```java
// $Id: roboclient.java,v 1.5 2012-05-29 20:56:50-07 - - $

//
// Roboclient hostname port username delaysec cycles message message...
//
// The roboclient connects to hostname:port with the username.
// Then it writes messages given by the trailing words in args
// the number of cycles followed by a certain delay in seconds.
// Then it quits.  Useful for testing the server.
//

import java.io.*;
import java.net.*;
import java.util.*;
import static java.lang.System.*;

class roboclient {

   static void quit (String format, Object... params) {
      err.printf (format, params);
      exit (1);
   }

   static String ident (options opts) {
      return String.format ("%s: %s %d", opts.progname,
                            opts.hostname, opts.portnumber);
   }

   static class options {
      final String progname = "roboclient";
      String hostname;
      int portnumber;
      String username;
      long delaysec;
      int cycles;
      String robomessage;
      options (String[] args) {
         try {
            if (args.length < 5) throw new NumberFormatException ();
            hostname = args[0];
            portnumber = Integer.parseInt (args[1]);
            username = args[2];
            delaysec = Long.parseLong (args[3]);
            cycles = Integer.parseInt (args[4]);
            robomessage = "roboclient";
            for (String arg: args) robomessage += " " + arg;
         }catch (NumberFormatException exn) {
            quit ("Usage: %s hostname port username delaysec cycles "
                + "message message%n", progname);
         }
      }
   }


   static class reader implements Runnable {
      Socket socket;
      Scanner scanner;
      reader (Socket _socket, Scanner _scanner) {
         scanner = _scanner;
         socket = _socket;
      }
      public void run () {
         while (! socket.isInputShutdown() && scanner.hasNextLine ()) {
            out.printf ("%s%n", scanner.nextLine ());
         }
         scanner.close ();
      }
   }

   static class writer implements Runnable {
      Socket socket;
      options opts;
      PrintWriter writer;
      writer (Socket _socket, options _opts, PrintWriter _writer) {
         socket = _socket;
         opts = _opts;
         writer = _writer;
      }
      public void run () {
         writer.printf ("%s%n", opts.username);
         writer.flush ();
         for (int count = 0; count < opts.cycles; ++count) {
            if (socket.isOutputShutdown()) break;
            try {
                Thread.currentThread ().sleep (opts.delaysec * 1000);
            }catch (InterruptedException error) {
            }
            writer.printf ("%s%n", opts.robomessage);
            writer.flush ();
         }
         writer.close ();
      }
   }

   public static void main (String[] args) {
      Scanner stdin = new Scanner (System.in);
      options opts = new options (args);
      try {
         Socket socket = new Socket (opts.hostname, opts.portnumber);
         out.printf ("%s: socket OK%n", ident (opts));
         Thread reading = new Thread (new reader (socket,
                          new Scanner (socket.getInputStream ())));
         Thread writing = new Thread (new writer (socket, opts,
                          new PrintWriter (socket.getOutputStream ())));
         reading.start ();
         writing.start ();
      }catch (IOException exn) {
         quit ("%s: %s%n", ident (opts), exn);
      }catch (IllegalArgumentException exn) {
         quit ("%s: %s%n", ident (opts), exn);
      }
   }

}

```