

Consignes pour finir le TP qui consiste à implémenter un "Contract Net Protocol" simple : Le 26/11/2019

S'inspirer des classes `examples.protocols.FIPAResponseInitiatorAgent`, `examples.protocols.FIPAResponseResponderAgent`, `examples.protocols.ContractNetResponderAgent` et `examples.protocols.ContractNetInitiatorAgent`, pour créer vos propres classes à compiler séparément. Vous les adapterez à votre mini-projet dans un second temps.

Pour cela :

Copier les fichiers `FIPAResponseInitiatorAgent.java`, `FIPAResponseResponderAgent.java`, `ContractNetResponderAgent.java` et `ContractNetInitiatorAgent.java` dans un dossier nommé (par exemple) `JINProject` dans le dossier `examples`.

Renommer les fichiers (et donc les classes) en ajoutant (par exemple) le préfixe `JIN`.

Cela donnera : `JINFIPAResponseInitiatorAgent.java`, `JINFIPAResponseResponderAgent.java`, `JINContractNetResponderAgent.java` et `JINContractNetInitiatorAgent.java`

Penser à modifier le nom du package de chaque classe comme suit :

```
7  * Lesser General Public License for more details.
8  *
9  * You should have received a copy of the GNU Lesser General Public
10 * License along with this library; if not, write to the
11 * Free Software Foundation, Inc., 59 Temple Place - Suite 330,
12 * Boston, MA 02111-1307, USA.
13 * *****
14 */
15 package examples.JINProject;
16
17 import jade.core.Agent;
18 import jade.lang.acl.ACLMessage;
19 import jade.lang.acl.MessageTemplate;
20 import jade.proto.ContractNetResponder;
21 import jade.domain.FIPANames;
22 import jade.domain.FIPAAgentManagement.NotUnderstoodException;
23 import jade.domain.FIPAAgentManagement.RefuseException;
24 import jade.domain.FIPAAgentManagement.FailureException;
25
26 /**
27  * This example shows how to implement the responder role in
28  * a FIPA-contract-net interaction protocol. In this case in particular
29  * we use a <code>ContractNetResponder</code>
30  * to participate into a negotiation where an initiator needs to assign
31  * a task to an agent among a set of candidates.
32  * @author Giovanni Caire - TILAB
33  */
34 public class JINContractNetResponderAgent extends Agent {
35
36     protected void setup() {
37         System.out.println("Agent "+getLocalName()+" waiting for CFP...");
38         MessageTemplate template = MessageTemplate.and(
39             MessageTemplate.MatchProtocol(FIPANames.InteractionProtocol.FIPA_CONTRACT_NET),
40             MessageTemplate.MatchPerformative(ACLMessage.CFP) );
41     }
42
43     public void onReceive(ACLMessage msg) {
44         if (msg.isPerformative(ACLMessage.CFP)) {
45             ContractNetResponder cnp = new ContractNetResponder(this, template);
46             cnp.receive(msg);
47         }
48     }
49 }
```

Pour compiler :

Se positionner dans le dossier parent de `examples` :

```
javac examples/JINProject/JINContractNetResponderAgent.java
```

ce qui génère un fichier `JINContractNetResponderAgent.class`

Par contre, la commande :

```
javac examples/JINProject/JINContractNetInitiatorAgent.java
```

```
MacBook-Pro-de-Nadia:src nadiaabchiche-mimouni$ javac examples/JINProject/JINContractNetResponderAgent.java
MacBook-Pro-de-Nadia:src nadiaabchiche-mimouni$ javac examples/JINProject/JINContractNetInitiatorAgent.java
Note: examples/JINProject/JINContractNetInitiatorAgent.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
MacBook-Pro-de-Nadia:src nadiaabchiche-mimouni$
```

