



Modeling & simulation exam (2nd years of master's IDO)

First name :Last name :

/20

➤ Course Comprehension Questions (12 pts) : put a cross on the correct answers

1- Modeling of a system is:
- A step in the simulation X
- Equivalent to the simulation
- Independent to the simulation
2- A behavior of an agent must contain:
- A setup method
- An action method X
- A done method
3- Simulation is appropriate for :
- Studying the behavior of a complex system X
- Testing scenarios without real risks X
- Analyzing systems that are difficult or impossible to observe directly X
4- Dynamic discrete events system is :
- Discontinuous system X
- Continious system
- Deterministic systems
5- The descriptive methods are :
- Modeling methods X
- Simulation methods
- Testing methods
6- A discrete event system is based on the concepts of :
- Event, Activity, Process X
- Event, Entity, Process
- Event, Activity, Object
7- Multi-Agent simulation is well-suited for:
- Complex systems X
- Distributed systems X
- Intelligent systems X
8- A MAS (Multi-Agent System) composed of reactive agents requires:
- An average number of agents
- A small number of agents
- A large number of agents X
9- Point-to-point message communication requires :
- Cognitive agents X
- Reactive agents
- Hybrid agents
10- Autonomy for an agent is :
- An interesting characteristic
- A mandatory characteristic X
- An optional characteristic
11- Agent-oriented simulation arises from:
- The interaction between different agents X
- The individual functioning of each agent
- The communication between different agents
12- JADE platform provides :
- Containers X

- Debugging agents X
- Management agents X

➤ **Exercise 1 (5 pts):** Explain the result of this code (step by step)

```
public class A1 extends Agent{
    public void setup(){
        this.addBehaviour(new OneShotBehaviour(){
            public void action(){
                System.out.println("I want to play !");
                ACLMessage Msg = new ACLMessage();
                Msg.setContent("Go");
                Msg.setSender(this.myAgent.getAID());
                Msg.addReceiver(new AID(new String("B1"),
                    AID.ISLOCALNAME));
                this.myAgent.send(Msg); } } ); } }
    public class B1 extends Agent{
        public void setup(){
            this.addBehaviour(new CyclicBehaviour (){
                public void action(){
                    ACLMessage Msg = this.myAgent.receive();
                    if (Msg != null)
                        { System.out.println("a message is received from the
                            agent: "+ this.myAgent.getLocalName());
                          if (Msg.getContent().equals("Go"))
                              { System.out.println("the message received is: Go");
                                ACLMessage Msg1 = Msg.createReply();
                                Msg1.setContent("OK");
                                this.myAgent.send(Msg1);
                                System.out.println("a messageOK is sent ");
                              } } else{ block(); } } }
                }
```

- The agent A1 has a OneShot behaviour
- The agent A1 print **I want to play**
- The agent A1 create an ACL message
- The agent A1 set the content of the message to **Go**
- The agent A1 set the sender's address to his AID
- The agent A1 set the receiver's address to the AID of the agent B1
- The agent A1 send the message created
-
- The agent B1 has a cyclicBehaviour
- The agent B1 receive a message from A1
- The agent B1 test the received message
if Msg is not empty then
 - B1 print a message is received from the agent A1
 - B1 test if the content of Msg equals Go then
B1 print the message received is: Go
B1 create a reply
B1 set the content of the reply to OK
B1 send the reply
B1 print a message OK is sent
- Else
 - B1 execute block()

➤ **Exercise 2 (3 pts)** give the result of each instruction **print**

```
fruits = ['apple', 'banana', 'cherry', 'date']
print("First fruit:", fruits[0])
print("Last fruit:", fruits[-1])
fruits.append('elderberry')
print("After", fruits)
fruits.insert(2, 'fig')
print("After:", fruits)
fruits.remove('banana')
print("After:", fruits)
popped_fruit = fruits.pop(1)
print("After popping an element:", fruits)
print("Length of the fruits list:", len(fruits))
```

First fruits : appple
 Last fruit : date
 After ['apple', 'banana', 'cherry', 'date', 'elderberry']
 After:['apple','banana','fig','cherry','date', elderberry']
 After:['apple','fig','cherry','date', elderberry']
 After popping an element: ['apple', 'cherry', 'date', 'elderberry']
 Length of the fruits list: 4

Good Luck