Упражнение 4: Комуникации в JADE. Изпращане и получаване на съобщение

Съобщенията в JADE са имплементирани като обект на класа jade.lang.acl.ACLMessage, който предоставя get и set методи за достъпване на всички полета описани във формата.

Изпращането на съобщения до друг агент е сведено до попълване на полетата на съобщението и извикването на метода *send*() в класа на агента. Пример за структурата на съобщение в JADE.

```
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID("Peter", AID.ISLOCALNAME));
msg.setLanguage("English");
msg.setOntology("Timetable-ontology");
msg.setContent("19:30 – meeting Tom");

send(msg);
```

При получаване на съобщение, то постъпва в кутията на получателя. За да вземе дадено съобщение от кутията, получателят извиква метода *receive*(). Този метод връща първото съобщение в пощенската кутия или *null* ако няма съобщения.

```
ACLMessage msg = receive();
if (msg != null) {
        // Process the message

}
```

За да се демонстрира комуникацията между агентите, ще се реализират два агента, които да разговарят чрез предаване на съобщение, което се имплементира от класа *jade.lang.acl.ACLMessage*.

Първият агент *SenderAgent* изпраща съобщение до агент с име *Peter*. Вторият агент *ReceiverAgent* реализира поведение, което изчаква получаването на съобщение и го разпечатва на екрана.

Агентът *SenderAgent* създава инстанция на класа *ACLMessage* и задава атрибутите на съобщението, след което го изпраща с метода *send(ACLMessage m)*.

```
import jade.core.AID;
import jade.core.Agent;
import jade.lang.acl.ACLMessage;

public class SenderAgent extends Agent {
        protected void setup() {
                System.out.println("Hello. My name is " +
                                                        this.getLocalName());
                sendMessage();
        }

        private void sendMessage() {
                AID r = new AID("Peter", AID.ISLOCALNAME);
```

```
            ACLMessage aclMessage = new
                                    ACLMessage(ACLMessage.REQUEST);
            aclMessage.addReceiver(r);
            aclMessage.setContent("Hello! How are you?");
            this.send(aclMessage);
        }
}
```

Получаването на съобщението се реализира чрез поведение, което се добавя в setup() метода:

```
public class ReceiverAgent extends Agent {
        protected void setup() {
            System.out.println("Hello. My name is " +
                                                        this.getLocalName());
            addBehaviour(new ResponderBehaviour(this));
        }
}
```

Вторият агент *ResponderBehaviour* е наследник на абстрактния клас *SimpleBehaviour* class, на който са пренаписани методите *action*() и *done*(). Методът action() обработва постъпилите съобщения. Методът *done*() връща стойност *true* след като е приключило изпълнението на агента. Трябва да се укаже и какъв тип съобщение се обработва. В този пример това е съобщение от вида REQUEST, което се задава от класа *MessageTemplate*.

```
class ResponderBehaviour extends SimpleBehaviour {
        private static final MessageTemplate mt = MessageTemplate
                    .MatchPerformative(ACLMessage.REQUEST);

        public ResponderBehaviour(Agent agent) {
            super(agent);
        }

        public void action() {
            ACLMessage aclMessage = myAgent.receive(mt);
            if (aclMessage != null) {
                System.out.println(myAgent.getLocalName()
                                + ": I receive message.\n" +
                                                        aclMessage);
            } else {
                this.block();
            }
        }

        public boolean done() {
            return false;
        }
}
```

Задача 1:

Модифицирайте задачата 1 от упражнение 3 така, че агента *EntryGUIAgent* да предава съобщение с името на ястието на агент *MealAgent*. Дефинирайте подходящото поведение за агента *MealAgent*, който да обхожда и претърсва колекцията с рецепти и да намиране съвпадение.

Схемата на комуникацията между двата агента е дадена на фигурата по-долу: