Magdalena Osiewicz, IO, gr 2.

Rozproszona sztuczna inteligencja.

Sprawozdanie nr 7.

Temat: Zachowania – część 1.

W ramach zajęć zrealizowałam następujące kroki:

1. Utworzenie klasy agenta o nazwie Klasa_1.

a) Agent zawsze na samym początku wypisuje „startuję”,
b) Agent zawsze przed swoim usunięciem wypisywać „zaraz się usunę”.

```
INFO: MTP addresses:
http://192.168.0.52:7778/acc
cze 16, 2018 12:56:09 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
startuje!
cze 16, 2018 12:56:23 AM jade.domain.ams killContainerAction
INFO: Agent ams - KillContainer request received from rma. Container=Main-Container@<Unknown Host>
zaraz sie usune!
cze 16, 2018 12:56:23 AM jade.core.messaging.MessageManager shutdown
INFO: MessageManager shutting down ...
cze 16, 2018 12:56:23 AM jade.core.Runtime$1 run
INFO: JADE is closing down now.
```

```
INFO: MTP addresses:
http://192.168.0.52:7778/acc
cze 16, 2018 12:57:05 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
startuje!
zaraz sie usune!
```

Powyżej wynik działania kolejno: dla usunięcia Agenta spowodowanego usunięciem Container i bezpośrednim usunięciem Agenta za pomocą opcji *kill*. Niezależnie od sposobu usunięcia Agenta komunikat zostaje wyświetlony.

Za wyświetlnenie komunikatu tuż po inicjalizacji Agent odpowiada metoda *setup()*, natomiast komunikat wyświetlony tuż po usunięcie Agenta jest wynikiem działania metody *takeDown()*.

2. Utworzenie klasy agenta o nazwie Klasa_2 na podstawie kodu Klasa_1.

Zadania agentów implementowane są w ramach *zachowań*. Zachowaniem jest klasa dziedzicząca po klasie *jade.core.behaviours.Behaviour*. Klasa ta musi implementować metody *action()*, w której kodowane jest zadanie agenta oraz *done()* zwracającą wartość typu boolean oznaczającą zakończenie zadania (true) lub nie (false).
Agent może być wyposażony w wiele zachowań, z których każde przypisywane za pomocą metody agenta *addBehaviour()*. Zachowania mogą być dodawane w metodzie *setup()* agenta lub wewnątrz innych zachowań. Agenta posiada kolejkę dodanych zachowań, które wykonują się bez wywłaszczeń, co oznacza, że wykonanie metody *action()* nie może być przerwane przez inne zachowanie danego agenta. Wszystkie zachowania wykonywane są w ramach jednego wątku.

