```
// Demonstrate thread priorities.
class clicker implements Runnable {

    long click = 0;
    Thread t;

    private volatile boolean running = true;

        public clicker(int p) {
            Thread(this);
            t.setPriority(p);
        }

        public void run() {
            while (running) {
                    click++;
            }
        }

        public void stop() {
            running = false;
        }

        public void start() {
            t.start();
        }
}
```

```
class HiLoPri {
    public static void main(String args[]) {
            Thread.currentThread().setPriority(Thread.MAX_PRIORITY);

            clicker hi = new clicker(Thread.NORM_PRIORITY + 2);

            clicker lo = new clicker(Thread.NORM_PRIORITY - 2);

            lo.start();

            hi.start();

            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println("Main thread interrupted.");
            }


    lo.stop();

    hi.stop();

    // Wait for child threads to terminate.
    try {
        hi.t.join();
        lo.t.join();
    } catch (InterruptedException e) {
        System.out.println("InterruptedException caught");
    }

    System.out.println("Low-priority thread: " + lo.click);

    System.out.println("High-priority thread: " + hi.click);

    }

}
```

Output:

```
Low-priority thread: 5798834241

High-priority thread: 5852293016
```