```
code : AgentPi


Afficheur.java
import org.javact.util.BehaviorProfile;

public interface Afficheur extends BehaviorProfile {

}
************************************************************
AgentTourisk.java

import org.javact.util.ActorProfile;

public interface AgentTourisk extends ActorProfile, Recherche,
  Superviseur, Afficheur {

}


************************************************************
Recherche.java

import org.javact.util.ActorProfile;
import org.javact.util.BehaviorProfile;
import org.javact.util.StandAlone;

public interface Recherche extends BehaviorProfile {
    public void become(Superviseur beh);
}

************************************************************

Superviseur.java

import org.javact.util.BehaviorProfile;
import org.javact.util.StandAlone;

public interface Superviseur extends BehaviorProfile, StandAlone {
    public void become(Afficheur beh);
}

************************************************************
Tourisk.java

import org.javact.util.ActorProfile;

public interface Tourisk extends Recherche, ActorProfile {

}


```

```
53 ************************************************************
54 Skeleton1.java
55
56 import java.lang.management.ManagementFactory;
57 import java.lang.management.OperatingSystemMXBean;
58 import java.math.BigDecimal;
59
60 import org.javact.net.rmi.CreateCt;
61 import org.javact.util.StandAlone;
62
63 /**
64  * Behaviour for the actor Recherche
65  */
66 class RechercheBeh extends RechercheQuasiBehavior implements StandAlone {
67
68     String[] itineraire;
69     int position;
70     String firstPlace=null;
71     String bestPlace=null;
72     double bestLoad=1;
73
74     public RechercheBeh() {
75         itineraire = JavActProbe.probe(2000);
76         position = 0;
77     }
78
79     @Override
80     public void run() {
81         System.out.println("Je suis sur " + myPlace());
82
83         if (firstPlace==null) {
84             firstPlace=myPlace();
85         }
86
87         /* cas d'arret */
88         if (position == itineraire.length - 1) {
89             become(new SuperviseurBeh(firstPlace));
90             go(bestPlace);
91         } else {
92             /* cas de base */
93             OperatingSystemMXBean bean =
   ManagementFactory.getOperatingSystemMXBean();
94             double load = bean.getSystemLoadAverage();
95             System.out.println(load);
96             if (load<bestLoad) {
97                 bestLoad=load;
98                 bestPlace=myPlace();
99             }
100            go(itineraire[position++] + ":2000");
101        }
102    }
103 }
104
```

```
105 /**
106 * Behaviour for the actor Superviseur
107 */
108 class SuperviseurBeh extends SuperviseurQuasiBehavior {
109 
110     private String firstPlace;
111 
112     public SuperviseurBeh(String firstPlace) {
113         this.firstPlace=firstPlace;
114     }
115 
116     public void run() {
117         System.out.println("Je fais mon calcul sur la machine
    "+myPlace());
118         BigDecimal result=new Pi(10000).call();
119         become(new AfficheurBeh(result));
120         go(firstPlace);
121     }
122 
123 }
124 
125 /**
126 * Behaviour for the actor Afficheur
127 */
128 class AfficheurBeh extends AfficheurQuasiBehavior implements StandAlone {
129     private BigDecimal result;
130 
131     public AfficheurBeh(BigDecimal result) {
132         this.result=result;
133     }
134 
135     @Override
136     public void run() {
137         System.out.println("Je suis rentré à la maison ! "+ myPlace());
138         System.out.println(result);
139     }
140 }
141 
142 public class Skeleton1 {
143     public static void main(String[] args) {
144         CreateCt.STD.create("localhost:2000", new RechercheBeh());
145     }
146 }
147 
148 
149 fin code : AgentPi
150 ************************************************************
151 
```

```
1 hello_correction.java
2
3 ************************************************
4 Hello.java
5
6 import org.javact.util.ActorProfile;
7 import org.javact.util.BehaviorProfile;
8
9 /**
10 * Behaviour for the actor Hello
11 */
12
13
14
15 ************************************************
