```
// $Id: observereport.java,v 1.2 2018-05-23 15:05:02-07 - - $

//
// Observers/ reporters problem.
// Observer waits for intervals and counts events happening.
// Reporter periodically reports these events
//

import java.util.*;
import static java.lang.Math.*;
import static java.lang.String.*;
import static java.lang.System.*;

class observereport {
   static long nanostart = nanoTime();

   static void print (String fmt, Object... args) {
      Thread self = Thread.currentThread();
      double millisec = (nanoTime() - nanostart) / 1e6;
      String ident = format ("%10.3f: %s(%d): ",
                             millisec, self.getName(), self.getId());
      String message = format (fmt, args);
      out.println (ident + message);
      out.flush();
   }

   static class counter {
      private long count = 0;
      public boolean stop = false;
      synchronized void click () {
         ++count;
      }
      synchronized long reset () {
         long result = count;
         count = 0;
         return result;
      }
   }

   public static void main (String[] args) {
      print ("starting");
      counter counted = new counter ();
      Thread obs = new Thread (new observer (counted, 10));
      Thread rep = new Thread (new reporter (counted, 1000, 10));
      obs.start ();
      rep.start ();
      print ("finished");
   }


   static class observer implements Runnable {
      counter counted;
      long delay_msec;
      Random random = new Random ();
      observer (counter _counted, long _delay_msec) {
         counted = _counted;
         delay_msec = _delay_msec;
      }
      public void run () {
         Thread self = Thread.currentThread();
         self.setName ("observer");
         print ("starting");
         while (! counted.stop) {
            try {
               long delay = abs ((long) (delay_msec
                          + delay_msec * random.nextGaussian ()));
               Thread.sleep (delay);
            }catch (InterruptedException exn) {
               print ("observer interrupted: %s", exn);
            }
            counted.click ();
         }
         print ("finished");
      }
   }

   static class reporter implements Runnable {
      counter counted;
      long delay_msec;
      int cycles;
      reporter (counter countinit, long _delay_msec, int _cycles) {
         counted = countinit;
         delay_msec = _delay_msec;
         cycles = _cycles;
      }
      public void run () {
         Thread self = Thread.currentThread();
         self.setName ("reporter");
         print ("starting");
         for (int count = 0; count < cycles; ++count) {
            try {
               Thread.sleep (delay_msec);
            }catch (InterruptedException exn) {
               print ("reporter interrupted: %s", exn);
            }
            long events = counted.reset ();
            print ("report %3d: events: %5d", count, events);
         }
         counted.stop = true;
         print ("finished");
      }
   }
}

//TEST// ./observereport >observereport.out
//TEST// mkpspdf observereport.ps observereport.java* observereport.out

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting observereport.java
observereport.java:
     $Id: observereport.java,v 1.2 2018-05-23 15:05:02-07 - - $
javac -Xlint observereport.java
jar cfm observereport Manifest observereport$counter.class observereport$observer.class observereport$reporter.class observereport.class
chmod +x observereport
rm -f observereport$counter.class observereport$observer.class observereport$reporter.class observereport.class
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished observereport.java
```

```
    0.056: main(1): starting
   13.188: observer(8): starting
   13.984: main(1): finished
   14.634: reporter(9): starting
 1015.226: reporter(9): report   0: events:    78
 2015.878: reporter(9): report   1: events:    91
 3016.621: reporter(9): report   2: events:    83
 4017.399: reporter(9): report   3: events:    96
 5018.138: reporter(9): report   4: events:    92
 6019.014: reporter(9): report   5: events:    91
 7020.257: reporter(9): report   6: events:    77
 8020.961: reporter(9): report   7: events:    88
 9021.756: reporter(9): report   8: events:    93
10022.385: reporter(9): report   9: events:    92
10022.942: reporter(9): finished
10023.554: observer(8): finished
```