

Patryk Kurzeja

Inżynieria Obliczeniowa

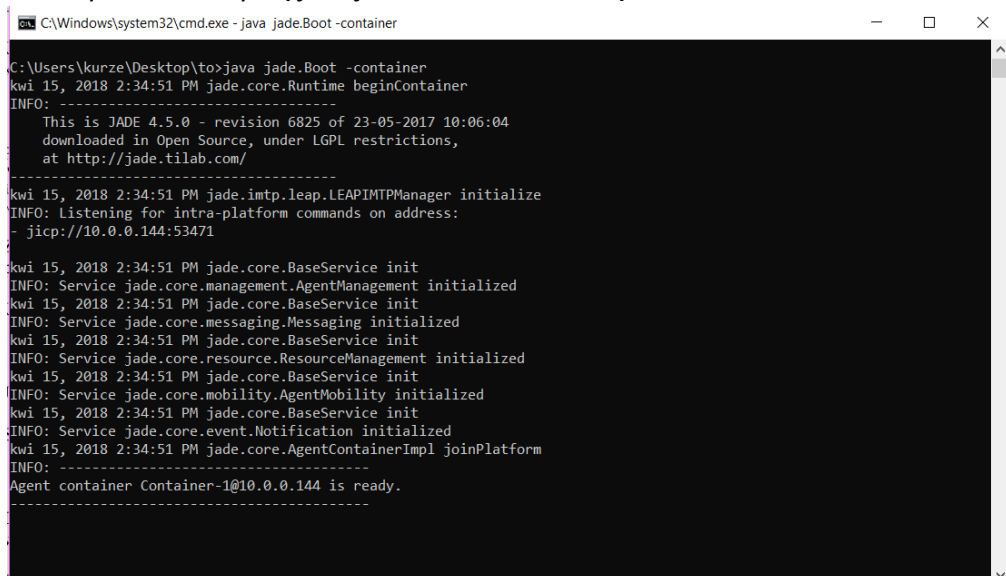
Nr albumu : 286112

Rozproszona sztuczna inteligencja - Platformy agentowe w języku java

## Ćwiczenia 6 – Podstawy JADE :

### Wykonanie :

1. Utworzyłem pliki compiljade.bat oraz runjade.bat pozwalające na kompilowanie kodu agentów oraz uruchomienie platformy.
2. Uruchomiłem platformę JADE :
  - a. W trybie tekstowym ( **java jade.Boot -container** ) :

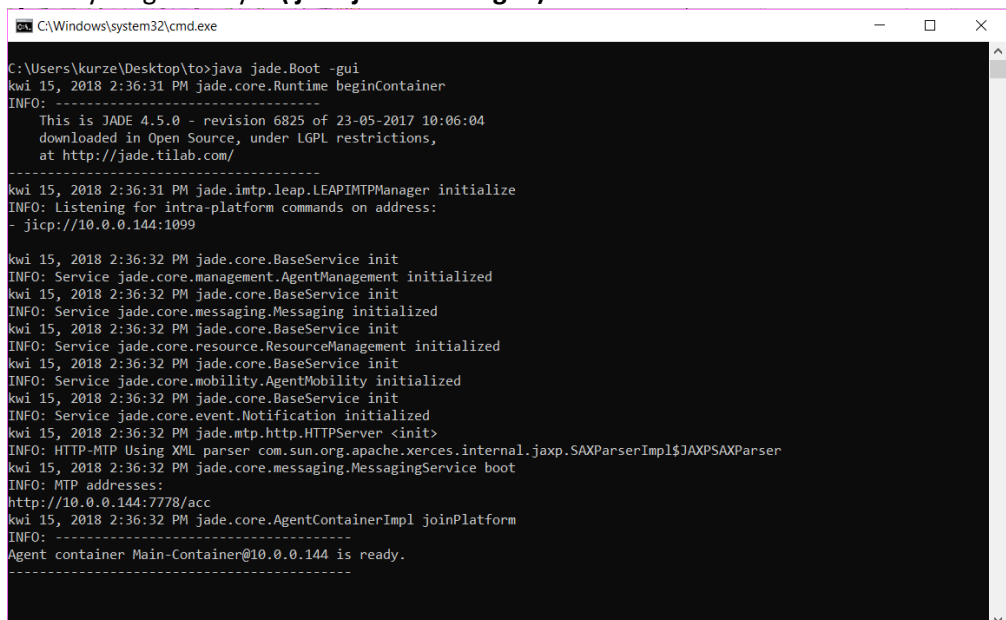


```
C:\Windows\system32\cmd.exe - java jade.Boot -container

C:\Users\kurze\Desktop\to>java jade.Boot -container
kwi 15, 2018 2:34:51 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 15, 2018 2:34:51 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://10.0.0.144:53471

kwi 15, 2018 2:34:51 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
kwi 15, 2018 2:34:51 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
kwi 15, 2018 2:34:51 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
kwi 15, 2018 2:34:51 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
kwi 15, 2018 2:34:51 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
kwi 15, 2018 2:34:51 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Container-1@10.0.0.144 is ready.
-----
```

