

Urszula Ślusarz

Inżynieria Obliczeniowa, WIMiP

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

*Rozproszona sztuczna inteligencja*

## Platformy agentowe w języku Java Zachowania (1)

### 1. Utwórz klasę agenta o nazwie klasa\_1. Agent ten:

- zawsze na samym początku wypisuje „startuję”,
- zawsze przed swoim usunięciem wypisuje „zaraz się usunę”.

```
1 import jade.core.Agent;
2 public class klasa_1 extends Agent {
3     protected void setup() {
4         System.out.println("startuję!");
5     }
6     protected void takeDown() {
7         System.out.println("zaraz się usunę!");
8     }
9 }
10
```

**takeDown()**-Metoda ta jest wywoływana tuż przed zakończeniem działania agenta w celu wykonania różnych operacji czyszczenia.

The screenshot displays the JADE Remote Agent Management GUI on the left and its corresponding terminal output on the right.

**JADE Remote Agent Management GUI:**

- Window title: rma@192.168.1.110:1099/JADE - JADE Remote Agent Management GUI
- Menu: File, Actions, Tools, Remote Platforms, Help
- Tree view (AgentPlatforms):
  - "192.168.1.110:1099/JADE"
    - Main-Container
      - Klasa1@192.168.1.110:1099/
      - ams@192.168.1.110:1099/JADE
      - df@192.168.1.110:1099/JADE
      - rma@192.168.1.110:1099/JADE
- Table (Agent Details):

name	addresses	state	owner
NAME	ADDRES...	STATE	OWNER

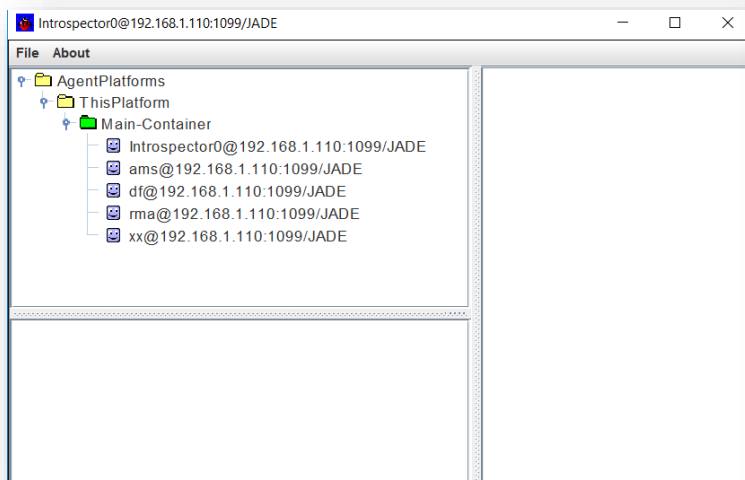
**Terminal Output:**

```
D:\Studia\Java_projekty\RSI\lab7\src>javac klasa_1.java
D:\Studia\Java_projekty\RSI\lab7\src>java jade.Boot -gui Klasa1:klasa_1
kwi 29, 2018 1:24:51 PM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
kwi 29, 2018 1:24:52 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.1.110:1099
kwi 29, 2018 1:24:52 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
kwi 29, 2018 1:24:52 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
kwi 29, 2018 1:24:52 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
kwi 29, 2018 1:24:52 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
kwi 29, 2018 1:24:52 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
kwi 29, 2018 1:24:52 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
kwi 29, 2018 1:24:52 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DESKTOP-11C2852.local:7778/acc
startuję!
kwi 29, 2018 1:24:52 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@192.168.1.110 is ready.
-----
zaraz się usunę!
```

