```
package io.scalecube.examples;

import io.scalecube.cluster.Cluster;
import io.scalecube.transport.Message;

/**
 * Basic example for member transport between cluster members to run the example
Start ClusterNodeA
 * and cluster ClusterNodeB A listen on transport messages B send message to member
A.
 *
 * @author ronen hamias, Anton Kharenko
 */
public class MessagingExample {

  /** Main method. */
  public static void main(String[] args) throws Exception {
    // Start cluster node Alice to listen and respond for incoming greeting
messages
    Cluster alice = Cluster.joinAwait();
    alice
        .listen()
        .subscribe(
            msg -> {
              System.out.println("Alice received: " + msg.data());
              alice.send(msg.sender(), Message.fromData("Greetings from Alice"));
            });

    // Join cluster node Bob to cluster with Alice, listen and respond for incoming
greeting
    // messages
    Cluster bob = Cluster.joinAwait(alice.address());
    bob.listen()
        .subscribe(
            msg -> {
              System.out.println("Bob received: " + msg.data());
              bob.send(msg.sender(), Message.fromData("Greetings from Bob"));
            });

    // Join cluster node Carol to cluster with Alice and Bob
    Cluster carol = Cluster.joinAwait(alice.address(), bob.address());

    // Subscribe Carol to listen for incoming messages and print them to system out
    carol.listen().map(msg -> "Carol received: " +
msg.data()).subscribe(System.out::println);

    // Send from Carol greeting message to all other cluster members (which is
Alice and Bob)
    Message greetingMsg = Message.fromData("Greetings from Carol");
    carol.otherMembers().forEach(member -> carol.send(member, greetingMsg));

    // Avoid exit main thread immediately ]:->
    Thread.sleep(1000);
  }
}
```