- b. W trybie graficznym ( **java jade.Boot -gui** ) :

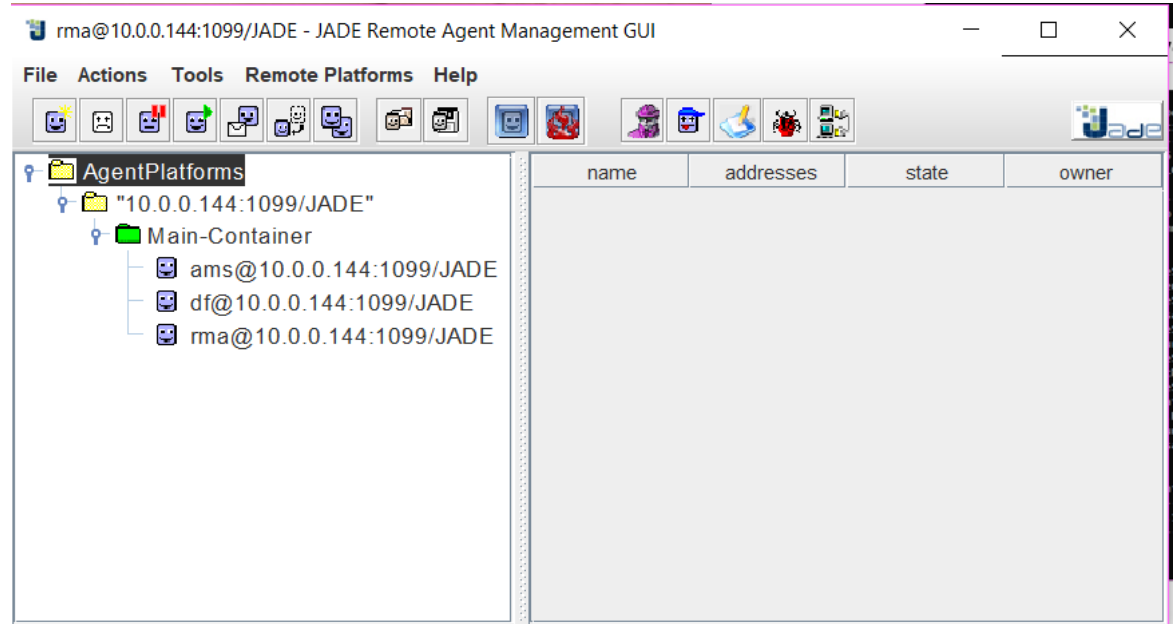


```
C:\Windows\system32\cmd.exe

C:\Users\kurze\Desktop\to>java jade.Boot -gui
kwi 15, 2018 2:36:31 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 15, 2018 2:36:31 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://10.0.0.144:1099

kwi 15, 2018 2:36:32 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
kwi 15, 2018 2:36:32 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
kwi 15, 2018 2:36:32 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
kwi 15, 2018 2:36:32 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
kwi 15, 2018 2:36:32 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
kwi 15, 2018 2:36:32 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
kwi 15, 2018 2:36:32 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://10.0.0.144:7778/acc
kwi 15, 2018 2:36:32 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@10.0.0.144 is ready.
-----
```

### GUI :



Aby uruchomić platformę JADE wystarczy w CMD wpisać **java jade.Boot -container** lub **java jade.Boot -gui** w zależności czy chcemy tryb graficzny czy nie.

