

Urszula Ślusarz

Inżynieria Obliczeniowa, WIMiP

Akademia Górniczo-Hutnicza w Krakowie

Platformy agentowe w języku Java

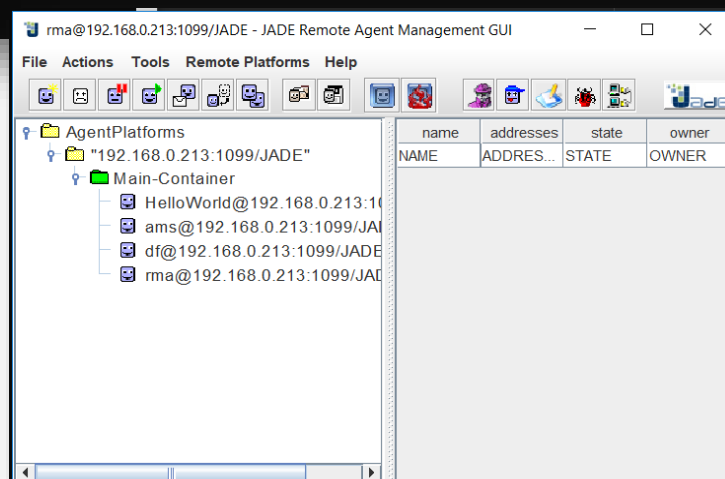
Ćwiczenie 6-Podstawy JADE

1. Dokonałam uruchomienia platformy JADE.
2. Skopiowałam przykład hello (**klasę HelloWorldAgent**). A następnie uruchomiłam agenta z linii komend i przy pomocy Gui.
3. Dokonałam modyfikacji pliku **HelloWorldAgent.java** tak, aby agent nie usuwał się po wypisaniu tekstu na ekranie.

```
protected void setup() {  
    System.out.println("Hello World! My name is "+getLocalName());  
  
    // Make this agent terminate  
    //doDelete();  
}
```

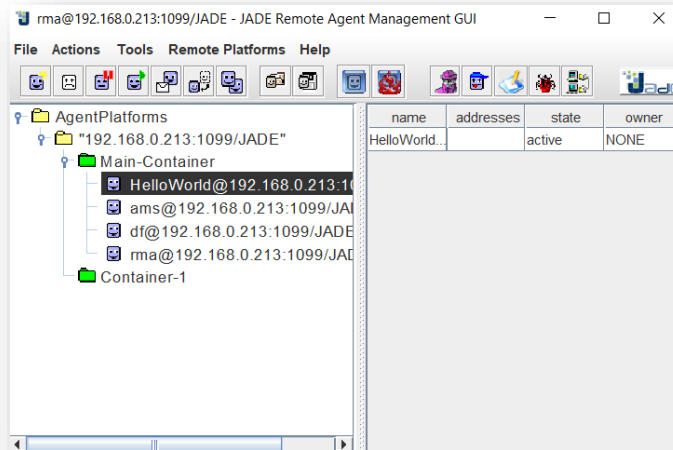
Aby agent nie usuwał się po wypisaniu tekstu na ekranie zakomentowałam metodę „doDelete()”.

```
Wiersz polecenia - java jade.Boot -gui HelloWorld:HelloWorldAgent  
D:\Studia\Java_projekty\RSI\hello>javac HelloWorldAgent.java  
D:\Studia\Java_projekty\RSI\hello>java jade.Boot -gui HelloWorld:HelloWorldAgent  
kwi 28, 2018 11:14:33 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 28, 2018 11:14:33 AM jade.imtp.leap.LEAPIMTPManager initialize  
INFO: Listening for intra-platform commands on address:  
- jicp://192.168.0.213:1099  
  
kwi 28, 2018 11:14:33 AM jade.core.BaseService init  
INFO: Service jade.core.management.AgentManagement initialized  
kwi 28, 2018 11:14:33 AM jade.core.BaseService init  
INFO: Service jade.core.messaging.Messaging initialized  
kwi 28, 2018 11:14:33 AM jade.core.BaseService init  
INFO: Service jade.core.resource.ResourceManagement initialized  
kwi 28, 2018 11:14:33 AM jade.core.BaseService init  
INFO: Service jade.core.mobility.AgentMobility initialized  
kwi 28, 2018 11:14:33 AM jade.core.BaseService init  
INFO: Service jade.core.event.Notification initialized  
kwi 28, 2018 11:14:33 AM jade.mtp.http.HTTPServer <init>  
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser  
kwi 28, 2018 11:14:33 AM jade.core.messaging.MessagingService boot  
INFO: MTP addresses:  
http://192.168.0.213:7778/acc  
kwi 28, 2018 11:14:33 AM jade.core.AgentContainerImpl joinPlatform  
INFO: -----  
Agent container Main-Container@192.168.0.213 is ready.  
-----  
Hello World! My name is HelloWorld
```

