

Klaudia Dziewit  
Inżynieria Obliczeniowa  
III rok gr. 1

## Rozproszona sztuczna inteligencja - ćwiczenia 6

### Wykonanie

Wykonanie ćwiczenia rozpoczęłam od zapoznania się z plikami “programmersguide.pdf” oraz “administratorguide.pdf”. Stworzyłam pliki “compiljade.bat” oraz “runjade.bat”. Umożliwiają one kolejno kompilację pliku oraz umożliwiają uruchomienie platformy jade z Gui. Można również uruchomić platformę Jade za pomocą konsoli.

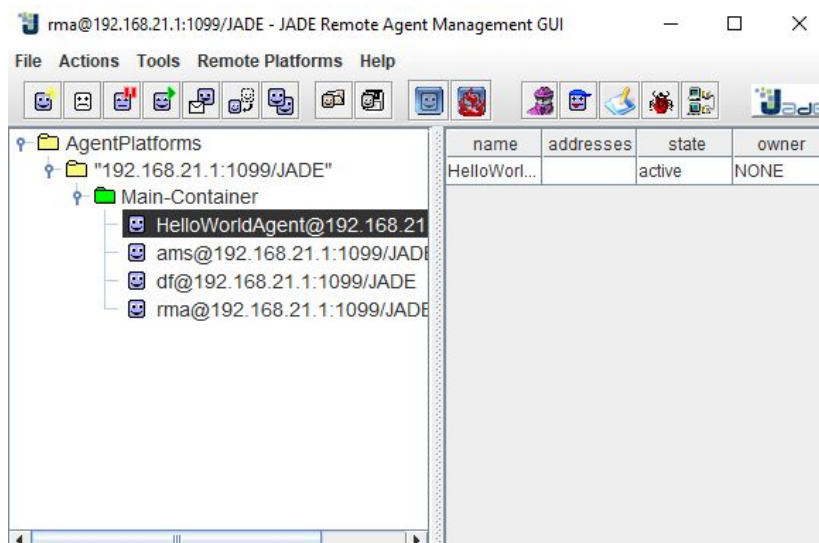
Następnie skopiowałam przykład hello z folderu Jade. Skompilowałam klasę HelloWorldAgent.

```
Agent container Main-Container@192.168.21.1 is ready.
-----
Hello World! My name is HelloWorldAgent
```

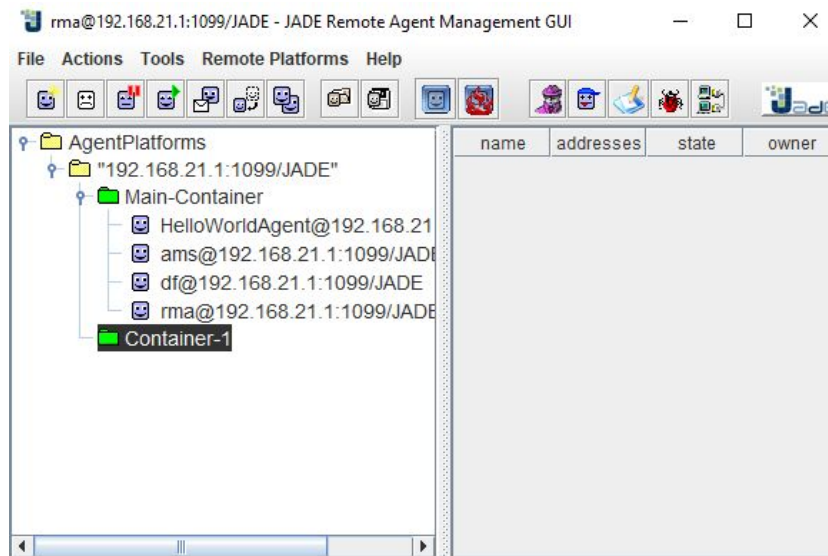
Następnie dokonałam modyfikacji pliku HelloWorldAgent.java tak, aby agent nie usuwał się po wypisaniu tekstu na ekranie i skompilowałam kod. Aby to się udało należało usunąć polecenie, odpowiadające za jego usuwanie czyli: “doDelete();”.

```
public class Agent1 extends Agent{
    protected void setup() {
        System.out.println("Hello Wold! My name is " + getLocalName());
    }
}
```

Obecność agenta w kontenerze potwierdza działanie zmodyfikowanego kodu:

