JAVA.MATH.BIGINTEGER.SHIFTRIGHT METHOD

Description

The **java.math.BigInteger.shiftRight*intn*** returns a BigInteger whose value is $this >> n$. Sign extension is performed. The shift distance, n, may be negative, in which case this method performs a left shift. It computes floor(this / 2^n).

Declaration

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

```
public BigInteger shiftRight(int n)
```

Parameters

- **n** - shift distance, in bits

Return Value

This method returns a BigInteger object whose value is this >> n .

Exception

- **ArithmeticException** - if the shift distance is Integer.MIN_VALUE

Example

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

```
package com.tutorialspoint;

import java.math.*;

public class BigIntegerDemo {

public static void main(String[] args) {

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

 bi1 = new BigInteger("4");

 // perform right shift operation on bi1 using 2 and -2
 bi2 = bi1.shiftRight(2);
 bi3 = bi1.shiftRight(-2);

 String str1 = "Right shift on " +bi1+ ", 2 times gives " +bi2;
 String str2 = "Right shift on " +bi1+ ", -2 times gives " +bi3;

 // print bi2, bi3 values
 System.out.println( str1 );
 System.out.println( str2 );
   }
}
```

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

```
Right shift on 4, 2 times gives 1
Right shift on 4, -2 times gives 16
```