Tema 3

Am reusit sa trimit la interval de 1 secunda un heartbeat

package org.example;  
  
import java.io.DataInputStream;  
import java.io.DataOutputStream;  
import java.io.IOException;  
import java.net.ServerSocket;  
import java.net.Socket;  
import java.net.UnknownHostException;  
import java.util.HashMap;  
  
public class HeartBeat implements Runnable{  
  
 // identificare proces in functie de adresa IP  
 static int *self\_id* = 0; // urmeaza identificarea  
 static int *server\_Port* = 3005;  
 String operation; // tip fir RECEIVER, HEARTBEAT  
 static HashMap<Integer, String> *processes* = new HashMap<Integer, String>();  
 static boolean *received* = false;  
 static String *my\_ip*;  
  
 public HeartBeat(String operation) {  
 this.operation = operation;  
 }  
  
 /\* The main() method starts two threads, RECEIVER and HEARTBEAT \*/  
 public static void main(String args[]) throws UnknownHostException, IOException, InterruptedException {  
 Thread.*sleep*(1000);  
 //initialize();  
  
 Runnable receiver = new HeartBeat("receiver");  
 new Thread(receiver).start();  
  
 Runnable heartbeat = new HeartBeat("heartbeat");  
 new Thread(heartbeat).start();  
  
 *processes*.put(1, "localhost");  
  
  
 while (true) {  
// // se identifica adresa IP  
// try (final DatagramSocket socket = new DatagramSocket()) {  
// socket.connect(InetAddress.getByName("8.8.8.8"), 10002);  
// my\_ip = socket.getLocalAddress().getHostAddress();  
// }  
 } // bucla principala  
 }  
  
 public void run() {  
 if (operation.equals("receiver")) {  
 ServerSocket serverSocket = null;  
  
 try {  
 serverSocket = new ServerSocket(*server\_Port*);  
  
 while (true) {  
 Socket socket = serverSocket.accept();  
 System.*out*.println("Connection established.....");  
  
 DataInputStream in = new DataInputStream(socket.getInputStream());  
 String option = in.readUTF();  
  
 if (option.equals("heartbeat")) {  
 int sender = Integer.*parseInt*(option);  
  
 System.*out*.println("HEARTBEAT received from: " + *processes*.get(sender));  
 }  
 socket.close();  
 } // while  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 } else if (operation.equals("heartbeat")) {  
 while (true) {  
 for (int key : *processes*.keySet()) {  
 if (key != *self\_id*) {  
 String destination\_server = *processes*.get(key);  
  
 try {  
 Thread.*sleep*(1250);  
 System.*out*.println("try to check " + destination\_server);  
 Socket socket = new Socket(destination\_server, *server\_Port*);  
 DataOutputStream out = new DataOutputStream(socket.getOutputStream());  
  
 out.writeUTF("heartbeat");  
 out.writeUTF(*self\_id* + "");  
  
 System.*out*.println("Sent HEARTBEAT to: " + destination\_server);  
 } catch (Exception e) {  
 System.*out*.println("\n\*\*\*\tpeer has FAILED!\t\*\*\*");  
 }  
 }  
 }  
 }  
 }  
 }  
}