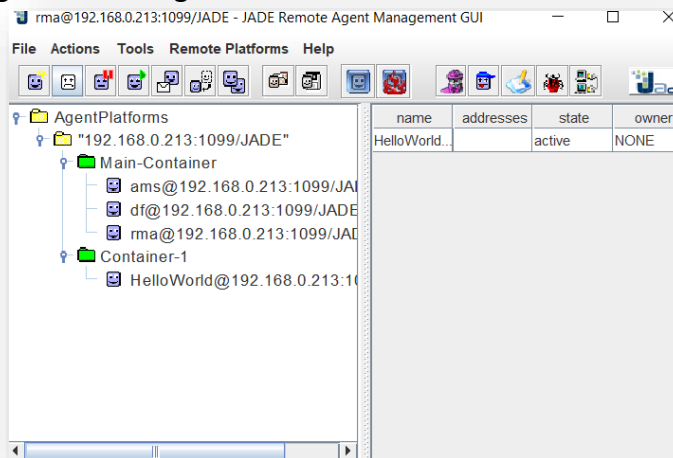


4. Do uruchomionej platformy:

a). dodałam jeszcze jeden kontener

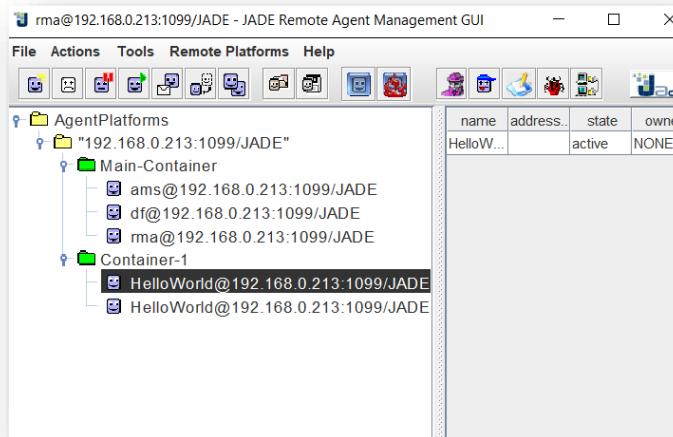


b).Przeniosłam agenta do innego kontenera



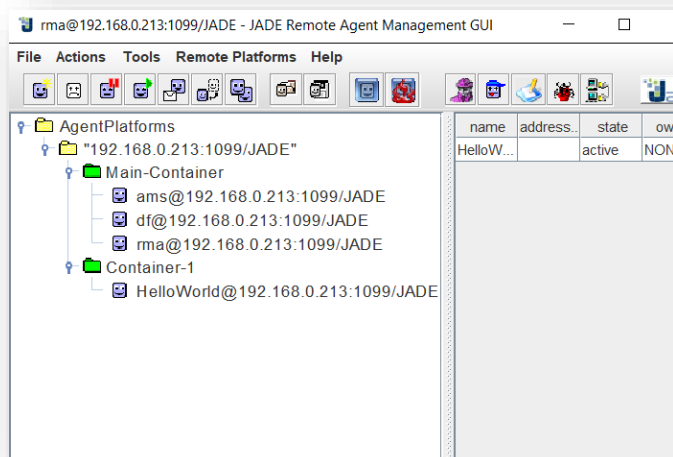
Migrate Agent

c. Sklonowałam agenta



Clone Agent

d. Usunęłam wcześniej sklonowanego agenta



Kill

5. Zmodyfikowałam przykład **TimeAgent.java** tak, aby agent usuwał się dopiero po 2 minutach.

```
public class TimeAgent extends Agent {

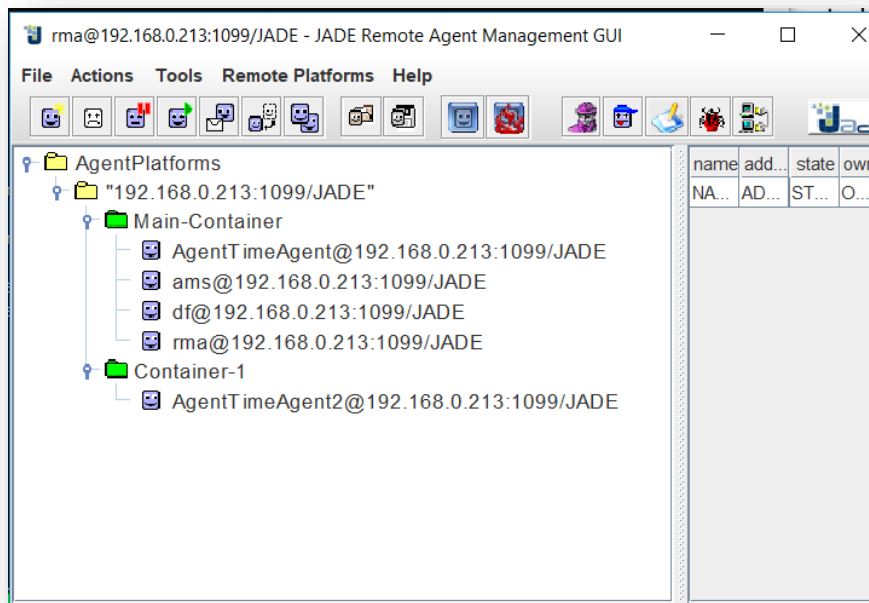
    protected void setup() {
        System.out.println("Agent "+getLocalName()+" started.");

        // Add the TickerBehaviour (period 1 sec)
        addBehaviour(new TickerBehaviour( @: this, period: 1000) {
            protected void onTick() { System.out.println("Agent "+myAgent.getLocalName()+" : tick="+getTickCount()); }
        });

        // Add the WakerBehaviour (wakeup-time 10 secs)
        addBehaviour(new WakerBehaviour( @: this, timeout: 120000) {
            protected void handleElapsedTimeout() {
                System.out.println("Agent "+myAgent.getLocalName()+" : It's wakeup-time. Bye...");