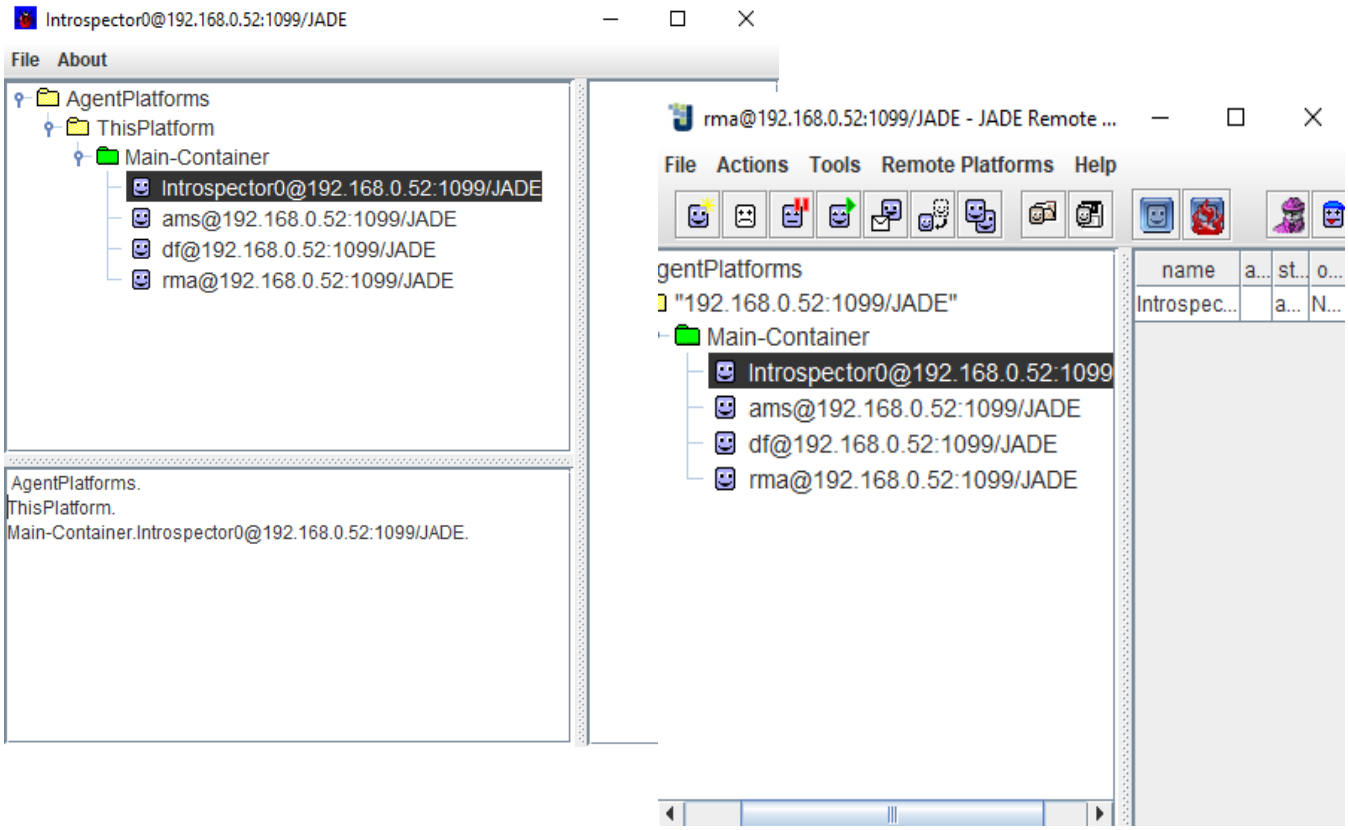
a). Dodanie do agenta zachowania polegające na jednokrotnym wykonaniu operacji wypisania na ekranie słowa „wykonuję”:

```java
import jade.core.Agent;
import jade.core.behaviours.OneShotBehaviour;

public class Klasa_2 extends Agent{

    protected void setup() {
        System.out.println("startuje!");

        addBehaviour(new OneShotBehaviour() {
            @Override
            public void action() {
                System.out.println("wykonuje!!!");
            }
        });
    }

    protected void takeDown() {
        System.out.println("zaraz sie usune!");
    }
}
```

Biblioteka JADE dostarcza kilku klas modelujących chrakterystyczne zachowania: (jade.core.behaviours.OneShotBehaviour) daje możliwość jednokrotne wykonania określonego zadania.

```
INFO: MTP addresses:
http://192.168.0.52:7778/acc
startuje!
wykonuje!!!
cze 16, 2018 6:24:01 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
zaraz sie usune!
```

b). Kolejnym krokiem jest uruchomienie agenta introspektora.



IntrospectorAgent, czyli dodatkowe narzędzia dołączone do dystrybucji JADE w ramach interfejsu graficznego - pozwala na wybranie agenta i monitorowanie jego zachowań oraz wiadomości jakie do niego przychodzą i są przez niego wysyłane, możliwy jest zarówno podgląd kolejki wiadomości agenta (oczekujących na odebranie) jak i tych odebranych. W tym przypadku jednak nie ma możliwości zaobserowania zachowania (nie są one widoczne), ponieważ dzieje się ono bardzo szybko.

