**INDICACIONS PER REALITZAR LA PRÀCTICA SOBRE JADE.**

1) Tenir clar quin ha de ser el paper de cada agent en el protocol CONTRACT-NET. Definir quins continguts es passaran en cada missatge i com seran interpretats.

2) Cada agent ha de ser una extensió de la classe Agent. En particular ha de tenir definit un mètode anomenat *setup* que és el que s'executarà en el moment d'instanciar la classe i un comportament (d'iniciador per l'agent brossa i de contractat per l'agent robot).

   a) Agent robot

```
import java.io.*;
import java.util.*;

import jade.core.*;
import jade.lang.acl.ACLMessage;
import jade.proto.FipaContractNetResponderBehaviour;
import jade.domain.FIPAException;
import jade.domain.DFServiceCommunicator;
import jade.domain.FIPAAgentManagement.*; // Per DFAgentDescription

public class agentRobot extends Agent {
       // ... Possibles variables
       protected void setup() {...}
       // ... altres mètodes/funcions que es necessitin
       public class meuFipaContractNetResponderBehaviour extends
                      FipaContractNetResponderBehaviour {...}
}
```

   b) Agent brossa

```
import java.io.*;
import java.util.*;

import jade.core.*;
import jade.lang.acl.ACLMessage;
import jade.proto.FipaContractNetInitiatorBehaviour;
import jade.domain.FIPAException;
import jade.domain.DFServiceCommunicator;
import jade.domain.FIPAAgentManagement.*; // Per DFAgentDescription

public class agentBrossa extends Agent {
       // ... Possibles variables
       protected void setup() {...}
       // ... altres mètodes/funcions que es necessitin
       public class meuFipaContractNetInitiatorBehaviour extends
                      FipaContractNetInitiatorBehaviour {...}
}
```

3) Un agent robot haurà de registrar-se a fi de que l'agent brossa sàpiga que està en el sistema. Per enregistrar-se cal fer servir el DF, tal i com es fa en l'exemple 2 (protocols FIPA):

```
private DFAgentDescription dfd = new DFAgentDescription();
// Registre en el DF
      ServiceDescription sd = new ServiceDescription();
      sd.setType("agentRobot");
      sd.setName(getName());
      // sd.setOwnershio("");
      // sd.addOntologies("");
      dfd.setName(getAID());
      dfd.addServices(sd);
      try {
        DFServiceCommunicator.register(this,dfd);
      } catch (FIPAException e) {
        System.out.println(getLocalName() + "fracas en el registre DF. +e.getMessage());
        doDelete();
        }
```

4) Un agent brossa haurà de conèixer la identificació dels agents robots (els seus AID) per tal de poder-los sol·licitar els seus serveis. En l'exemple de protocols trobareu com es fa per un únic agent. Per tots els agents podeu seguir el mateix model, iterant el tractament tal i com es mostra a continuació:

```
DFAgentDescription actual;
DFAgentDescription dfad= new DFAgentDescription();
ServiceDescription sd= new ServiceDescription();
SearchConstraints c= new SearchConstraints();

sd.setType("agentRobot");
dfad.addServices(sd);
try {  // Esperem que hi hagi algun agent en el sistema
       while  (true) {
          System.out.println(getLocalName() + "espera un robot registrat en el
                             directori DF");
          List result=DFServiceCommunicator.search(this,getDefaultDF(),dfad,c);
          if (result.size() > 0) {
            for ( int j=0; j<=result.size()-1; j++) {
                actual =(DFAgentDescription)result.get(j);
                cfp.addReceiver(actual.getName());
             };
            break;
           }
           Thread.sleep(10000);
         }  // while true
      } catch (Exception exc) {System.out.println(exc.toString()); }
```

5) Cada agent, ha de tenir definit el seu corresponent comportament basat en els FipaContractNet...Behaviour. Així:
   a) Un agent robot estendrà la classe FipaContractNetResponderBehaviour
   b) Un agent brossa estendrà la classe FipaContractNetInitiatorBehaviour.

En el manual "Complete API documentation" (on-line des de la docs→index.html de JADE ) teniu descrit aquests comportaments, i explicitat clarament quins mètodes s'han de definir en cadascun d'ells. En particular

a) Un agent robot ha de tenir els mètodes següents:

```
public class meuFipaContractNetResponderBehaviour extends
                FipaContractNetResponderBehaviour {
// Constructor
       public meuFipaContractNetResponderBehaviour (Agent a) {
        super(a);
       }
//Mètodes
       public ACLMessage handleCfpMessage(ACLMessage cfp) {...}

       public ACLMessage handleAcceptProposalMessage(ACLMessage msg) {...}
       public void handleRejectProposalMessage (ACLMessage msg) {...}
       public void handleOtherMessages (ACLMessage msg) {...}
    }
```

b) Un agent brossa ha de tenir els mètodes següents:

```
public class meuFipaContractNetInitiatorBehaviour extends
FipaContractNetInitiatorBehaviour {
// Constructor
    public meuFipaContractNetInitiatorBehaviour (Agent a, ACLMessage msg,
                                                 List group) {
         super(a,msg,group);
       }
// Mètodes
         public Vector handleProposeMessages (Vector proposals) {...}
         public void handleOtherMessages (ACLMessage msg) {...}
         public Vector handleFinalMessages (Vector messagers) {...}
         public String createCfpContent (String cfpContent, String receiver) {...}
```

6) D'aquesta manera, els *setup* de cada tipus d'agents, hauran d'incloure com a darrera instrucció afegir el comportament corresponent. Aquest serà el comportament que s'activarà quan els hi arribi un missatge.

   a) Agent robot:

```
protected void setup() {
 ...
addBehaviour(new meuFipaContractNetResponderBehaviour(this));
    }
```

   b) Agent brossa:

```
protected void setup() {
     ...
     addBehaviour (new meuFipaContractNetInitiatorBehaviour (agentBrossa.this, cfp,
group));
}
```

7) Finalment, recordar que l'agent brossa ha de ser iteratiu, ja que anirà generant una brossa darrera de l'altre: quan un robot hagi confirmat que la brossa ha estat recollida, aleshores en generarà una altra. Per això cal que inclogueu el mètode reset en el constructor del comportament tal i com s'indica en el comportament FipaContractNetInitiatorBehavior.
   Notar que és important la seqüencialitat en la generació de la brossa.