                myAgent.doDelete();
            }
        });
    }
}
```

6. Przyłączyłam kontener na moim komputerze do zdalnego hosta.

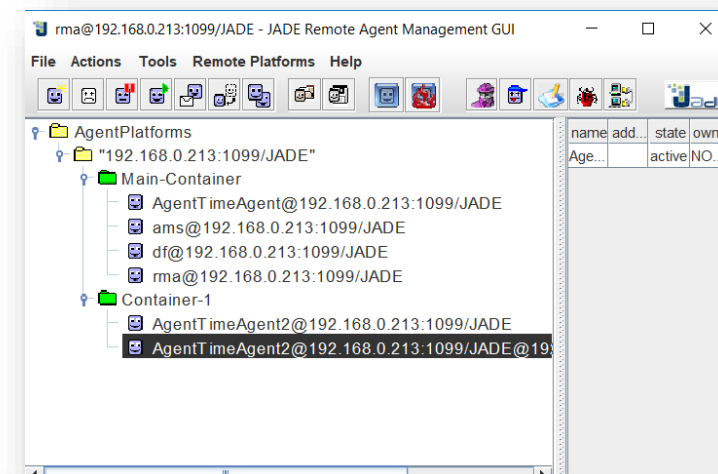
Przyłączyłam raz jeszcze raz z dodaniem agenta klasy **TimeAgent**, w wyniku tego otrzymujemy dwóch agentów, którzy odliczają czas.



```
Agent AgentTimeAgent: tick=7
Agent AgentTimeAgent: tick=8
Agent AgentTimeAgent: tick=9
Agent AgentTimeAgent: tick=10
Agent AgentTimeAgent: tick=11
Agent AgentTimeAgent: tick=12
Agent AgentTimeAgent: tick=13
Agent AgentTimeAgent: tick=14
Agent AgentTimeAgent: tick=15
Agent AgentTimeAgent: tick=16
Agent AgentTimeAgent: tick=17
Agent AgentTimeAgent: tick=18
Agent AgentTimeAgent: tick=19
Agent AgentTimeAgent: tick=20
Agent AgentTimeAgent: tick=21
Agent AgentTimeAgent: tick=22
Agent AgentTimeAgent: tick=23
Agent AgentTimeAgent: tick=24
Agent AgentTimeAgent: tick=25
Agent AgentTimeAgent: tick=26
Agent AgentTimeAgent: tick=27
Agent AgentTimeAgent: tick=28
Agent AgentTimeAgent: tick=29
Agent AgentTimeAgent: It's wakeup-time. Bye...
```