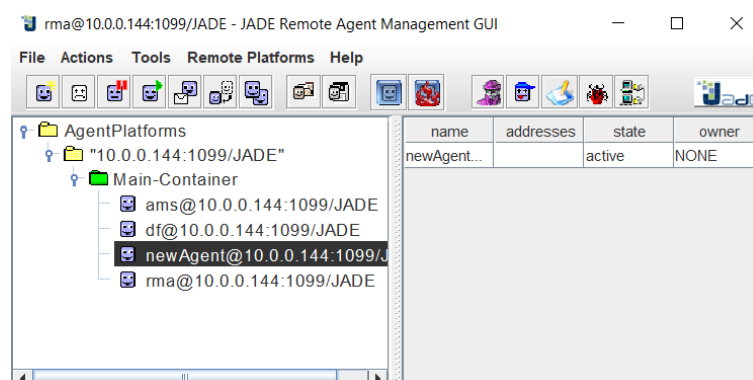
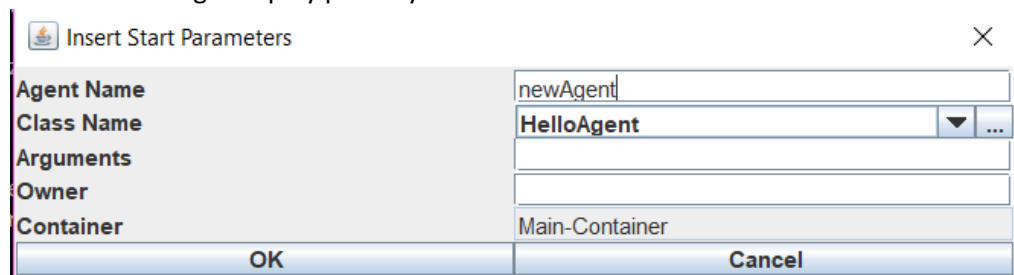
3. Skopiowałem przykład hello oraz skompilowałem klasę HelloWorldAgent a następnie :

a. Uruchomiłem agenta z linii komend ( **java jade.Boot -gui x:HelloAgent** ) :

Agent wyświetlił :

```
Hello World.  
My name is NewAgent
```

b. Uruchomiłem agenta przy pomocy GUI :

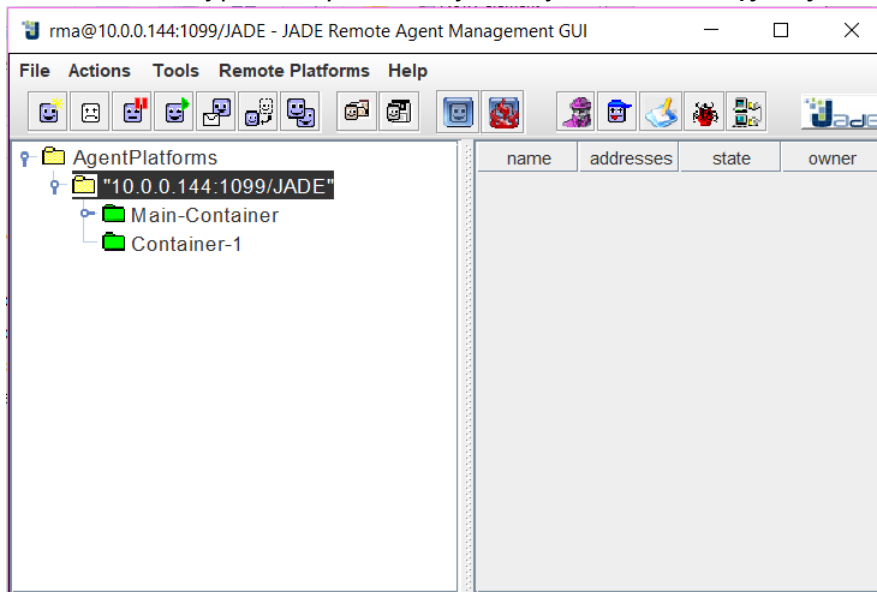


4. Dokonałem modyfikacji pliku **HelloWorldAgent.java**, tak aby agent nie usuwał się po wypisaniu tekstu na ekranie oraz skompilowałem kod :