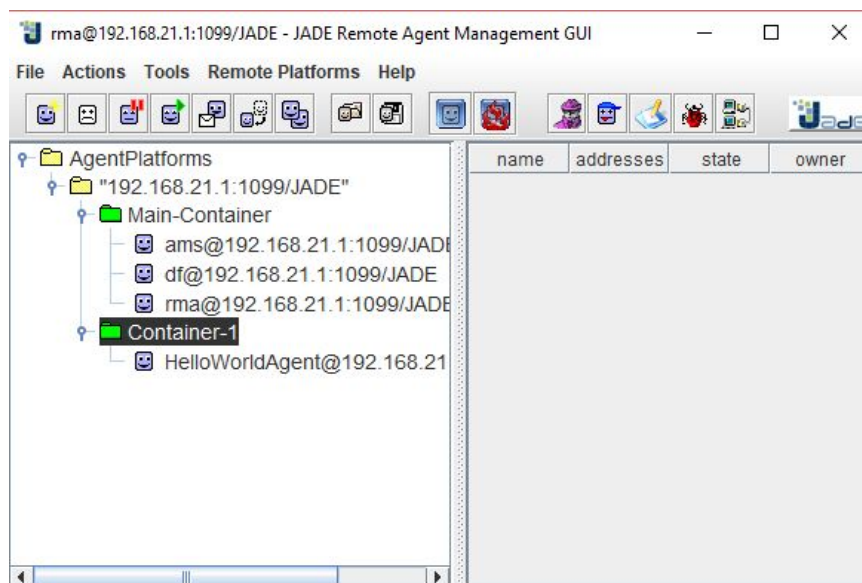


Do uruchomionej platformy dodałam jeszcze jeden kontener (oprócz Main-Container) za pomocą polecenia *-container* z konsoli.

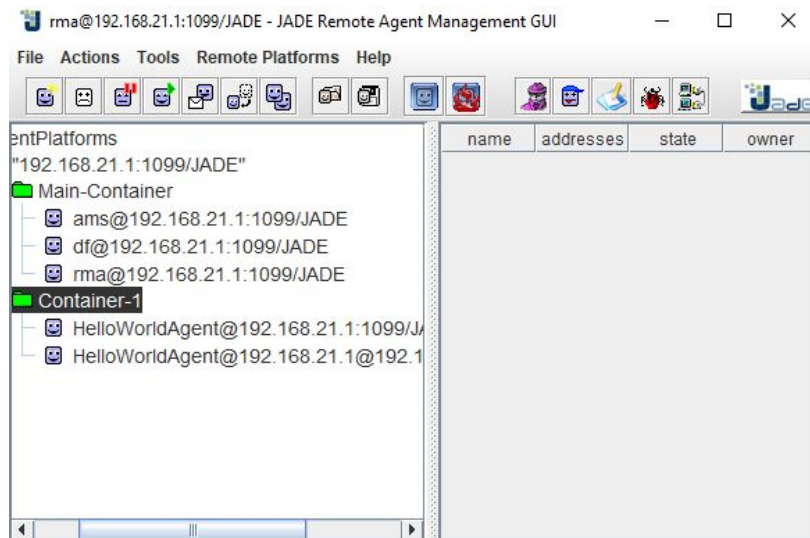


Następnie:

- uruchomiłam agenta klasy HelloWorldAgent w nowym kontenerze i przeniosłam agenta do innego kontenera (opcja *migrate agent*)



- sklonowałam agenta przy pomocy *clone agent*



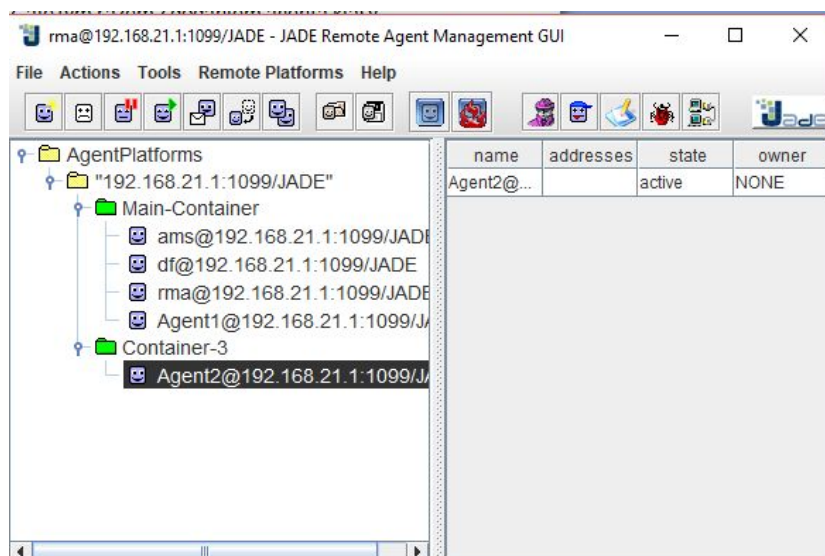
- usunęłam agenta przy użyciu opcji *Kill*.