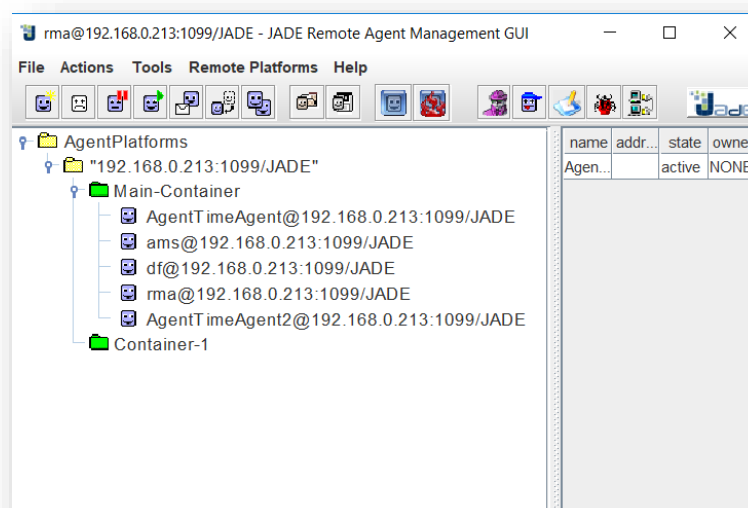
```
Agent AgentTimeAgent2: tick=12
Agent AgentTimeAgent2: tick=13
Agent AgentTimeAgent2: tick=14
Agent AgentTimeAgent2: tick=15
Agent AgentTimeAgent2: tick=16
Agent AgentTimeAgent2: tick=17
Agent AgentTimeAgent2: tick=18
Agent AgentTimeAgent2: tick=19
Agent AgentTimeAgent2: tick=20
Agent AgentTimeAgent2: tick=21
Agent AgentTimeAgent2: tick=22
Agent AgentTimeAgent2: tick=23
Agent AgentTimeAgent2: tick=24
Agent AgentTimeAgent2: tick=25
Agent AgentTimeAgent2: tick=26
Agent AgentTimeAgent2: tick=27
Agent AgentTimeAgent2: tick=28
Agent AgentTimeAgent2: tick=29
Agent AgentTimeAgent2: It's wakeup-time. Bye...
```

Sklonowałam agenta, kopiowanie stanu agenta powoduje, że oba agenty (pierwotny i sklonowany) odliczają czas.



```
C:\> Wiersz polecenia - java jade.Boot -gui -container AgentTimeAgent2:TimeAgent
Agent AgentTimeAgent2: tick=80
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=81
Agent AgentTimeAgent2: tick=81
Agent AgentTimeAgent2: tick=82
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=82
Agent AgentTimeAgent2: tick=83
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=83
Agent AgentTimeAgent2: tick=84
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=84
Agent AgentTimeAgent2: tick=85
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=85
Agent AgentTimeAgent2: tick=86
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=86
Agent AgentTimeAgent2: tick=87
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=87
Agent AgentTimeAgent2: tick=88
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=88
Agent AgentTimeAgent2: tick=89
Agent AgentTimeAgent2@192.168.0.213:1099/JADE: tick=89
```

Kolejnym krokiem było dokonanie migracji agenta z „*Container-1 do Main-Container AgentTimeAgent2*”.



```

C:\> Wiersz polecenia - java jade.Boot -gui -container
Agent AgentTimeAgent2: tick=1
Agent AgentTimeAgent2: tick=2
Agent AgentTimeAgent2: tick=3
Agent AgentTimeAgent2: tick=4
Agent AgentTimeAgent2: tick=5
Agent AgentTimeAgent2: tick=6
Agent AgentTimeAgent2: tick=7
Agent AgentTimeAgent2: tick=8
Agent AgentTimeAgent2: tick=9
Agent AgentTimeAgent2: tick=10
Agent AgentTimeAgent2: tick=11
Agent AgentTimeAgent2: tick=12
Agent AgentTimeAgent2: tick=13
Agent AgentTimeAgent2: tick=14
Agent AgentTimeAgent2: tick=15
Agent AgentTimeAgent2: tick=16
Agent AgentTimeAgent2: tick=17
Agent AgentTimeAgent2: tick=18
Agent AgentTimeAgent2: tick=19