## 2. Utworzyłam klasę agenta o nazwie klasa\_2 na podstawie kodu klasa\_1.

Do agenta dodałam zachowanie polegające na jednokrotnym wykonaniu operacji wypisania na ekranie słowa „wykonuję”. Uruchomiłam agenta introspektora.

```
7
8 protected void setup() {
9     System.out.println("startuje!");
10
11     addBehaviour(new OneShotBehaviour() {
12         @Override
13         public void action() {
14             System.out.println("wykonuje!");
15         }
16     });
17
18 }
```



```
D:\Studia\Java_projekty\RSI\lab7\src>java jade.Boot -gui xx:klasa_2
kwi 30, 2018 8:40:45 AM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
kwi 30, 2018 8:40:45 AM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.1.110:1099
-----
kwi 30, 2018 8:40:45 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
kwi 30, 2018 8:40:45 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
kwi 30, 2018 8:40:45 AM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
kwi 30, 2018 8:40:45 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
kwi 30, 2018 8:40:45 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
kwi 30, 2018 8:40:46 AM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
kwi 30, 2018 8:40:46 AM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://DESKTOP-11C28S2.local:7778/acc
startuje!
kwi 30, 2018 8:40:46 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@192.168.1.110 is ready.
-----
wykonuje!
zaraz sie usune!
D:\Studia\Java_projekty\RSI\lab7\src>
```

## 3. Utworzyłam klasę agenta o nazwie klasa\_3 na podstawie kodu klasa\_1.

Do agenta dodałam zachowanie polegające na wielokrotnym (cyklicznym) wykonaniu operacji wypisania na ekranie słowa „wykonuję”.

```
7 protected void setup() {
8     System.out.println("startuje!");
9
10     addBehaviour(new CyclicBehaviour() {
11         @Override
12         public void action() {
13             System.out.println("wykonuje!");
14         }
15     });
16
17 }
```

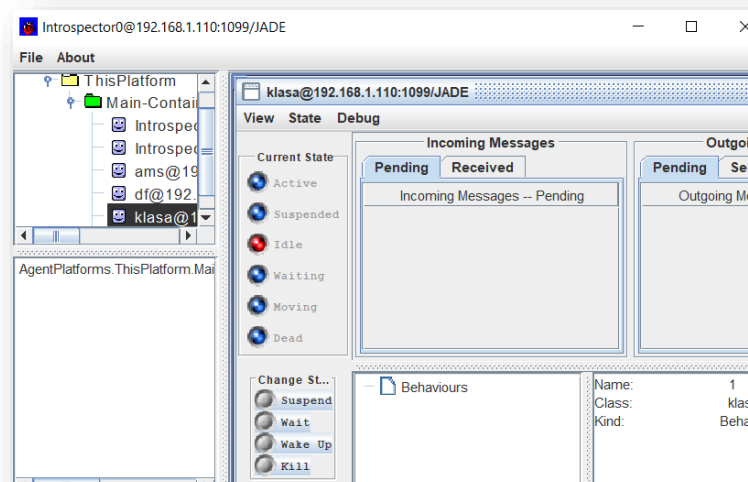
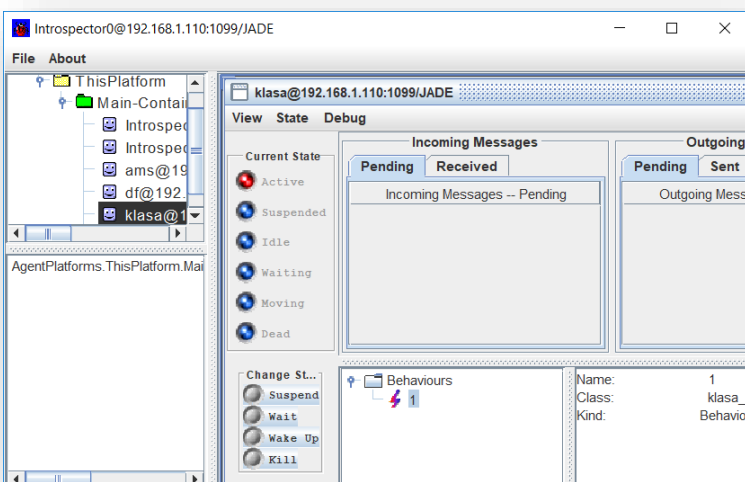