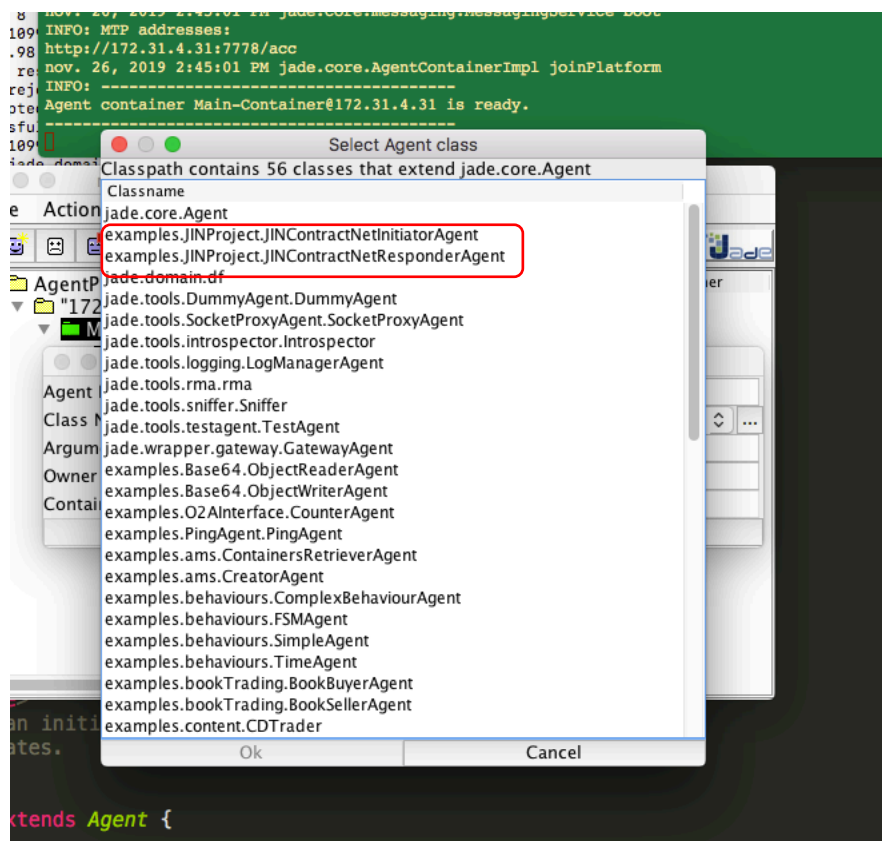
génère un warning, dont il est possible de visualiser les détails avec l'option `-Xlint`
Cependant, cela n'empêche pas la génération du `.class`

```
MacBook-Pro-de-Nadia:~ nadiaabchiche-mimouni$ cd jade/src/
MacBook-Pro-de-Nadia:src nadiaabchiche-mimouni$ ls -l examples/JINProject/
total 72
-rw-r--r--  1 nadiaabchiche-mimouni  staff   192 26 nov 12:55 APDescription.txt
-rw-r--r--  1 nadiaabchiche-mimouni  staff  3231 26 nov 14:43 JINContractNetInitiatorAgent$.class
-rw-r--r--  1 nadiaabchiche-mimouni  staff  2033 26 nov 14:43 JINContractNetInitiatorAgent.class
-rw-r--r--  1 nadiaabchiche-mimouni  staff  4960 26 nov 14:43 JINContractNetInitiatorAgent.java
-rw-r--r--  1 nadiaabchiche-mimouni  staff  3074 26 nov 14:34 JINContractNetResponderAgent$.class
-rw-r--r--  1 nadiaabchiche-mimouni  staff  1967 26 nov 14:34 JINContractNetResponderAgent.class
-rw-r--r--  1 nadiaabchiche-mimouni  staff  4080 26 nov 14:22 JINContractNetResponderAgent.java
-rw-r--r--  1 nadiaabchiche-mimouni  staff    29 26 nov 12:55 MTPs-Main-Container.txt
```

