

JAVA.LANG.THREAD.SLEEP METHOD

http://www.tutorialspoint.com/java/lang/thread_sleep_millis_nanos.htm

Copyright © tutorialspoint.com

Description

The **java.lang.Thread.sleep(long millis, int nanos)** method causes the currently executing thread to sleep for the specified number of milliseconds plus the specified number of nanoseconds, subject to the precision and accuracy of system timers and schedulers.

Declaration

Following is the declaration for **java.lang.Thread.sleep** method

```
public static void sleep(long millis, int nanos) throws InterruptedException
```

Parameters

- **millis** -- This is the length of time to sleep in milliseconds.
- **nanos** -- This is 0-999999 additional nanoseconds to sleep.

Return Value

This method does not return any value.

Exception

- **IllegalArgumentException** -- if the value of millis is negative or the value of nanos is not in the range 0-999999.
- **InterruptedException** -- if any thread has interrupted the current thread. The interrupted status of the current thread is cleared when this exception is thrown.

Example

The following example shows the usage of java.lang.Thread.sleep method.

```
package com.tutorialspoint;

import java.lang.*;

public class ThreadDemo implements Runnable {

    Thread t;

    public void run() {
        for (int i = 10; i < 13; i++) {

            System.out.println(Thread.currentThread().getName() + " " + i);
            try {
                // thread to sleep for 1000 milliseconds plus 500 nanoseconds
                Thread.sleep(1000, 500);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }

    public static void main(String[] args) throws Exception {
        Thread t = new Thread(new ThreadDemo());
        // this will call run() function
        t.start();

        Thread t2 = new Thread(new ThreadDemo());
```

```
// this will call run() function  
t2.start();  
}  
}
```

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

```
Thread-0 10  
Thread-1 10  
Thread-0 11  
Thread-1 11  
Thread-0 12  
Thread-1 12
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js