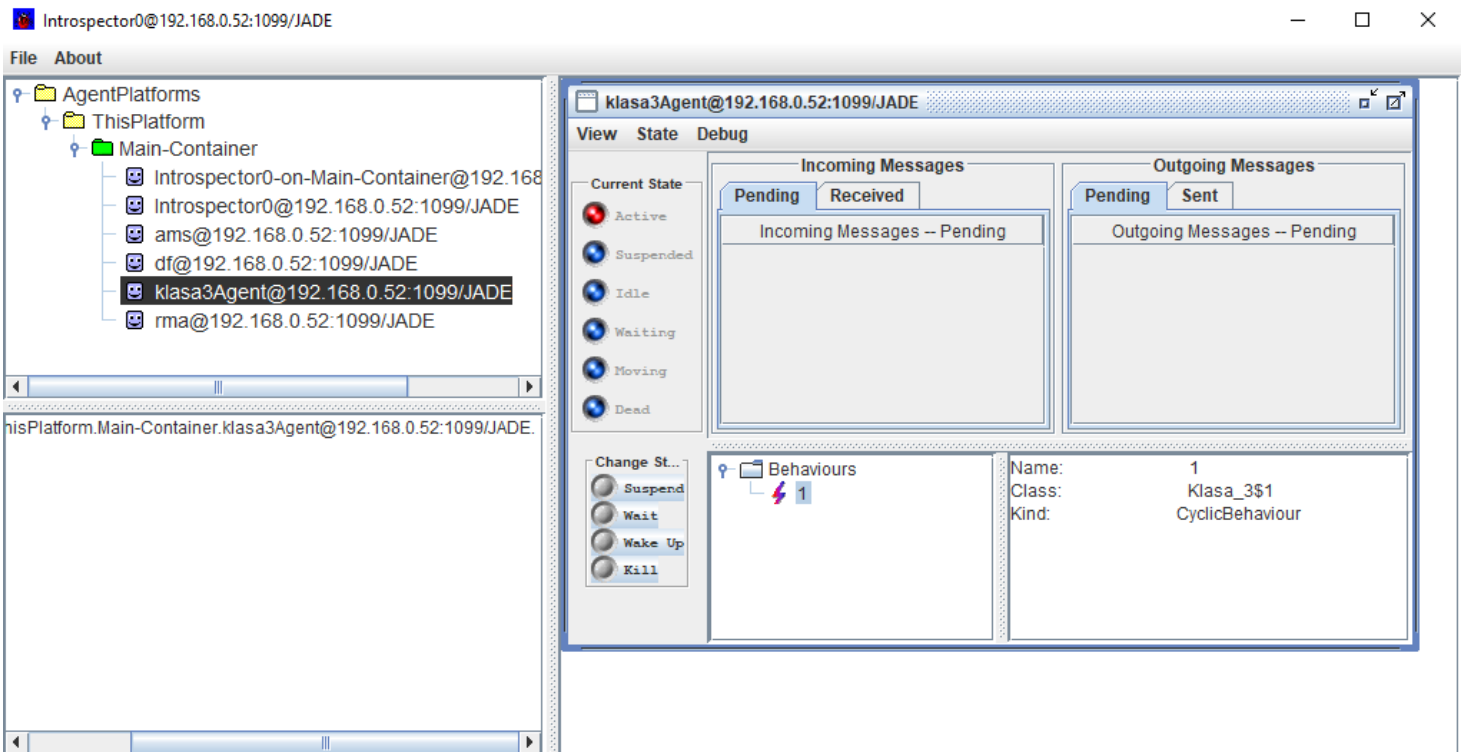
3. Utworzenie klasy agenta o nazwie Klasa_3 na podstawie kodu klasa_1.

a). Dodanie do agenta zachowania polegającego na wielokrotnym (cyklicznym) wykonaniu operacji wypisania na ekranie słowa „wykonuję” daje wynik:

```
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
wykonuje!
zaraz sie usune!
```

Cykliczne wykonywanie określonego zadania do momentu agent Klasy_3 zapewnia: (*jade.core.behaviours.CyclicBehaviour*)

b). Jakie zachowania są widoczne po uruchomieniu Agenta Introspectora?



Można zaobserwować stan Agenta, informacje dotyczące przysyłanych wiadomości(w tym przypadku żadne), z jakim Agentem mamy do czynienie, jaką klasą oraz z jakim rodzajem zachowania.