Kolejnym krokiem była modyfikacja przykładu TimeAgent.java tak, aby agent usuwał się dopiero po 2 minutach. Skompilowałam kod. Aby tego dokonać, należało zmodyfikować warunek metody dodającej zachowanie agenta (*WakerBehaviour*) na 120s opóźnienia.

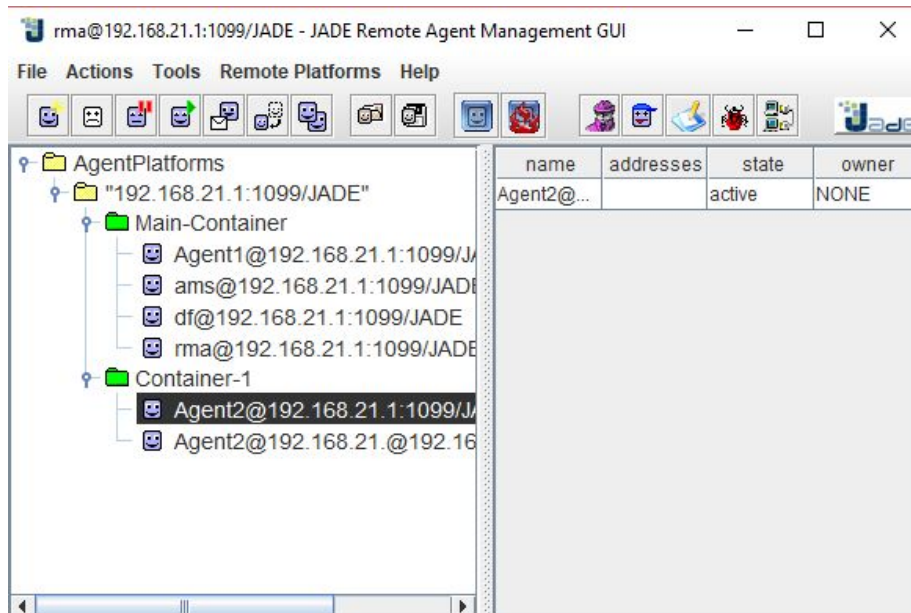
```
addBehaviour(new WakerBehaviour(this, 120000)
```

Następnie przyłączyłam kontener na moim komputerze do zdalnego hosta, korzystając z opcji *-container -name C -host<adres IP hosta>*.

- poprosiłam o usunięcie mojego kontenera (*Kill*)
- przyłączyłam raz jeszcze raz z dodaniem agenta klasy TimeAgent, dzięki poleceniu *-container Agent2: TimeAgent*