16 HelloQuasiBehavior.java
17
18 import org.javact.lang.*;
19 import org.javact.local.CreateCt;
20 import org.javact.local.SendCt;
21 import org.javact.util.StandAlone;
22
23 public abstract class HelloQuasiBehavior extends QuasiBehavior implements
   Hello
24 {
25 }
26
27
28 ************************************************************************
29
30 public class JAMstop implements org.javact.lang.Message
31 {
32     private int signatureNumber ;
33
34
35     public JAMstop()
36     {
37         signatureNumber = 0 ;
38     }
39
40     public final void handle(org.javact.lang.QuasiBehavior _behavior)
41     {
42         switch (signatureNumber)
43         {
44             case 0 :
45                 if ( _behavior instanceof Hello)
46                     ((Hello) _behavior).stop() ;
47                 else
48                     throw new org.javact.lang.MessageHandleException() ;
49                 break ;
50             default :
51                 throw new org.javact.lang.MessageHandleException() ;
52         }
```

```
53      }
54  }
55
56  ************************************************************************
    ****
57
58  public class JAMhello implements org.javact.lang.Message
59  {
60      private int signatureNumber ;
61
62
63      public JAMhello()
64      {
65          signatureNumber = 0 ;
66      }
67
68      public final void handle(org.javact.lang.QuasiBehavior _behavior)
69      {
70          switch (signatureNumber)
71          {
72              case 0 :
73                  if (_behavior instanceof Hello)
74                      ((Hello) _behavior).hello() ;
75                  else
76                      throw new org.javact.lang.MessageHandleException() ;
77                  break ;
78              default :
79                  throw new org.javact.lang.MessageHandleException() ;
80          }
81      }
82  }
```

```
 1
 2 code Mobile chat
 3 ***************************************************************************
 4 Mobile Chat.java
 5
 6 import org.javact.lang.Agent;
 7 import org.javact.util.ActorProfile;
 8 import org.javact.util.BehaviorProfile;
 9
10 public interface MobileChat extends BehaviorProfile, ActorProfile{
11     public void speak(String m);
12     public void setDest(Agent a);
13     public void move(String p, String text);
14 }
15
16
17 ***************************************************************************
18 MobileChatSwing.java
19
20 import javax.swing.BoxLayout;
21 import javax.swing.JComboBox;
22 import javax.swing.JFrame;
23 import javax.swing.JOptionPane;
24 import javax.swing.JPanel;
25 import javax.swing.JTextArea;
26 import javax.swing.JTextField;
27
28 import org.javact.lang.Agent;
29 import org.javact.net.rmi.SendCt;
30
31 /**
32  *
33  * @author leriche - @ENAC - 2017
34  */
35 public class MobileChatSwing {
36
37     private JTextArea ta;
38     private Agent myAgent;
39
40     /**
41      * @param beh
42      *            le comportement de l'agent mobile
43      * @param list
44      *            le domaine de places accessibles
45      * @param text
46      *            l'état du texte
47      */
48     public MobileChatSwing(MobileChatBeh beh, Agent myAgent, String[]
   list, String text) {
49         this.myAgent = myAgent;
50         JFrame frame = new JFrame("Mobile Chat @ ENAC");
51
52         frame.setSize(300, 400);
```

```
53          frame.setResizable(false);
54          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
55          frame.setLocationRelativeTo(null);
56          JPanel panel = new JPanel();
57          frame.setContentPane(panel);
58
59          ta = new JTextArea(text);
60          ta.setEditable(false);
61
62          JTextField field = new JTextField();
63
64          JComboBox<String> box = new JComboBox<>();
65          if (list != null) {
66              for (String s : list) {
67                  box.addItem(s);
68              }
69          }
70
71          panel.setLayout(new BoxLayout(panel, BoxLayout.PAGE_AXIS));
72          panel.add(ta);
73          panel.add(field);
74          panel.add(box);
75
76          field.addActionListener((ActionListener) -> {