4. Utworz klasę agenta o nazwie klasa_4 na podstawie kodu klasa_1.

Dodanie do agenta zachowania „generycznego”, polegające na wykonaniu trzech kroków:
a). W pierwszym kroku wypisanie „pierwszy krok”;
b). W drugim kroku wypisuje „drugi krok”,
c). W trzecim kroku wypisuje „trzeci krok” i zachowanie zostaje usunięte z puli zachowań agenta:

```
INFO: MTP addresses:
http://192.168.0.52:7778/acc
startuje!
pierwszy krok! :)
drugi krok!
trzeci krok!
cze 16, 2018 7:51:24 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
------------------------------------------
zaraz sie usune!
```

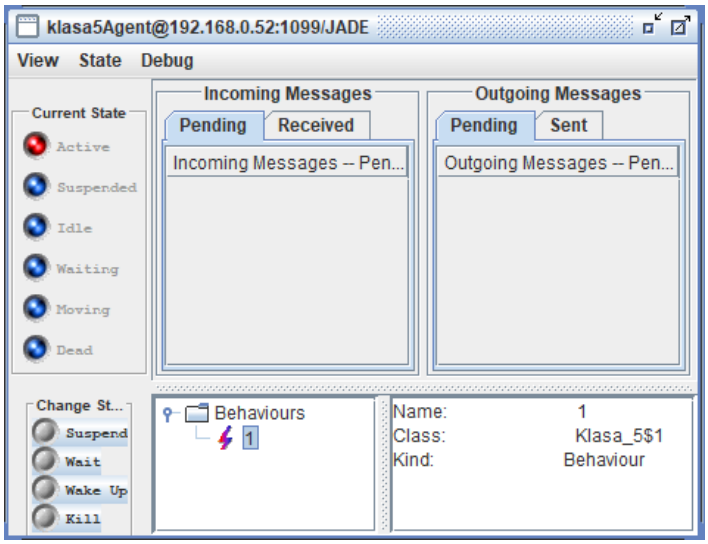
W tym celu wykorzystana została instrukcja switch oraz wywołania metody *done()* zwracającej prawdziwą wartość w momencie wykonywania kroku trzeciego, tuż po wypisaniu „trzeci krok!”.

5. Utworzenie agenta o nazwie klasa_5 na podstawie kodu Klasa_1.

Dodanie do agenta zachowania, które będzie polegało na pobieraniu z klawiatury liczby całkowitej. Jeśli użytkownik poda liczbę ujemną, to zachowanie zostaje usunięte.

```
INFO: MTP addresses:
http://192.168.0.52:7778/acc
startuje!
cze 16, 2018 8:09:58 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
Podaj liczbe calkowita: 8
liczba dodatnia!
Podaj liczbe calkowita: -6
liczba ujemna- usuwam Agenta!
zaraz sie usune!
```

```
addBehaviour( new Behaviour()
{
    public void action() {
        System.out.print("Podaj liczbe calkowita: ");
        int number = scanner.nextInt();
        if(number>0)
        {
            System.out.println("liczba dodatnia!");
        }
        if(number<0)
        {
            System.out.println("liczba ujemna- usuwam Agenta! ");
            removeBehaviour( b: this);
        }
    }

    @Override
    public boolean done() {
        return false;
    }
});
```

