

BASICS OF JAVA **OOPS CONCEPTS** **STRING HANDLING** **EXCEPTION HANDLING** **JAVA MULTITHREADING**

Introduction to Multithreading

Thread Class

Creating a thread

Joining threads

Sleeping Thread in Java

Naming Thread in Java

Thread Priority in Java

Daemon Thread in Java

Synchronization

Interthread Communication

Thread group

ADVANCED TOPICS **COLLECTION FRAMEWORK** **JAVA GUI** **REFLECTION API** **RMI APPLICATION** **INNER CLASS** **WRAPPER CLASS** **FILE HANDLING** **LIST** **SET** **MAP** **QUEUE & DEQUE** **JDBC** **LAYOUT MANAGERS**

ADVERTISEMENT

Java Sleeping Thread

ADVERTISEMENT

To sleep a thread for a specified time, Java provides sleep method which is defined in Thread class. The sleep method is an overloaded method which are given below. It throws interrupted exception so make sure to provide proper handler.

It always pause the current thread execution. Any other thread can interrupt the current thread in sleep, in that case InterruptedException is thrown.

Syntax

```
sleep(long millis) throws InterruptedException
sleep(long millis, int nanos) throws InterruptedException
```

Example : Sleeping a thread

In this example, we are sleeping threads by using sleep method. Each thread will sleep for 1500 milliseconds and then resume its execution. See the below example.

```
public class MyThread extends Thread
{
    MyThread(String str)
    {
        super(str);
    }

    public void run()
    {
        System.out.println(Thread.currentThread().getName()+" Started");
        try{
            MyThread.sleep(1500);
        }catch(InterruptedException ie){
            System.out.println(ie);
        }
        System.out.println(Thread.currentThread().getName()+" Finished");
    }

    public static void main(String[] args)
    {
        ...
    }
}
```

OUTPUT:

```
first thread Started
second thread Started
first thread Finished
second thread Finished
```

Example

```
public class MyThread extends Thread
{
    MyThread(String str)
    {
        super(str);
    }

    public void run()
    {
        System.out.println(Thread.currentThread().getName()+" Started");
        try{
            MyThread.sleep(1500);
            System.out.println(Thread.currentThread().getName()+" Sleeping..");
        }catch(InterruptedException ie){
            System.out.println(ie);
        }
        System.out.println(Thread.currentThread().getName()+" Finished");
    }

    public static void main(String[] args)
    {
        ...
    }
}
```

OUTPUT:

```
first thread state: NEW
first thread state: RUNNABLE
second thread state: NEW
first thread Started
second thread state: RUNNABLE
second thread Started
first thread Sleeping..
second thread Sleeping..
second thread Finished
first thread Finished
```

`Thread.sleep()` interacts with the thread scheduler to put the current thread in wait state for specified period of time. Once the wait time is over, thread state is changed to runnable state and wait for the CPU for further execution. So the actual time that current thread sleep depends on the thread scheduler that is part of operating system.

[← Prev](#)[Next →](#)

ADVERTISEMENT

Java MCQ Tests

Prepare for Java Interview in TCS, Infosys, etc. companies.

[Explore](#)**Java Programs**

Java programs with code and output for practice.

[Explore](#)**Spring Framework**

Learn the most widely used Java framework in the World.

[Explore](#)

ADVERTISEMENT

ADVERTISEMENT

