exp
 bi3 = bi1.modPow(exponent, bi2);

        String str = bi1 + "^" +exponent+ " 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:

```
7^2 mod 20 is 9
```