```
// A Peep is a member of the clump
class Peep {
   String name;

   Peep( String n);   // ctor

   String getName();   // getter
}

// A PeepNeighbor is a Peep to whom I am directly connected.
class PeepNeighbor extends Peep {
   String ipAddress;    // data fields
   int port;
   InetAddress realAddress;

   PeepNeighbor( String name, String ip, int port);   // ctor

   String getIpAddress();   // getters
   int getPort();
   InetAddress getInetAddress();
}

// Commands are used to build Clump protocol commands. Commands are
// string-based. Commands have 1) a name, 2) a list of options/value
// pairs and 3) a body.
class Command {
   String name;    // data fields
   ArrayList<String> options;
   String body;
   DatagramPacket packet;    // this is a cleeb

   Command( String name);    // ctor

   String getName();

   void addOption( String option, String value);
   String getOption( String option);

   void addBody( String body);
   String getBody();

   void setPacket( DatagramPacket p);
   DatagramPacket getPacket();

   String toString();   // encode
   static Command parse( String s);    // decode

   byte[] pack();
   static Command unpack( byte[] array);
}
```

```
class ClumpApp {

   // ctor. The firstContact is null if no JOIN to start.
   ClumpApp( String name, int maxNeighbors, String ip, int port,
                  PeepNieghbor firstContact);

   // the biggie… all the action happens here
   void run() {
      // 1. initConnection
      // 2. start a thread with a ClumpAppRcvr in it
      // 3. loop/block to send msgs from console
   }

   private void initConnection() {
      // open UDP socket for b’ness
      // if firstContact
      // 1. send JOIN
      // 2. wait to rcv ACCEPT
      // 3. send NEIGHBORS
   }

   // pack up the command and send it using UDP
   private void sendCommand( Command c, PeepNeighbor to);

   // send the command to all, except Peep who sent it to you
   private void sendQueryFlood( Command c);

   // blocks until you rcv a command
   private Command rcvCommand();

   // execute the given command
   private void doCommand( Command) {
      // switch… and just do it!
   }


   // This inner class is used to rcv UDP segments
   // and pass the commands to the ClumpApp
   class ClumpAppRcvr implements Runnable {
      void run() {
         // loop forever…
         // c = rcvCommand();
         // doCommand( c);
      }
   }


   // the main() I run to execute a ClumpApp
   static void main( String[] args) {
      // The arguments are the important part:
      //    ClumpApp <name> <port> [<fcName> <fcIp> <fcPort>]
      // The “fc” arguments stand for “first contact”.
      // They identify the (name, ip, port) for the first contact peep.
   }
}
```