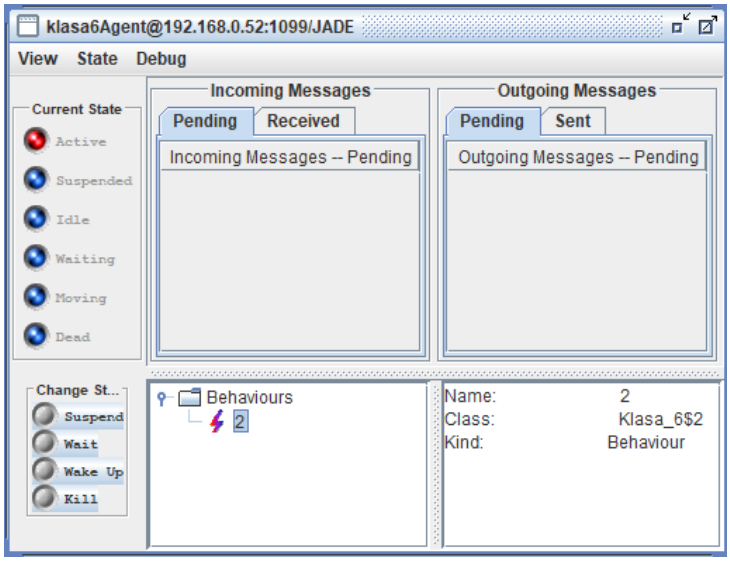


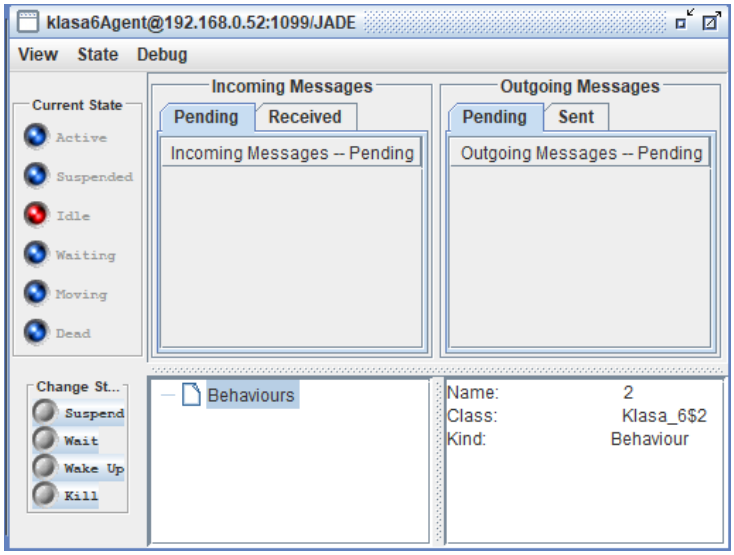
6. Utworz klasy agenta o nazwie Klasa_6 na podstawie kodu Klasa_5.

Modyfikacja kodu w taki sposób, aby zawsze zachowanie na początku wypisywało „zachowanie startuje”, a na samym końcu wypisywało „zachowanie zakończone”.

```
INFO: MTP addresses:
http://192.168.0.52:7778/acc
cze 16, 2018 8:22:11 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
startuje!
zachowanie startuje
Podaj liczbe calkowita: 45
liczba dodatnia!
Podaj liczbe calkowita: 2
liczba dodatnia!
Podaj liczbe calkowita: 32
liczba dodatnia!
Podaj liczbe calkowita: 12
liczba dodatnia!
Podaj liczbe calkowita: -160
liczba ujemna- usuwam zachowanie Agenta!
zachowanie zakonczone
zaraz sie usunÄ?!!
```

Widoczne zachowania kolejno: przed podaniem liczby ujemnej oraz po.






```
addBehaviour( new Behaviour()
{
    public void action() {
        System.out.println( "zachowanie startuje" );
    }
    @Override
    public boolean done() {
        return true; }
});
```

Modyfikacja kodu polegała głównie na dodaniu jeszcze jednego zachowania odpowiadającego za wypisanie komunikatu „zachowanie startuje!”. W przeciwieństwie do zachowaniu następującego tuż po nim metoda *done()* wraca wartośc true.

7. Utworzenie klasy agenta o nazwie Klasa_7 na podstawie kodu Klasa_4.

Dodanie do istniejącego zachowania „generycznego” dwóch kolejnych:
a). Pierwsze jest dodawane na poziomie metody *setup()* agenta i polega na jednokrotnym wypisaniu „pierwsze”:

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

    addBehaviour(oneShotBehaviour1);
```

```
addBehaviour(new Behaviour() {
    @Override
    public void action() {

        switch(step){
            case 0:
                OneShotBehaviour oneShotBehaviour2=new OneShotBehaviour() {
                    @Override
                    public void action() {
                        System.out.println("drugie");
                    }
                };
                addBehaviour(oneShotBehaviour2);
                System.out.println("pierwszy krok.");
                step++;
                break;
```

b). Drugie jest dodane z poziomu zachowania „generycznego” – dokładnie jest dodane w pierwszym kroku i polega na jednokrotnym wypisaniu „drugie”.

W wyniku otrzymujemy:

