/\*\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* JADE - Java Agent DEvelopment Framework is a framework to develop

\* multi-agent systems in compliance with the FIPA specifications.

\* Copyright (C) 2000 CSELT S.p.A.

\*

\* GNU Lesser General Public License

\*

\* This library is free software; you can redistribute it and/or

\* modify it under the terms of the GNU Lesser General Public

\* License as published by the Free Software Foundation,

\* version 2.1 of the License.

\*

\* This library is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

\* Lesser General Public License for more details.

\*

\* You should have received a copy of the GNU Lesser General Public

\* License along with this library; if not, write to the

\* Free Software Foundation, Inc., 59 Temple Place - Suite 330,

\* Boston, MA 02111-1307, USA.

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

package examples.behaviours;

import jade.core.Agent;

import jade.core.behaviours.Behaviour;

import jade.core.behaviours.CyclicBehaviour;

import jade.core.behaviours.OneShotBehaviour;

/\*\*

\* This example shows the basic usage of JADE behaviours.<br>

\* More in details this agent executes a <code>CyclicBehaviour</code> that shows

\* a printout at each round and a generic behaviour that performs four successive

\* "dummy" operations. The second operation in particular involves adding a

\* <code>OneShotBehaviour</code>. When the generic behaviour completes the

\* agent terminates.

\* @author Giovanni Caire - TILAB

\*/

public class SimpleAgent extends Agent {

protected void setup() {

System.out.println("Agent "+getLocalName()+" started.");

// Add the CyclicBehaviour

addBehaviour(new CyclicBehaviour(this) {

public void action() {

System.out.println("Cycling");

}

});

// Add the generic behaviour

addBehaviour(new FourStepBehaviour());

}

/\*\*

\* Inner class FourStepBehaviour

\*/

private class FourStepBehaviour extends Behaviour {

private int step = 1;

public void action() {

switch (step) {

case 1:

// Perform operation 1: print out a message

System.out.println("Operation 1");

break;

case 2:

// Perform operation 2: Add a OneShotBehaviour

System.out.println("Operation 2. Adding one-shot behaviour");

myAgent.addBehaviour(new OneShotBehaviour(myAgent) {

public void action() {

System.out.println("One-shot");

}

});

break;

case 3:

// Perform operation 3: print out a message

System.out.println("Operation 3");

break;

case 4:

// Perform operation 3: print out a message

System.out.println("Operation 4");

break;

}

step++;

}

public boolean done() {

return step == 5;

}

public int onEnd() {

myAgent.doDelete();

return super.onEnd();

}

} // END of inner class FourStepBehaviour

}