Daemon Thread

A daemon thread is a thread that does not prevent the JVM from exiting when the program finishes but the thread is still running.

An example for a daemon thread is the garbage collection.

When a new thread is created it inherits the daemon status of its parent.

When all non-daemon threads finish, the JVM halts, and any remaining daemon threads are abandoned:

* + finally blocks are not executed,
  + stacks are not unwound - the JVM just exits.

class WorkerThread1 extends Thread

{

public WorkerThread1()

{

setDaemon(true);

}

public void run()

{

int count = 0;

while (true)

{

System.out.println("Hello from Worker1 "+count++);

try

{

sleep(1000);

}

catch (InterruptedException e) {}

}

}

}

class WorkerThread2 extends Thread

{

public void run()

{

int count = 0;

while (true)

{

System.out.println("Hello from Worker2 "+count++);

try

{

sleep(1200);

} catch (InterruptedException e) {}

}

}

}

public class DaemonTest

{

public static void main(String[] args)

{

new WorkerThread1().start();

WorkerThread2 t1=new WorkerThread2();

t1.setDaemon(true);

t1.start();

try

{

Thread.sleep(7500);

}

catch (InterruptedException e) {}

System.out.println("Main Thread ending") ;

}

}