5. Utworzyłam agenta o nazwie klasa\_5 na podstawie kodu klasa\_1. Do agenta dodałam zachowanie, które polega na pobieraniu z klawiatury liczby całkowitej. Jeśli użytkownik poda liczbę ujemną, to zachowanie zostaje usunięte.

```
11 protected void setup() {
12     System.out.println("startuje!");
13
14     Scanner scanner = new Scanner(System.in);
15     addBehaviour( new Behaviour()
16     {
17         public void action() {
18             System.out.print("Podaj liczbę całkowita: ");
19             int number = scanner.nextInt();
20             if(number>0)
21             {
22                 System.out.println("liczba dodatnia!");
23             }
24             if(number<0)
25             {
26                 System.out.println("liczba ujemna!-usuwaam Agenta");
27                 removeBehaviour( B: this);
28             }
29         }
30
31         @Override
32         public boolean done() {
33             return false;
34         }
35     });
36 }
```

```
startuje!
kwi 30, 2018 9:57:36 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@192.168.1.110 is ready.
-----
Podaj liczbę całkowita: 3
liczba dodatnia!
Podaj liczbę całkowita: -3
liczba ujemna!-usuwaam Agenta
zaraz sie usune!

D:\Studia\Java_projekty\RSI\lab7\src>
```



6. Utworzyłam klasę agenta o nazwie klasa\_6 na podstawie kodu klasa\_5. Zmodyfikowałam kod tak, aby zawsze zachowanie na początku wypisywało „zachowanie startuje”, a na samym końcu wypisywało „zachowanie zakończone”.

```
9 protected void setup() {
10
11     Scanner scanner = new Scanner(System.in);
12     System.out.println("startuje!");
13
14     addBehaviour( new Behaviour()
15     {
16         public void action() {
17             System.out.println( "zachowanie startuje" );
18         }
19         @Override
20         public boolean done() {
21             return true; }
22     });
23
24     addBehaviour( new Behaviour()
25     {
26         public void action() {
27             System.out.print("Podaj liczbę całkowita: ");
28             int number = scanner.nextInt();
29             if(number>0) {
30                 System.out.println("liczba dodatnia!");
31             }
32             if(number<0) {
33                 System.out.println("liczba ujemna!-usuwaam zachowanie Agenta");
34                 removeBehaviour( B: this);
35                 System.out.println("zachowanie zakończone!");
36             }
37         }
38     });
39 }
```

```
startuje!
zachowanie startuje
Podaj liczbę całkowita: 3
liczba dodatnia!
Podaj liczbę całkowita: -3
liczba ujemna!-usuwaam zachowanie Agenta
zachowanie zakończone!
```

**7. Utwórz klasę agenta o nazwie klasa 7 na podstawie kodu klasa 4.** Do istniejącego zachowania

„generycznego” dodaj dwa kolejne:

- Pierwsze ma być dodawane na poziomie metody setup() agenta i ma polegać na jednokrotnym wypisaniu „pierwsze”
- Drugie ma być dodane z poziomu zachowania „generycznego” – dokładnie ma być dodane w pierwszym kroku i ma polegać na jednokrotnym wypisaniu „drugie”.

```
protected void setup() {
    System.out.println("startuje!");
    OneShotBehaviour oneShotBehaviour1=new OneShotBehaviour() {
        @Override
        public void action() {
            System.out.println("pierwsze");
        }
    };
    addBehaviour(oneShotBehaviour1);
}
```

```
18 addBehaviour(new Behaviour() {
19     @Override
20     public void action() {
21
22         switch(step){
23             case 0:
24                 OneShotBehaviour oneShotBehaviour2=new OneShotBehaviour() {
25                     @Override
26                     public void action() {
27                         System.out.println("drugie");
28                     }
29                 };
30                 addBehaviour(oneShotBehaviour2);
31
32                 System.out.println("pierwszy krok.");
33                 step++;
34                 break;
35             case 1:
36                 System.out.println("drugi krok.");
37                 step++;
38                 break;
39             case 2:
40                 System.out.println("trzeci krok.");
41                 step++;
42                 break;
43         }
44     }
45 }
```

```
startuje!
pierwsze
kwi 30, 2018 10:58:57 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@192.168.1.110 is ready.
-----
pierwszy krok.
drugi krok.
drugie
trzeci krok.
zaraz sie usune!
D:\Studia\Java_projekty\RSI\lab7\src>
```

**8. Utworzyłam klasę agenta o nazwie klasa 8 na podstawie kodu klasa 1.** Do agenta dodaj zachowania, które spowodują:

- Wypisanie „mały tick” co 2 sekundy,
- Wypisaniu „duży tick” co 5 sekund,
- Po 50 sekundach usunięcie zachowania z punktu b,
- Po 100 sekundach usunięcie całego agenta.



Działanie **Introspektora** na końcu wypisywania :