ATTENTION : pour créer des agents depuis JADE, il faut ajouter le dossier de vos nouvelles classes dans CLASPPATH

```
Last login: Tue Nov 26 12:59:11 on ttys002
MacBook-Pro-de-Nadia:~ nadiaabchiche-mimouni$ echo $CLASSPATH
/Users/nadiaabchiche-mimouni/jade/lib/jade.jar:/Users/nadiaabchiche-mimouni/jade/lib/jadeExamples.jar:/Users/
mmons-codec/commons-codec-1.3.jar:/Users/nadiaabchiche-mimouni/jade/src/
```

On peut lancer les agents dans JADE comme cela a été fait pour les classes initiales définies dans le package "examples.protocols"



L'exécution avec JADE :

The screenshot displays the JADE environment during the execution of a multi-agent system. The interface is divided into several panels:

- Console Window (Top Right):** Shows the boot process of the JADE system. Key messages include:
 - INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl\$JAXPSAXParser
 - INFO: MTP addresses;
 - http://172.31.4.31:7778/acc
 - nov. 26, 2019 2:45:01 PM jade.core.AgentContainerImpl joinPlatform
 - INFO: Agent container Main-Container@172.31.4.31 is ready.
 - Agent AR1 waiting for CFP...
 - Agent AR2 waiting for CFP...
 - Trying to delegate dummy-action to one out of 2 responders.
 - Agent AR1: CFP received from InitAG@172.31.4.31:1099/JADE. Action is dummy-action
 - Agent AR1: Proposing 9
 - Agent AR2: CFP received from InitAG@172.31.4.31:1099/JADE. Action is dummy-action
 - Agent AR1@172.31.4.31:1099/JADE proposed 9
 - Agent AR2@172.31.4.31:1099/JADE refused
 - Accepting proposal 9 from responder AR1@172.31.4.31:1099/JADE
 - Agent AR1: Proposal accepted
 - Agent AR1: Action successfully performed
 - Agent AR1@172.31.4.31:1099/JADE successfully performed the requested action
- Graphical Message Exchange Diagram (Top Left):** Visualizes the communication between agents. It shows a sequence of messages:
 - 0: CFP(9, 0, 3, 0) from Other to AR1
 - 1: CFP(9, 0, 4, 1) from Other to AR2
 - 2: PROPOSE(9, 0, 240, 3, 0) from AR1 to AR2
 - 3: REFUSED(9, 0, 242, 4, 1) from AR2 to AR1
 - 4: ACCEPT-PROPOSAL(9, 0, 1, 0, 240) from AR2 to AR1
 - 5: INFORM(9, 0, 272, 0) from AR1 to Other
 - 6: INFORM(9, 0, 272, 0) from AR2 to Other
- Agent Platforms List (Bottom Left):** A table listing the agents and their properties.

Agent	Class	Argument	Owner	Container
jade.tools.logging.LogManagerAgent	jade.tools.rma.rma	jade.tools.sniffer.Sniffer	jade.tools.testagent.TestAgent	jade.wrapper.gateway.GatewayAgent
examples.Base64.ObjectReaderAgent	examples.Base64.ObjectWriterAgent	examples.OZAlInterface.CounterAgent	examples.PingAgent.PingAgent	examples.ams.ContainersRetrieverAgent
examples.ams.CreatorAgent	examples.behaviours.ComplexBehaviourAgent	examples.behaviours.FSMAgent		
- Remote Agent Management GUI (Bottom Right):** A window titled "rma@172.31.4.31:1099/JADE - JADE Remote Agent Management GUI" showing a list of agent platforms. The "Main-Container" is expanded, showing a list of agents:
 - AR1@172.31.4.31:1099/JADE
 - AR2@172.31.4.31:1099/JADE
 - InitAG@172.31.4.31:1099/JADE
 - ams@172.31.4.31:1099/JADE
 - df@172.31.4.31:1099/JADE
 - rma@172.31.4.31:1099/JADE
 - sniffer0-on-Main-Container
 - sniffer0@172.31.4.31:1099/JADE

Idem pour les classes JINFIPARRequestInitiatorAgent, JINFIPARRequestResponderAgent
 Pour le TP/mini projet, c'est plus simple de partir de ces 2 dernières classes.