```

```

C:\> Wiersz polecenia - java jade.Boot -gui AgentTim
Agent AgentTimeAgent: tick=19
Agent AgentTimeAgent: tick=20
Agent AgentTimeAgent: tick=21
Agent AgentTimeAgent: tick=22
Agent AgentTimeAgent: tick=23
Agent AgentTimeAgent2: tick=20
Agent AgentTimeAgent: tick=24
Agent AgentTimeAgent2: tick=21
Agent AgentTimeAgent: tick=25
Agent AgentTimeAgent2: tick=22
Agent AgentTimeAgent: tick=26
Agent AgentTimeAgent2: tick=23
Agent AgentTimeAgent: tick=27
Agent AgentTimeAgent2: tick=24
Agent AgentTimeAgent: tick=28
Agent AgentTimeAgent2: tick=25
Agent AgentTimeAgent: tick=29
Agent AgentTimeAgent2: tick=26
Agent AgentTimeAgent: tick=30
Agent AgentTimeAgent2: tick=27
Agent AgentTimeAgent: tick=31
Agent AgentTimeAgent2: tick=28
Agent AgentTimeAgent: tick=32
Agent AgentTimeAgent2: tick=29
Agent AgentTimeAgent: tick=33
Agent AgentTimeAgent2: tick=30
Agent AgentTimeAgent: tick=34
Agent AgentTimeAgent2: tick=31
Agent AgentTimeAgent: tick=35

```

AgentTimeAgent2 został przeniesiony do Main-Container w 20s, i od tego czasu kontynuuje odliczanie czasu w nowym dla niego kontenerze.

7. Prześledziłam sytuacje wyjątkowe, gdy np. próbowałam stworzyć agenta w kontenerze, który nie istnieje, pojawiał się błąd.

8. Dokonałam modyfikacji pliku **HelloWorldAgent.java** tak, aby agent wypisywał dotychczasowy komunikat tyle razy, ile zostanie podane w parametrach agenta.

```

public class HelloWorldAgent extends Agent {

    protected void setup() {

        Object[] tab = getArguments();
        int liczba_komunikatu = Integer.parseInt(tab[0].toString());
        System.out.println("komunika numer= "+liczba_komunikatu);
        for(int i=0;i<liczba_komunikatu;i++)
            System.out.println("Hello World! My name is "+getLocalName()+"numer"+i);

        // Make this agent terminate
        //doDelete();
    }
}

```

```

D:\Studia\Java_projekty\RSI\hello>java jade.Boot -gui HelloWorld:HelloWorldAgent(2)
kwi 28, 2018 3:16:32 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 28, 2018 3:16:32 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.0.213:1099

kwi 28, 2018 3:16:32 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
kwi 28, 2018 3:16:32 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
kwi 28, 2018 3:16:32 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
kwi 28, 2018 3:16:32 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
kwi 28, 2018 3:16:32 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
kwi 28, 2018 3:16:32 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
kwi 28, 2018 3:16:32 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://192.168.0.213:7778/acc
komunika numer= 2kwi 28, 2018 3:16:32 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@192.168.0.213 is ready.
-----
Hello World! My name is HelloWorldnumer0
Hello World! My name is HelloWorldnumer1

```

9. Zmodyfikowałam swój dotychczasowy program tak, aby wszystko co było w metodzie **main()** zostało wywołane w klasie agenta.

```
import jade.core.Agent;

/**
 * This example shows a minimal agent that just prints "Hallo World!"
 * and then terminates.
 * @author Giovanni Caire - TILAB
 */
public class HelloWorldAgent extends Agent {

    protected void setup() {

        Object[] tab = getArguments();
        int liczba_komunikatu = Integer.parseInt(tab[0].toString());
        System.out.println("komunika numer= "+liczba_komunikatu);
        for(int i=0;i<liczba_komunikatu;i++)
            System.out.println("Hello World! My name is "+getLocalName()+"numer"+i);

        // Make this agent terminate
        //doDelete();
    }

    public static void main(String[] args){

    }

}
```