**ThreadGroup in Java**

Java provides a convenient way to group multiple threads in a single object. In such way, we can suspend, resume or interrupt group of threads by a single method call.

**Note: Now suspend(), resume() and stop() methods are deprecated.**

***Java thread group is implemented by java.lang.ThreadGroup class.***

**Constructors of ThreadGroup class**

|  |  |  |
| --- | --- | --- |
| **No.** | **Constructor** | **Description** |
| 1) | ThreadGroup(String name) | creates a thread group with given name. |
| 2) | ThreadGroup(ThreadGroup parent, String name) | creates a thread group with given parent group and name. |

**Important methods of ThreadGroup class**

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
| 1) | int activeCount() | returns no. of threads running in current group. |
| 2) | int activeGroupCount() | returns a no. of active group in this thread group. |
| 3) | void destroy() | destroys this thread group and all its sub groups. |
| 4) | String getName() | returns the name of this group. |
| 5) | ThreadGroup getParent() | returns the parent of this group. |
| 6) | void interrupt() | interrupts all threads of this group. |
| 7) | void list() | prints information of this group to standard console. |

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.\*; **import** java.util.Scanner; **import** java.util.concurrent.ExecutorService; **import** java.util.concurrent.Executors;  **public class** Main {   **public static void** main(String[] args){  Multis multis = **new** Multis();   ThreadGroup threadGroup = **new** ThreadGroup(**"first thread group"**);   Thread t1 = **new** Thread(threadGroup, multis, **"One"**);  t1.start();  Thread t2 = **new** Thread(threadGroup, multis, **"Two"**);  t2.start();  Thread t3 = **new** Thread(threadGroup, multis, **"Three"**);  t3.start();   System.***out***.println(**"Thread group name : "** + threadGroup.getName());  threadGroup.list();  } } | One  Thread group name : first thread group  java.lang.ThreadGroup[name=first thread group,maxpri=10]  Thread[Two,5,first thread group]  Thread[Three,5,first thread group]  Two  Three |
| **package** com.company;  **import** java.util.Date;  **public class** Multis **extends** Thread{   @Override  **public void** run(){  System.***out***.println(*currentThread*().getName());  } } |