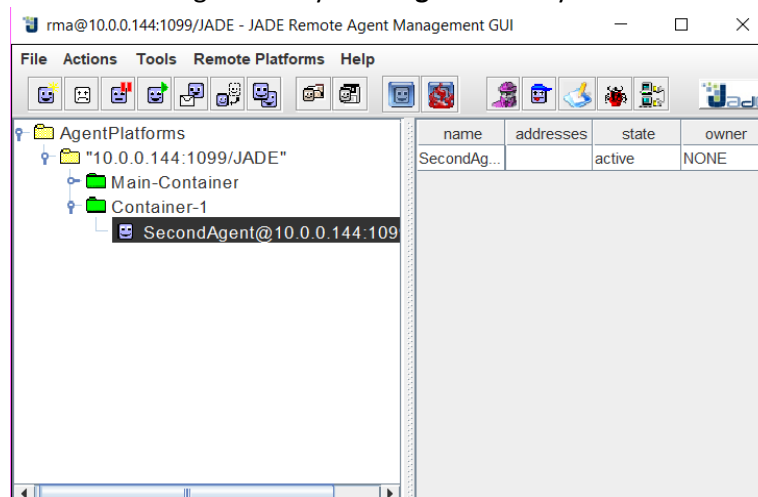
```
1  import jade.core.Agent;
2
3  public class HelloAgent extends Agent
4  {
5      protected void setup()
6      {
7          System.out.println("Hello World. ");
8          System.out.println("My name is " + getLocalName());
9
10         // Make this agent terminate
11         doDelete();
12     }
13 }
14
```

Należy usunąć zaznaczoną linię kodu.

5. Do uruchomionej platformy dodałem jeszcze jeden kontener ( **java jade.Boot -container** ) :



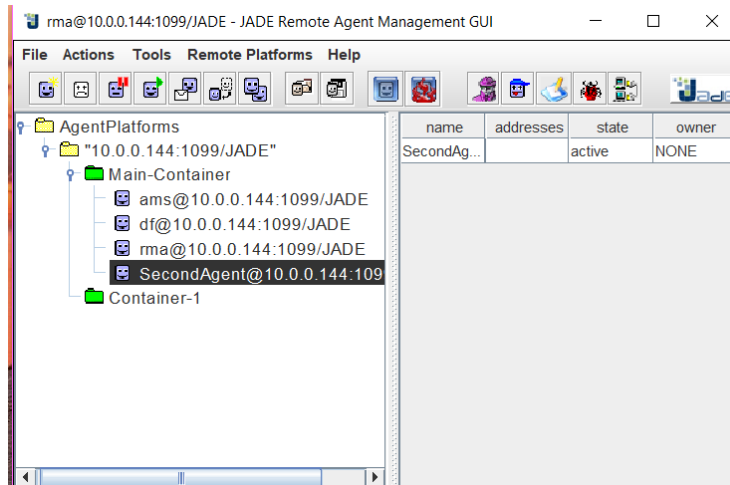
- a. Uruchomiłem agenta klasy **HelloAgent** w nowym kontenerze :



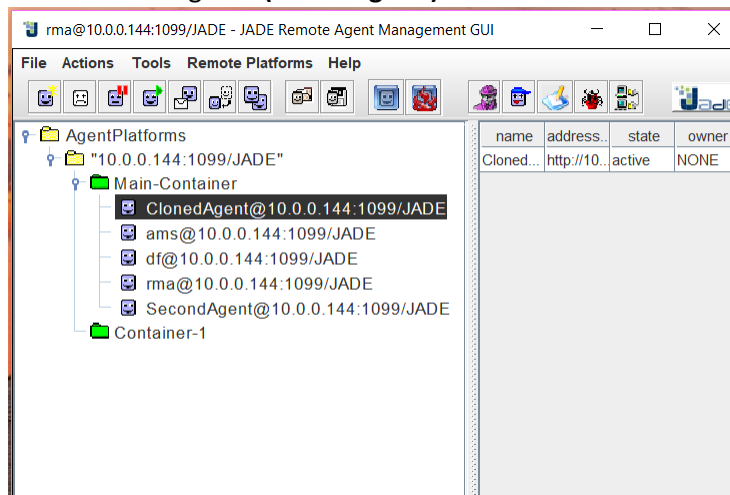
Agent swoje działanie wypisał w nowo utworzonym kontenerze :

```
Hello World.  
My name is SecondAgent
```

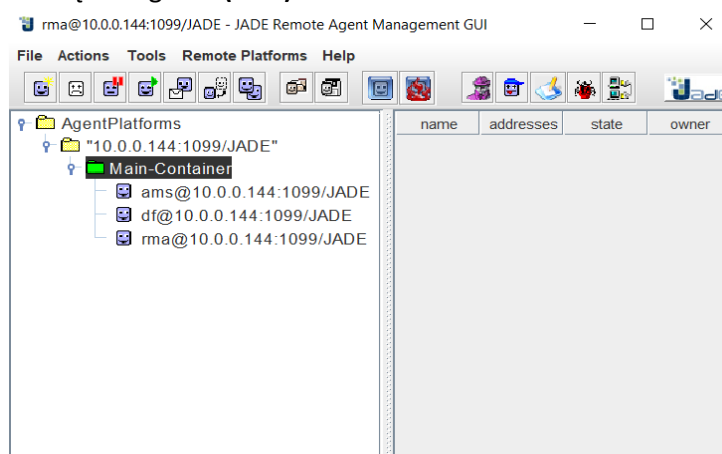
b. Przeniósłem agenta do innego kontenera ( **Migrate agent** ) :



c. Sklonowałem agenta ( **Clone agent** ) :



d. Usunąłem agenta ( **Kill** ) :



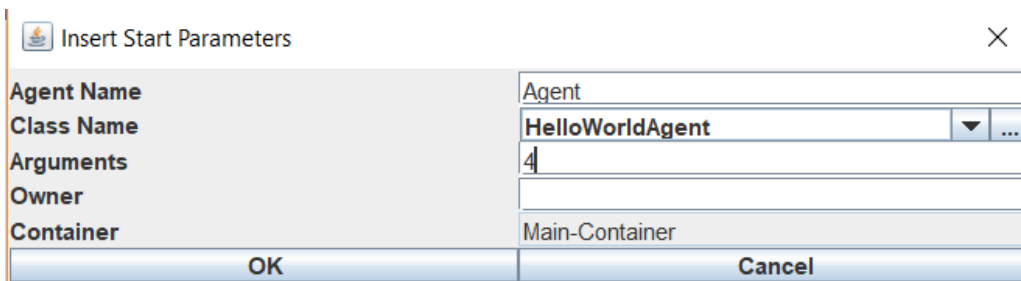
6. Zmodyfikowałem TimeAgent.java tak, aby agent usuwał się dopiero po 2 minutach oraz skompilowałem kod :

```
Agent newAgent: tick=110
Agent newAgent: tick=111
Agent newAgent: tick=112
Agent newAgent: tick=113
Agent newAgent: tick=114
Agent newAgent: tick=115
Agent newAgent: tick=116
Agent newAgent: tick=117
Agent newAgent: tick=118
Agent newAgent: tick=119
Agent newAgent: It's wakeup-time. Bye...
```

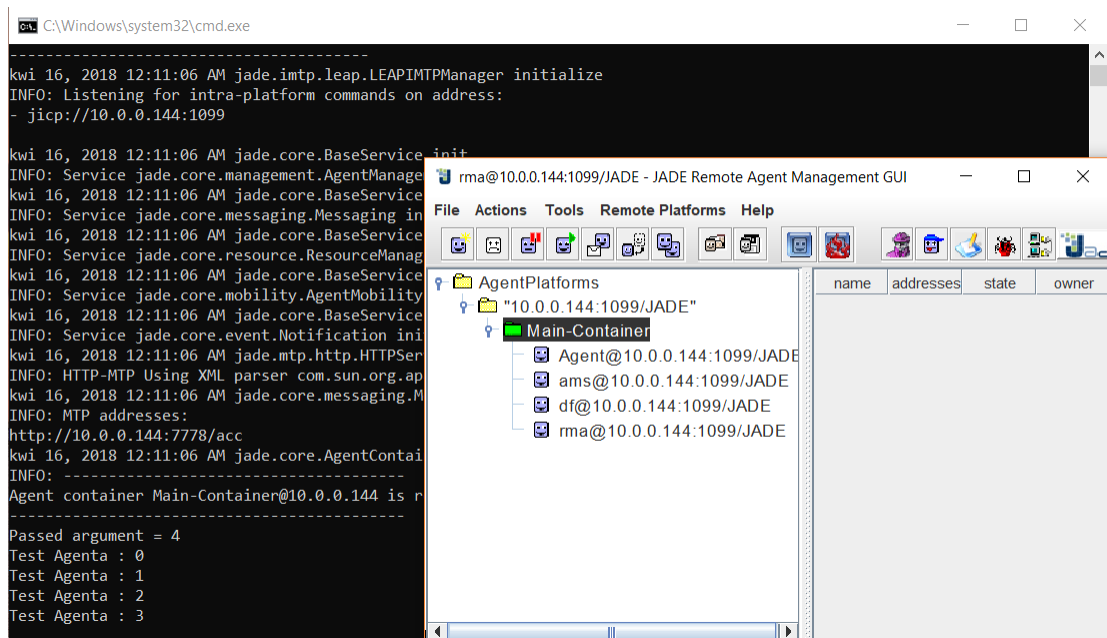
7. Przyłączyłem kontener na moim komputerze do zdalnego hosta znajdującego się na drugim komputerze za pomocą komendy :

**java jade.Boot -container -container-name C -host <adres IP hosta>**

- Poprosiłem o usunięcie mojego kontenera
  - Przyłączyłem raz jeszcze raz z dodaniem agenta klasy **TimeAgent**
  - Poprosiłem o sklonowanie agenta – Sklonowany agent będzie robić to samo co agent, który „pracuje”. Np. sklonowany agent będzie na takim samym etapie (tick) co jego pierwowzór.
  - Poprosiłem o migrację agenta – Agenci po migracji będą wykonywać swoją „pracę” w osobnych terminalach.
8. Podczas gdy brakuje tworzonego agenta, lub kontenera w którym moglibyśmy go umieścić, program zwróci błąd, że dana klasa nie istnieje i agent nie zostanie utworzony oraz że nie istnieje klasa kontenera w przypadku gdy chcemy użyć takiego, który nie istnieje.
9. Zmodyfikowałem plik **HelloWorldAgent.java** tak, aby agent wypisywał dotychczasowy komunikat tyle razy, ile zostanie podane w parametrach agenta :



Insert Start Parameters	
Agent Name	Agent
Class Name	HelloWorldAgent
Arguments	4
Owner	
Container	Main-Container
OK Cancel	



Agent wypisał komunikat na ekran 4 razy, czyli tyle ile podałem w argumentach.

10. Zmodyfikowałem program tak, aby wszystko co było w metodzie **main()** zostało wywołane w klasie agenta :

```

1  import jade.core.Agent;
2
3  public class HelloAgent extends Agent
4  {
5      protected void setup()
6      {
7          System.out.println("Hello World. ");
8          System.out.println("My name is "+ getLocalName());
9          this.main(null);
10     }
11
12     public static void main(String[] args) {
13         System.out.println("Main method test");
14     }
15
16 }

```

```

Hello World.
My name is newAgent
Main method test

```