

Temat: Podstawy JADE.

W ramach zajęć zrealizowałam następujące kroki:

1. Instalacja, konfiguracja ustawień i uruchomienie platformy JADE.
2. Pobranie i uruchomienie przykładowej klasy dostępnej w pobranej paczce *HelloWorldAgent.java*.
3. Modyfikacja pliku:
 - a) tak, aby agent nie usuwał się po wypisaniu tekstu na ekranie:

```
import jade.core.Agent;

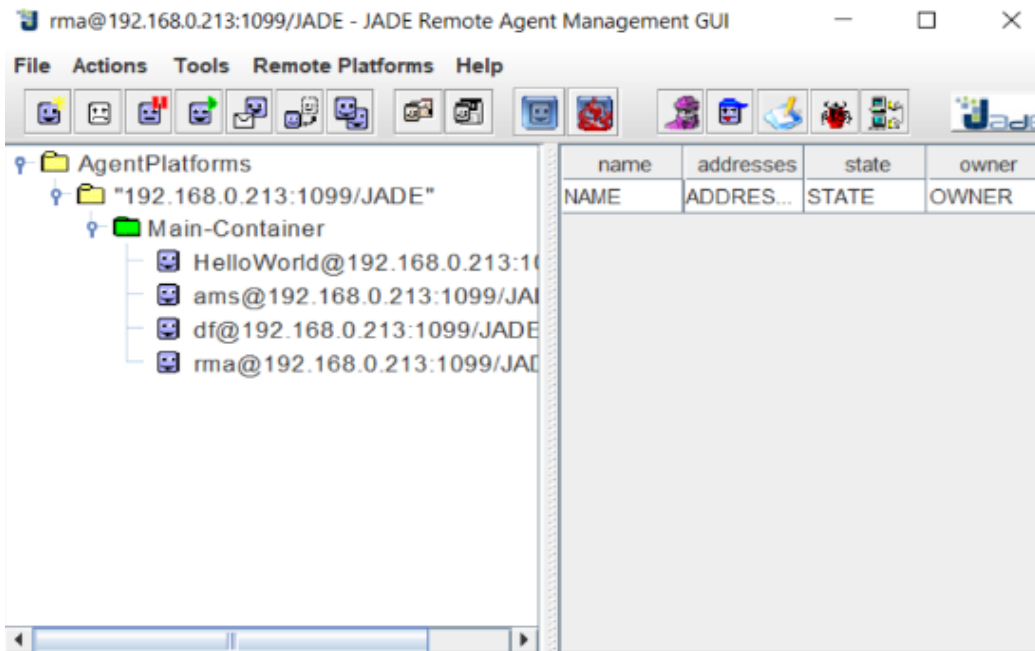
/**
 * This example shows a minimal agent that just prints "Hallo World!"
 * and then terminates.
 * @author Giovanni Caire - TILAB
 */
public class HelloWorldAgent extends Agent {

    protected void setup() {
        System.out.println("Hello World! My name is " + getLocalName());

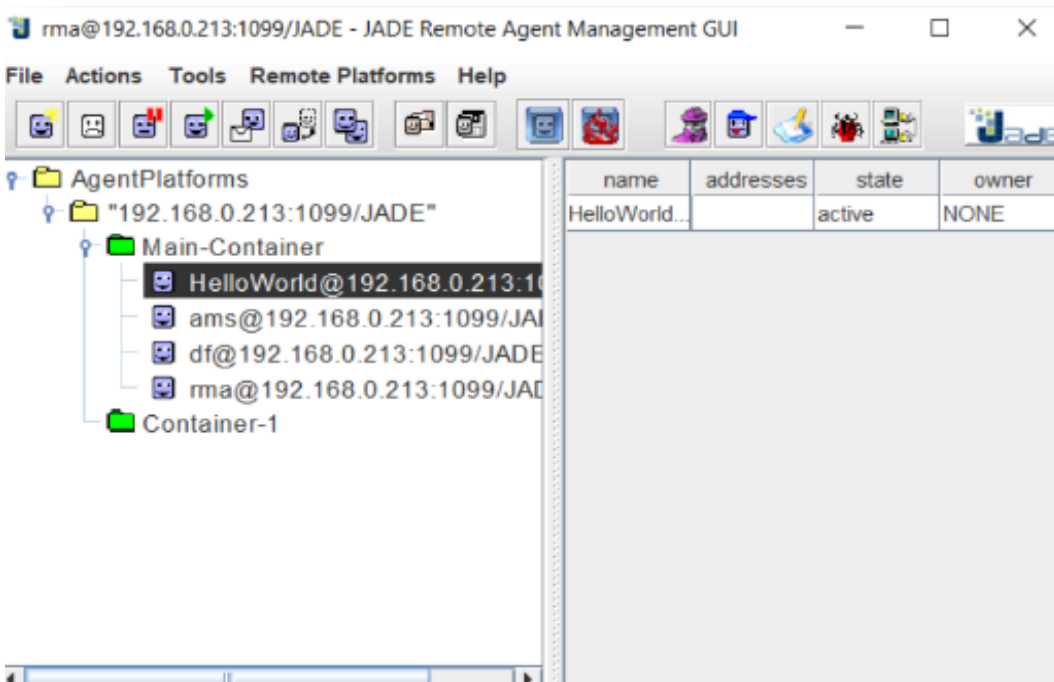
        // Make this agent terminate
        //doDelete();
    }
}
```

W tym celu zakomentowałam metodę *doDelete()*, która wymusza zakończenia działania. Po kombinacji i wyświetlenie agent jest wciąż widoczny:

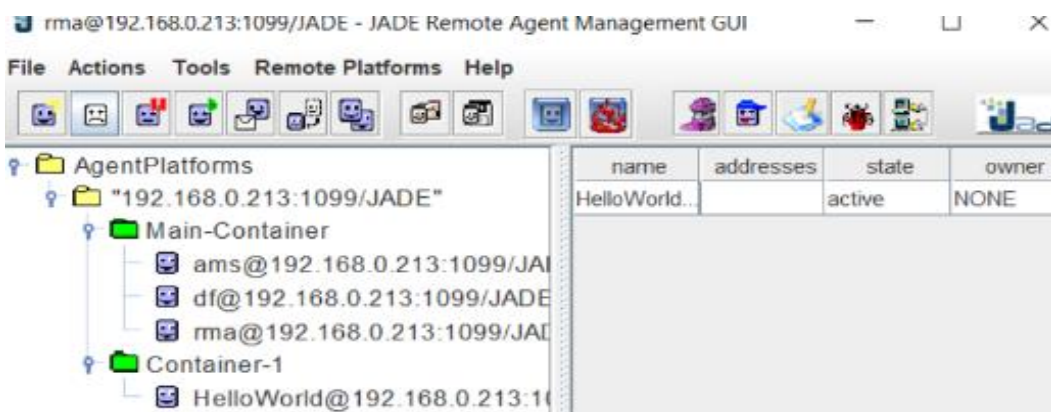
```
INFO: -----
Agent container Main-Container@192.168.0.213 is ready.
-----
Hello World! My name is HelloWorld
```



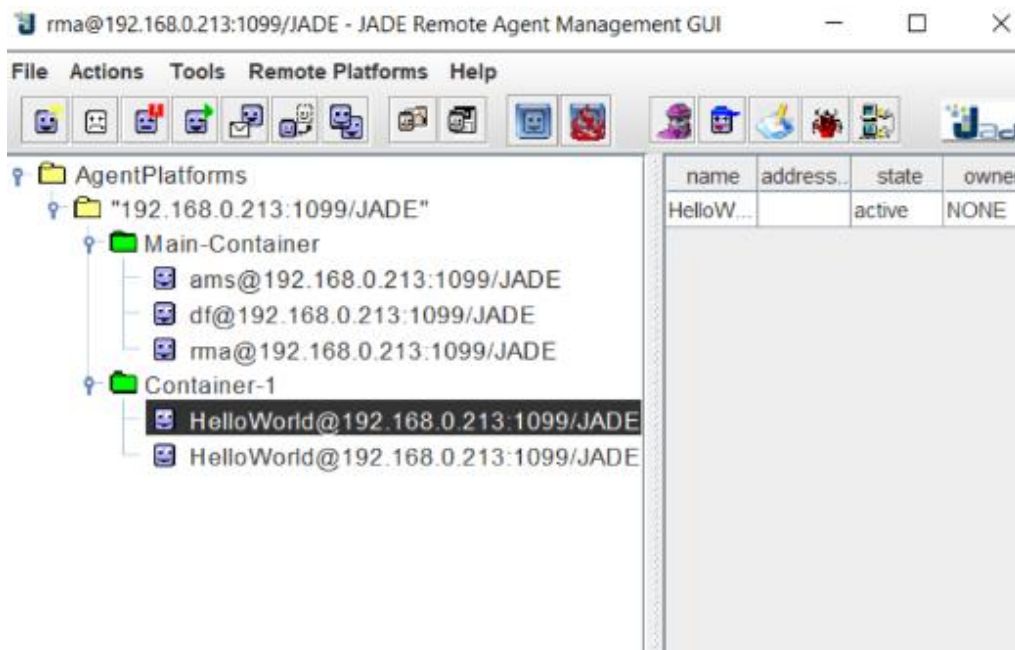
4 a). Kolejnym krokiem było dodanie jeszcze jednego kontenera (poza Main-Contener) i uruchomienie agenta klasy *HelloWorldAgent*:



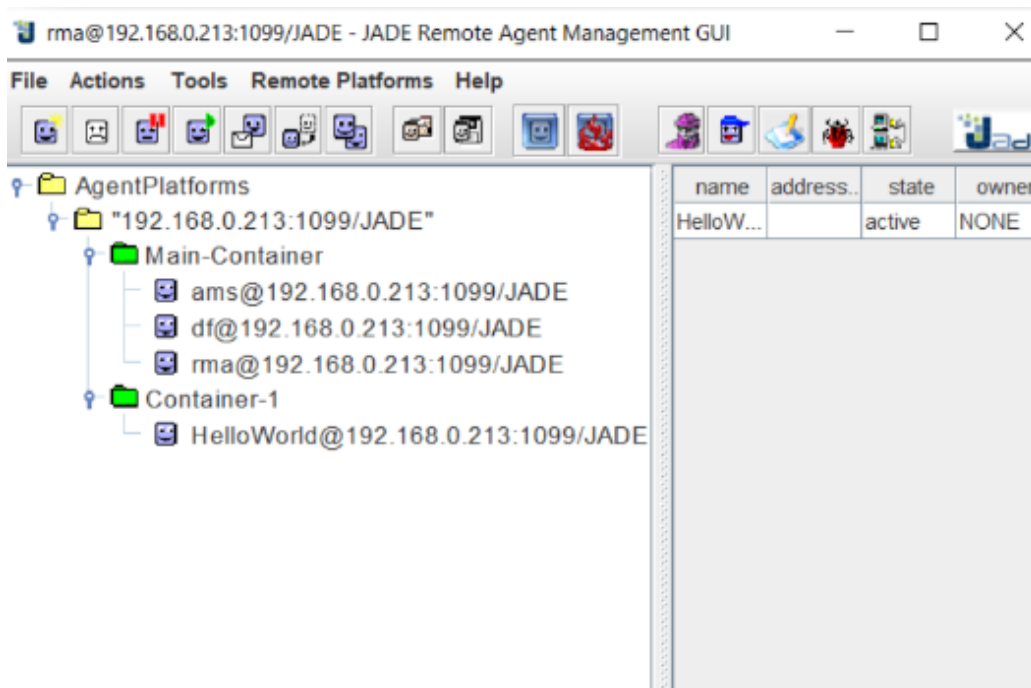
b. Przeniesienie agenta do innego kontenera (MIGRATE AGENT)



c). Sklonowanie agenta (CLONE AGENT)



d). Usunięcie agenta (KILL AGENT)



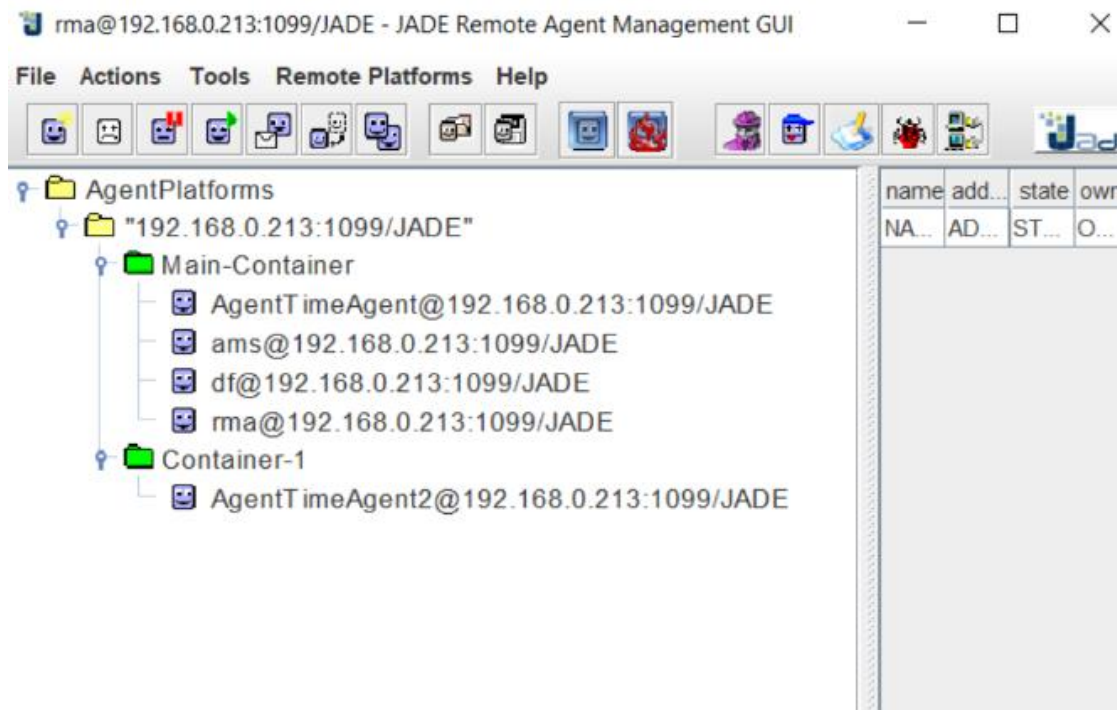
5. W dalszej części utworzyłam w IDE i sprawdziłam działanie TimeAgent.java.

```
public class TimeAgent extends Agent {  
  
    protected void setup() {  
        System.out.println("Agent "+getLocalName()+" started.");  
  
        // Add the TickerBehaviour (period 1 sec)  
        addBehaviour(new TickerBehaviour( a: this, period: 1000) {  
            protected void onTick() { System.out.println("Agent "+myAgent.getLocalName()+" : tick="+getTickCount()); }  
        });  
  
        // Add the WakerBehaviour (wakeup-time 10 secs)  
        addBehaviour(new WakerBehaviour( a: this, timeout: 10000) {  
            protected void handleElapsedTimeout() {  
                System.out.println("Agent "+myAgent.getLocalName()+" : It's wakeup-time. Bye...");  
                myAgent.doDelete();  
            }  
        });  
    }  
}
```

6. Modyfikacja TimeAgent.java tak, aby agent usuwał się dopiero po 2 minutach

```
// Add the WakerBehaviour (wakeup-time 10 secs)
addBehaviour(new WakerBehaviour( a: this, timeout: 120000) {
```

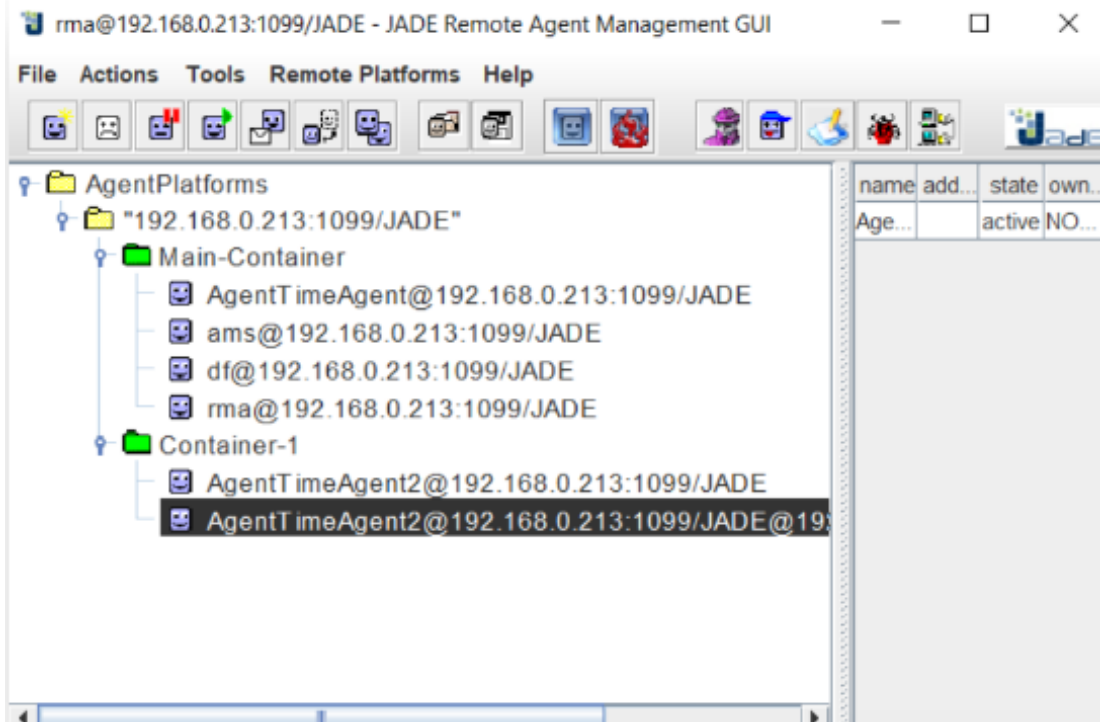
7. Przyłączenie kontener na moim komputerze do zdalnego hosta.



Po dodaniu jeszcze raz agenta uzyskałam dwóch odliczających 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...
```

8. Migracja agenta. (Container-1 -> Main-Container)

AgentTimeAgent2 został przeniesiony w 20s, po czym kontynuuje odliczanie w nowym kontenerze.

Wiersz polecenia - java jade.Boot -gui

```
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
```

Wiersz polecenia - java jade.Boot -gui

```
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
```

9. Modyfikacja *HelloWorldAgent.java*, żeby agent wypisywał komunikat konkretną liczbę razy w zależności od tego, co zostanie podane w parametrach tego agenta.

```
public class HelloWorldAgent extends Agent {

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

```
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
```