```
INFO: MTP addresses:
http://192.168.0.52:7778/acc
startuje!
pierwsze
pierwszy krok.
drugi krok.
cze 16, 2018 8:44:43 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
drugie
trzeci krok.
zaraz sie usune!
cze 16, 2018 9:06:57 AM jade.domain.ams killContainerAction
INFO: Agent ams - KillContainer request received from rma. Container=Main-Container@<Unknown Host>
cze 16, 2018 9:06:58 AM jade.core.messaging.MessageManager shutdown
INFO: MessageManager shutting down ...
cze 16, 2018 9:06:58 AM jade.core.Runtime$1 run
INFO: JADE is closing down now.
```

Kolejność wykonywania wypisywania komunikatów wynika tego, że Agenta posiada kolejkę dodanych zachowań, które wykonują się bez wywłaszczeń, co oznacza, że wykonanie metody *action()* nie może być przerwane przez inne zachowanie danego agenta. Wszystkie zachowania wykonywane są w ramach jednego wątku.

8. Utworzenie klasy agenta o nazwie Klasa_8 na podstawie kodu Klasa_1.

Dodanie zachowania do agenta, które spowodują:
a). Wypisanie „mały tick” co 2 sekundy,
b). Wypisaniu „duży tick” co 5 sekund,
c). Po 50 sekundach usunięcie zachowania z punktu b,
d). Po 100 sekundach usunięcie całego agenta. Przeanalizuj działanie agenta z użyciem Introspektora.

(jade.core.behaviours.TickerBehaviour) pozwoliło na wykonywanie zadania z określoną częstotliwością, którą podajemy jako argument konstruktora, kolejno 2s dla obiektu smallTicket typu Behaviour oraz 5s dla obiektu bigTick.

```
protected void setup() {
    System.out.println("startuje!");

    //a)co 2s "maly Tick"
    Behaviour smallTick=new TickerBehaviour( a: this, period: 2000) {
        @Override
        protected void onTick() {
            System.out.println("maly tick");
        }
    };

    //b) co 5s "duzy Tick"
    final Behaviour bigTick=new TickerBehaviour( a: this, period: 5000) {
        @Override
        protected void onTick() {
            System.out.println("duzy tick");
        }
    };
    addBehaviour(smallTick);
    addBehaviour(bigTick);
```

Kolejno za pomocą (jade.core.behaviours.WakerBehaviour) nastąpiło usunięcie zachowania przypisanego do obiektu bigTick odpowiadającego za wypisanie komunikatu „duży tick”. WakerBehaviour pozwala na wykonanie zadania w określonym punkcie czasowym, co podano jako argument.

Następnym krokiem jest usunięcie agenta za pomocą metody *doDelete()* wywołanej po 100sekundach.

```
addBehaviour(new WakerBehaviour( a: this, timeout: 50000) {
    protected void handleElapsedTimeout() {
        removeBehaviour(bigTick);
    }
});

addBehaviour(new WakerBehaviour( a: this, timeout: 100000) {
    protected void handleElapsedTimeout() {
        myAgent.doDelete();
    }
});
```

```
startuje!
cze 16, 2018 9:39:28 AM jade.core.AgentContainerImpl joinPlatform
INFO: ----------------------------------------
Agent container Main-Container@192.168.0.52 is ready.
--------------------------------------------
maly tick
maly tick
duzy tick
maly tick
maly tick
duzy tick
maly tick
maly tick
maly tick
duzy tick
maly tick
maly tick
duzy tick
maly tick
maly tick
maly tick
duzy tick
maly tick
maly tick
maly tick
duzy tick
maly tick
maly tick
duzy tick
maly tick
maly tick
maly tick
duzy tick
maly tick
maly tick
maly tick
maly tick
maly tick
maly tick
maly tick
maly tick
```

Jako wynik działania otrzymujemy:
-wypisanie „mały tick” co dwie sekundy i „duzy tick” co pięć sekund, co odpowiada schematowi pokazanemu na rysunku obok.
Przez pierwszych 50 sekund można zaobserować naprzemienne wypisywanie się tych dwóch komunikatów.
Po tym czasie nieustannie wyświetlany jest komunikat „mały tick”, co świadczy o powodzeniu zachowania polegającego na usunięcia zachowania bigTick.
Po 100 sekundach z kolei nastepuje usunięcie agenta i zaprzestanie wypisywania.

Na podstawie działania introspektora można zauważyc, że przez pierwszych 50s miało miejsce widoczne są wszystkie zdarzeńnia, a potem już tylko 1 i 4.: