

S.N.	Modifier and Type	Method	Description
1)	void	<u>start()</u>	It is used to start the execution of the thread.
2)	void	<u>run()</u>	It is used to do an action for a thread.
3)	static void	<u>sleep()</u>	It sleeps a thread for the specified amount of time.
4)	static Thread	<u>currentThread()</u>	It returns a reference to the currently executing thread object.
5)	void	<u>join()</u>	It waits for a thread to die.
6)	int	<u>getPriority()</u>	It returns the priority of the thread.
7)	void	<u>setPriority()</u>	It changes the priority of the thread.
8)	String	<u>getName()</u>	It returns the name of the thread.
9)	void	<u>setName()</u>	It changes the name of the thread.
10)	long	<u>getId()</u>	It returns the id of the thread.
11)	boolean	<u>isAlive()</u>	It tests if the thread is alive.
12)	static void	<u>yield()</u>	It causes the currently executing thread object to pause and allow other threads to execute temporarily.
13)	void	<u>suspend()</u>	It is used to suspend the thread.
14)	void	<u>resume()</u>	It is used to resume the suspended thread.
15)	void	<u>stop()</u>	It is used to stop the thread.
16)	void	<u>destroy()</u>	It is used to destroy the thread group and all of its subgroups.
17)	boolean	<u>isDaemon()</u>	It tests if the thread is a daemon thread.
18)	void	<u>setDaemon()</u>	It marks the thread as daemon or user thread.
19)	void	<u>interrupt()</u>	It interrupts the thread.
20)	boolean	<u>isinterrupted()</u>	It tests whether the thread has been interrupted.
21)	static boolean	<u>interrupted()</u>	It tests whether the current thread has been interrupted.
22)	static int	<u>activeCount()</u>	It returns the number of active threads in the current thread's thread group.
23)	void	<u>checkAccess()</u>	It determines if the currently running thread has permission to modify the thread.
24)	static boolean	<u>holdLock()</u>	It returns true if and only if the current thread holds the monitor lock on the specified object.

25)	static void	<u>dumpStack()</u>	It is used to print a stack trace of the current thread to the standard error stream.
26)	StackTraceElement[]	<u>getStackTrace()</u>	It returns an array of stack trace elements representing the stack dump of the thread.
27)	static int	<u>enumerate()</u>	It is used to copy every active thread's thread group and its subgroup into the specified array.
28)	Thread.State	<u>getState()</u>	It is used to return the state of the thread.
29)	ThreadGroup	<u>getThreadGroup()</u>	It is used to return the thread group to which this thread belongs
30)	String	<u>toString()</u>	It is used to return a string representation of this thread, including the thread's name, priority, and thread group.
31)	void	<u>notify()</u>	It is used to give the notification for only one thread which is waiting for a particular object.
32)	void	<u>notifyAll()</u>	It is used to give the notification to all waiting threads of a particular object.
33)	void	<u>setContextClassLoader()</u>	It sets the context ClassLoader for the Thread.
34)	ClassLoader	<u>getContextClassLoader()</u>	It returns the context ClassLoader for the thread.
35)	static Thread.UncaughtExceptionHandler	<u>getDefaultUncaughtExceptionHandler()</u>	It returns the default handler invoked when a thread abruptly terminates due to an uncaught exception.
36)	static void	<u>setDefaultUncaughtExceptionHandler()</u>	It sets the default handler invoked when a thread abruptly terminates due to an uncaught exception.

Java Thread Methods