package thread_ex;

public class Thread_ex {
    boolean done = false;

    public Thread_ex(int NumT1, int NumT2) {
        /*
         * Constructor for the main class, its function is to spawn the
         * T1 and T2 threads
         */
        for (int i = 0; i < NumT1; i++) {
            new T1().start();
        }
        for (int i = 0; i < NumT2; i++) {
            new T2().start();
        }
    }

    public class T1 extends Thread {
        /*
         * T1 sleeps until done
         */
        public void run() {
            try {
                Thread.sleep(1000);
            } catch (InterruptedException ex) {
                System.out.println("Interrupt in T1");
            }
            System.out.println("T1 thread finished");
        }
    }

    public class T2 extends Thread {
        /*
         * T2 sleeps until done
         */
        public void run() {
            try {
                Thread.sleep(1000);
            } catch (InterruptedException ex) {
                System.out.println("Interrupt in T2");
            }
            System.out.println("T2 thread finished");
        }
    }

    public static void main(String[] args) {
        Thread_ex tex = new Thread_ex(2,2);
        /*
         * Sleep for 5 seconds and kill
         */
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            tex.done = true;
        }
        System.out.println("main finished");
    }
}

T1 thread started
T1 thread started
T2 thread started
T2 thread started
main finished
T2 thread finished
T2 thread finished
T1 thread finished
T1 thread finished
package thread_join;
import java.util.ArrayList;

public class Thread_join {
    static ArrayList<Thread> tname = new ArrayList<Thread>();
    static boolean done = false;
    /* Constructor for Thread_join, it is responsible for firing off the
    * threads */
    Thread_join (int numT1, int numT2) {
        for (int i=0; i<numT1; i++) {
            tname.add(new Thread (new T1()));
        }
        for (int i=0; i<numT2; i++) {
            tname.add(new Thread(new T2()));
        }
        for (int i=0; i<tname.size(); i++) {
            tname.get(i).start();
        }
    }

    class T1 implements Runnable{
        public void run() {
            while (!done){
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println("T1 interrupted.");
                }
            }
            System.out.println("T1 thread exiting.");
        }
    }

    class T2 implements Runnable{
        public void run() {
            while (!done){
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println("T2 interrupted.");
                }
            }
            System.out.println("T2 thread exiting.");
        }
    }

    public static void main(String args[]) {
        /* Call the constructor to fire off the threads */
        new Thread_join (2,2);
        /* Check to see if the threads are alive */
        for (int i=0; i<tname.size(); i++)
            System.out.println ("Thread #" + i + " is alive: " + tname.get(i).isAlive());
        /* Sleep for 5 seconds */
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            System.out.println ("Main thread interrupted");
        }
        /* Set done to true and join the outstanding threads */
        done = true;
        try {
            for (int i=0; i<tname.size(); i++)
                tname.get(i).join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted");
        }
        /* Show that the threads are gone */
        for (int i=0; i<tname.size(); i++)
            System.out.println ("Thread #" + i + " is alive: " + tname.get(i).isAlive());
            System.out.println("Main thread exiting.");
    }
}

Thread #0 is alive: true
Thread #1 is alive: true
Thread #2 is alive: true
Thread #3 is alive: true
T1 thread exiting.
T2 thread exiting.
Thread #0 is alive: false
Thread #1 is alive: false
Thread #2 is alive: false
Thread #3 is alive: false
Main thread exiting.