77              ta.insert("\nMoi : " + field.getText(),
    ta.getText().length());
78              if (beh!=null) {
79                  beh.send(field.getText());
80              } else {
81                  System.err.println("Behavior not set (yet?)");
82              }
83              field.setText("");
84          });
85
86          box.addActionListener((ActionListener) -> {
87              JOptionPane.showMessageDialog(panel, "Je vais me déplacer sur
    " + box.getSelectedItem().toString());
88              SendCt.STD.send(new JAMmove(box.getSelectedItem().toString(),
    ta.getText()), myAgent);
89              frame.dispose();
90          });
91
92          frame.setVisible(true);
93      }
94
95      /**
96       *
97       * @param msg
98       *            le message à ajouter dans le textarea
99       */
100     public void receive(String msg) {
101         ta.insert("\n" + msg, ta.getText().length());
102     }
```

```
103
104     /**
105      * pour tester en statique seulement
106      *
107      * @param args
108      */
109     public static void main(String[] args) {
110         String[] l={"localhost"};
111         new MobileChatSwing(null, null, l, null);
112     }
113
114 }
115
116
117 ********************************************************************************
118 Skeleton1.java
119
120
121 /*
122 *
    ############################################################################
    #
123 * JavAct: A Java(TM) library for distributed and mobile actor-based
    computing
124 * Copyright (C) 2001-2007 I.R.I.T./C.N.R.S.-I.N.P.T.-U.P.S.
125 *
126 * This library is free software; you can redistribute it and/or modify it
127 * under the terms of the GNU Lesser General Public License as published by
128 * the Free Software Foundation; either version 2.1 of the License, or any
129 * later version.
130 *
131 * This library is distributed in the hope that it will be useful, but
132 * WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY
133 * or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public
134 * License for more details.
135 *
136 * You should have received a copy of the GNU Lesser General Public License
137 * along with this library; if not, write to the Free Software Foundation,
138 * Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA
139 *
140 * Initial developer(s): The I.A.M. Team (I.R.I.T.) - SMAC Team (IRIT)
    since 2007
141 * Contributor(s): The I.A.M. Team (I.R.I.T.) - SMAC Team (IRIT) since 2007
142 * Contact: javact@irit.fr
143 *
    ############################################################################
    #
144 */
145
146 import org.javact.lang.*;
147 import org.javact.net.rmi.CreateCt;
```

```
148 import org.javact.net.rmi.SendCt;
149 import org.javact.util.StandAlone;
150
151 /**
152 * Behaviour for the actor MobileChat
153 */
154 class MobileChatBeh extends MobileChatQuasiBehavior implements StandAlone
    {
155
156     private Agent dest;
157     private String text;
158     private transient MobileChatSwing ihm;
159     private String[] list = JavActProbe.probe(1099);
160
161     @Override
162     public void run() {
163         System.out.println("sur "+myPlace());
164         ihm = new MobileChatSwing(this, ego(), list, text);
165     }
166
167     public void setDest(Agent dest) {
168         this.dest=dest;
169     }
170
171     public void move(String place, String text2) {
172         text=text2;
173         go(place);
174     }
175
176     public void send(String msg) {
177         send(new JAMspeak(msg), dest);
178     }
179
180     public void speak(String msg) {
181         ihm.receive(msg);
182     }
183
184 }
185
186 public class Skeleton1 {
187
188     public static void main(String[] args) {
189         Agent a1= CreateCt.STD.create("localhost",new MobileChatBeh());
190         Agent a2 = CreateCt.STD.create("localhost",new MobileChatBeh());
191         SendCt.STD.send(new JAMsetDest(a2),a1);
192         SendCt.STD.send(new JAMsetDest(a1),a2);
193     }
194
195 }
196
197 fin code Mobile chat
198   ****************************************************************
```