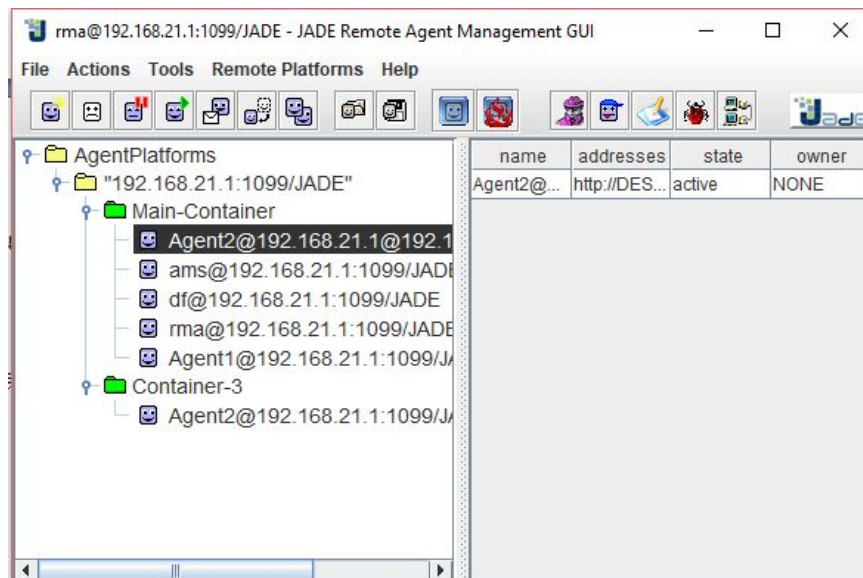


W tym momencie osobno wykonuje się tick Agent1 oraz Agent2, przy czym Agent1 kontynuuje swoje działanie a nie zaczyna od początku.  
Następnie poprosiłam o sklonowanie agenta, co spowodowało, że od tego momentu Agent2 wykonuje swoje działanie podwójnie.



```
Agent Agent2: tick=41
Agent Agent2@192.168.21.: tick=41
Agent Agent2: tick=42
Agent Agent2@192.168.21.: tick=42
Agent Agent2: tick=43
Agent Agent2@192.168.21.: tick=43
Agent Agent2: tick=44
Agent Agent2@192.168.21.: tick=44
Agent Agent2: tick=45
Agent Agent2@192.168.21.: tick=45
Agent Agent2: tick=46
Agent Agent2@192.168.21.: tick=46
Agent Agent2: tick=47
Agent Agent2@192.168.21.: tick=47
Agent Agent2: tick=48
Agent Agent2@192.168.21.: tick=48
Agent Agent2: tick=49
Agent Agent2@192.168.21.: tick=49
Agent Agent2: tick=50
Agent Agent2@192.168.21.: tick=50
Agent Agent2: tick=51
Agent Agent2@192.168.21.: tick=51
```

Następnie dokonałam migracji agenta do Main-Container:



Od tego momentu Agent1 wykonywał swoje działanie naprzemiennie z Agentem2.

```

Agent Agent1: tick=76
Agent Agent1: tick=77
Agent Agent1: tick=78
Agent Agent1: tick=79
Agent Agent2@192.168.21.1: tick=56
Agent Agent1: tick=80
Agent Agent2@192.168.21.1: tick=57
Agent Agent1: tick=81
Agent Agent2@192.168.21.1: tick=58
Agent Agent1: tick=82
Agent Agent2@192.168.21.1: tick=59
Agent Agent1: tick=83
Agent Agent2@192.168.21.1: tick=60
Agent Agent1: tick=84
Agent Agent2@192.168.21.1: tick=61
Agent Agent1: tick=85
Agent Agent2@192.168.21.1: tick=62
Agent Agent1: tick=86
Agent Agent2@192.168.21.1: tick=63
Agent Agent1: tick=87
Agent Agent2@192.168.21.1: tick=64
Agent Agent1: tick=88
Agent Agent2@192.168.21.1: tick=65

```

Jeżeli brakuje klasy tworzonego agenta to kompilator zwróci błąd. Tak samo stanie się w sytuacji, gdy będziemy chcieli stworzyć agenta w kontenerze, który nie istnieje. Ponadto, nie jest możliwe sklonowanie Agentu do kontenera, który nie istnieje.

Kolejnym krokiem było dokonanie modyfikacji pliku HelloWorldAgent.java tak, aby agent wypisywał dotychczasowy komunikat tyle razy, ile zostanie podane w parametrach agenta.



```

public class Agent3 extends Agent{
    int argsAmount;

    protected void setup() {
        System.out.println("Agent " + getLocalName() + " has started");
        Object[] arguments = getArguments();
        argsAmount = Integer.parseInt(arguments[0].toString());
        System.out.println("Argument " + argsAmount);
        for(int i=0; i<argsAmount; i++) {
            System.out.println("Agent " + getLocalName() + " Argument no: " + i);
        }
    }
}

```

Po zastosowaniu polecenia `-gui HelloWorldAgent:HelloWorldAgent(5)` otrzymujemy następujący rezultat:

```

-----
Agent HelloWorldAgent started.
Argument = 5
Hello World! My name is HelloWorldAgent. argument: 0
Hello World! My name is HelloWorldAgent. argument: 1
Hello World! My name is HelloWorldAgent. argument: 2
Hello World! My name is HelloWorldAgent. argument: 3
Hello World! My name is HelloWorldAgent. argument: 4

```

Ostatnim krokiem była modyfikacja dotychczasowego programu tak, aby wszystko co było w metodzie `main()` zostało wywołane w klasie agenta.

```

-----
Hello World.My name is HelloWorldAgent
Print from Main

```