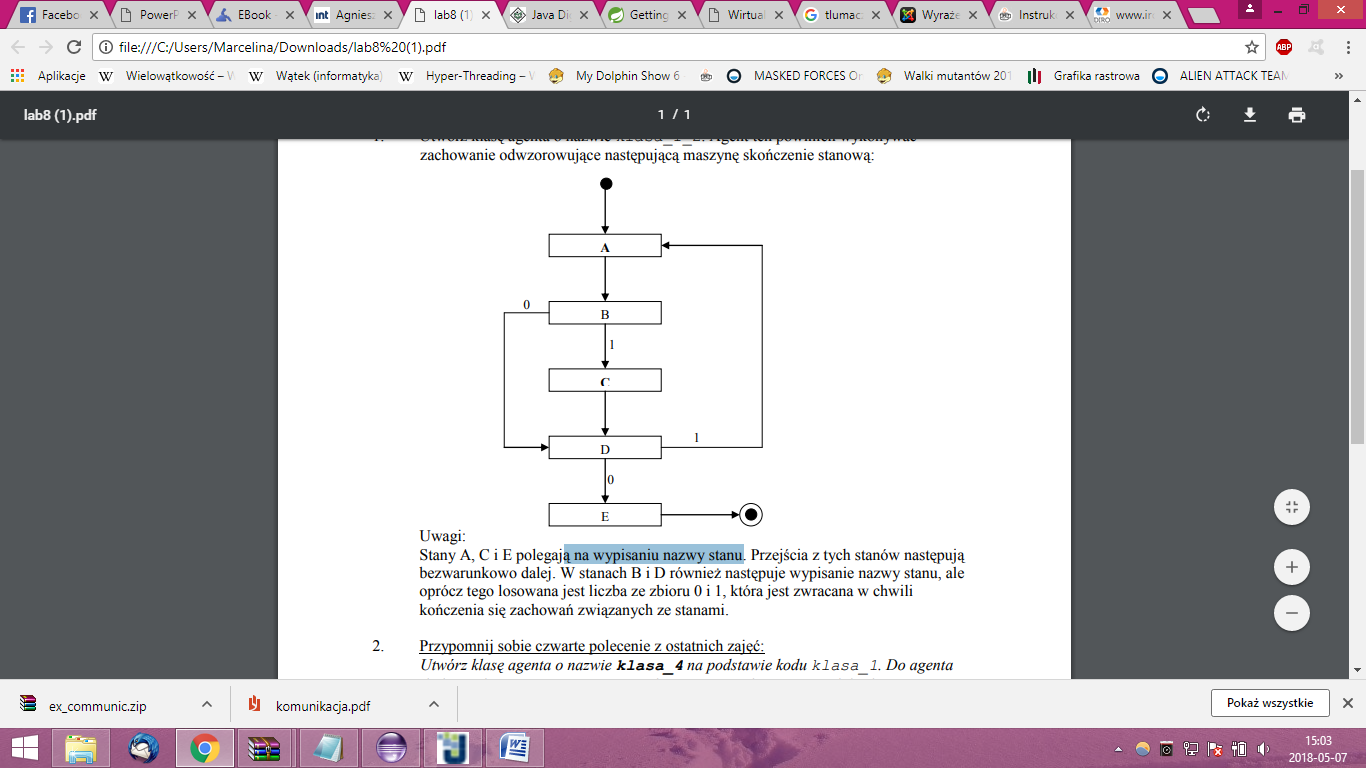
Marcelina Bałamut,

IO, gr1

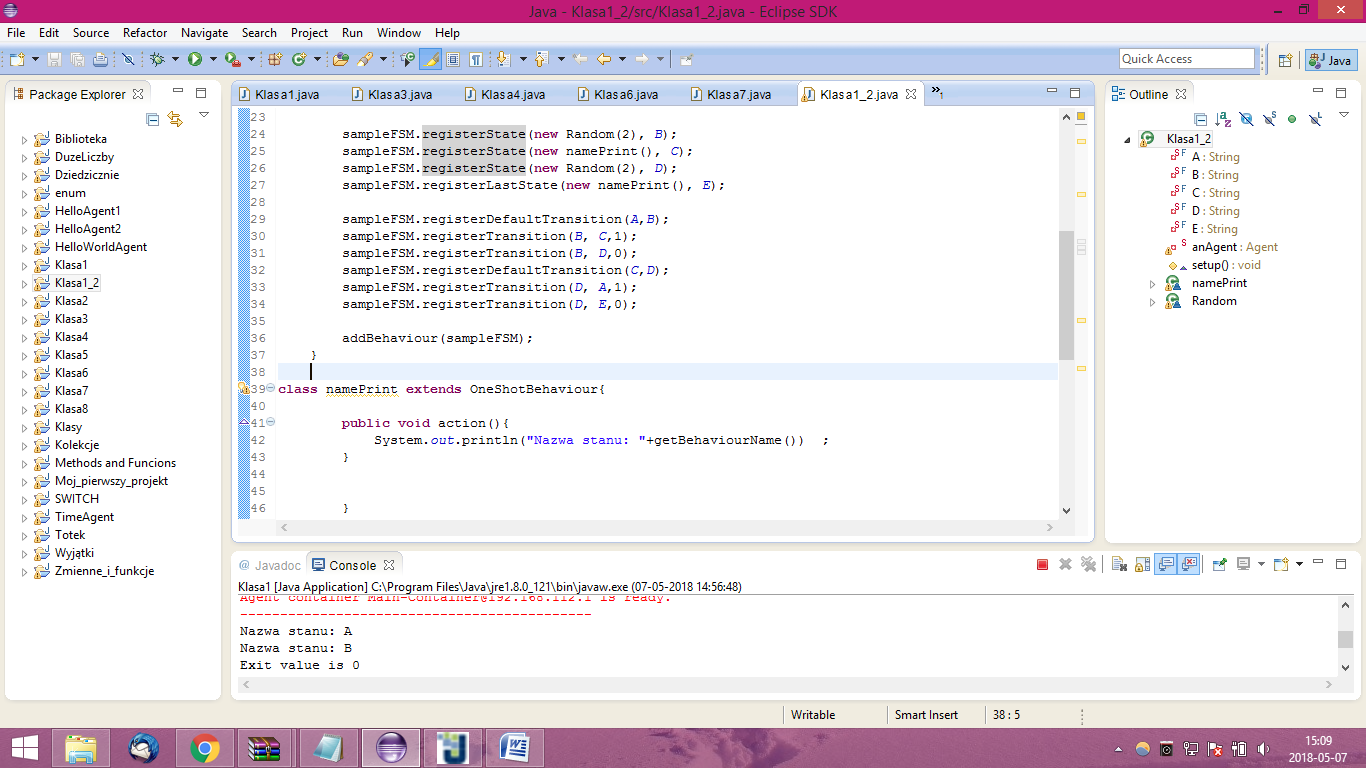
***Sprawozdanie z zagadnienia nr 8***

Pierwszym krokiem było stworzenie Klasy agenta Klasa1\_2. Agent ten odwzorowuje następującą maszynę stanową:

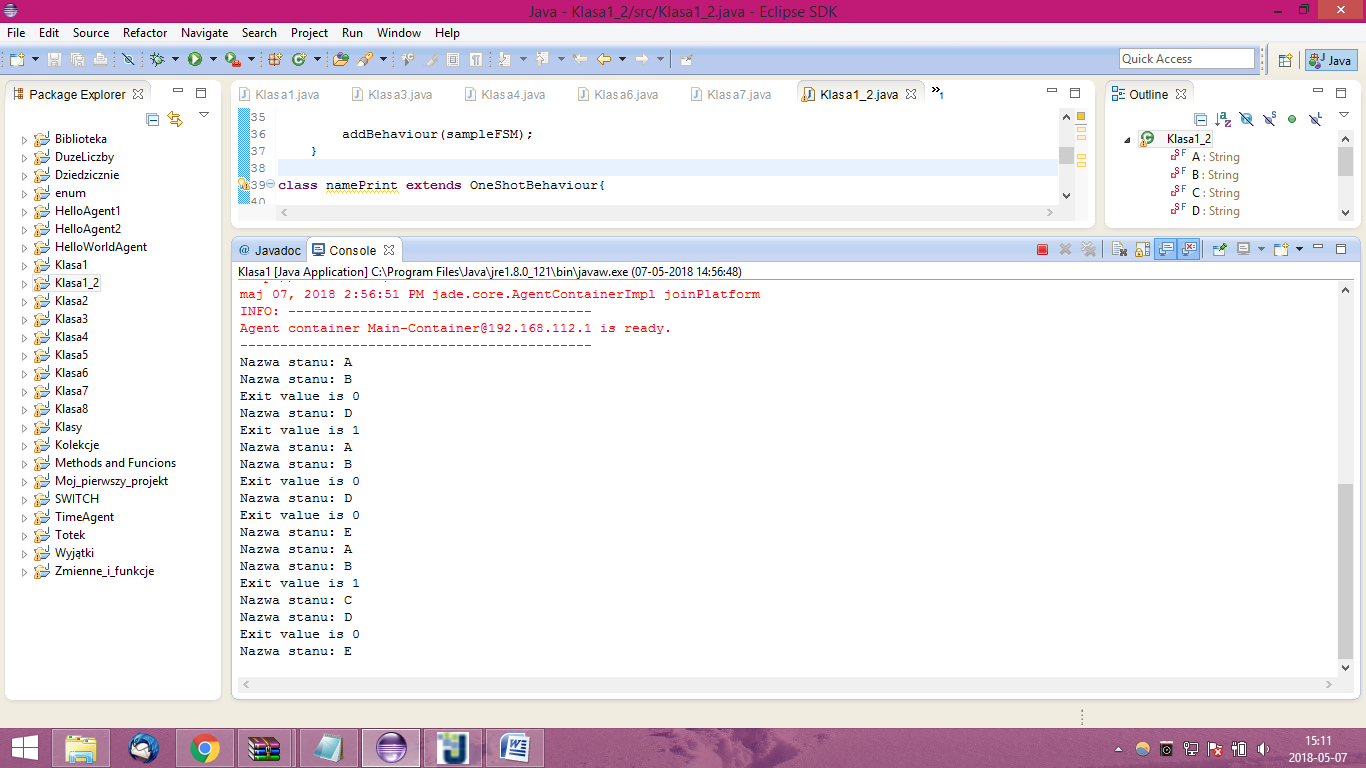


W stanach B i D następuje wypisanie nazwy stanu, oraz losowana jest liczba ze zbioru 0 i 1, która jest zwracana w chwili kończenia się zachowań związanych ze stanami. Natomiast stany A, C i E polegają wyłącznie na wypisaniu nazwy stanu.

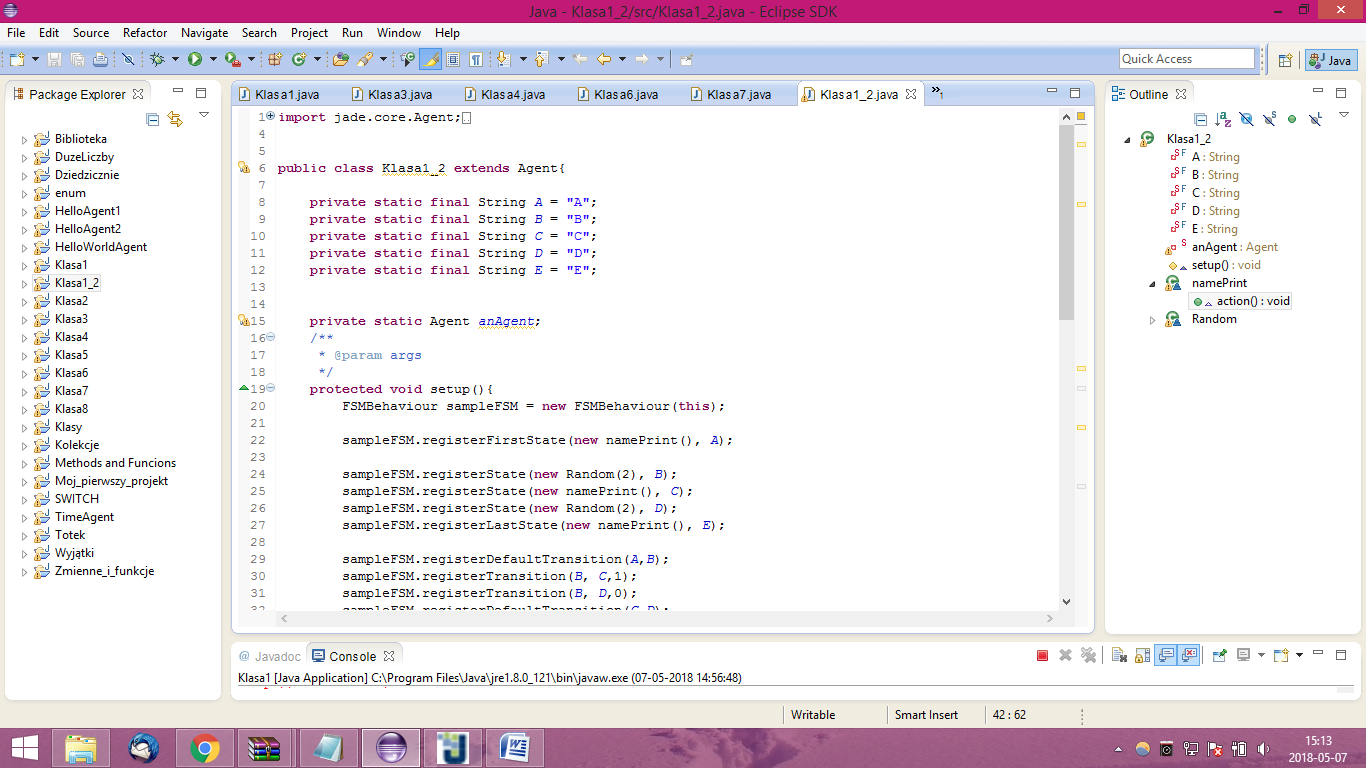
Do tego celu stworzyłam stany od A do E nadając im odpowiednie nazwy oraz zachowanie FSMBehaviour, dla którego wywołałam funkcje regoisterState w odpowiednich wersjach ( dla pierwszego stanu regoisterFirstState a dla ostatniego regoisterLastState) przyjmujące jako argument nazwę stanu i zachowanie dla tego stanu. Kolejno wywołałam następujące funkcje, aby określić zachowanie się maszyny stanowej, na ich podstawie maszyna wie który stan ma się wykonać:

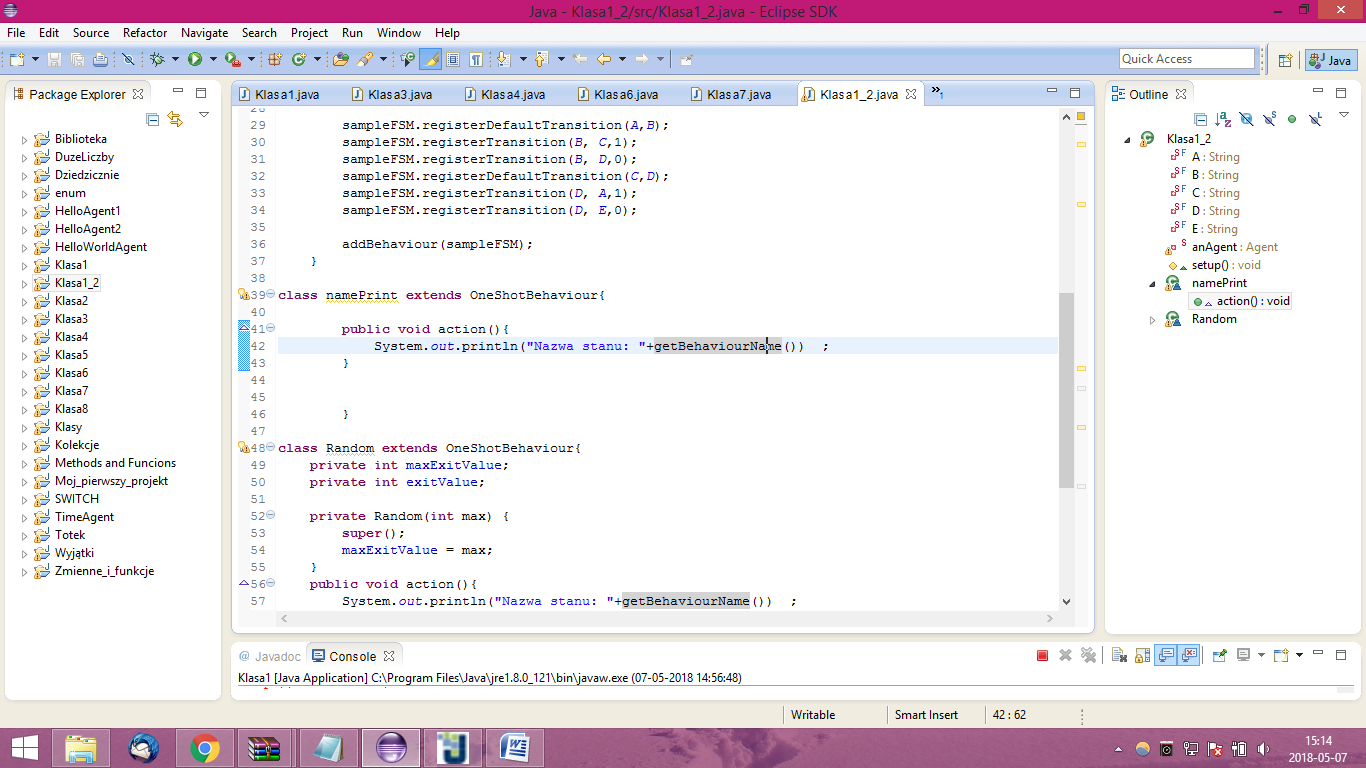


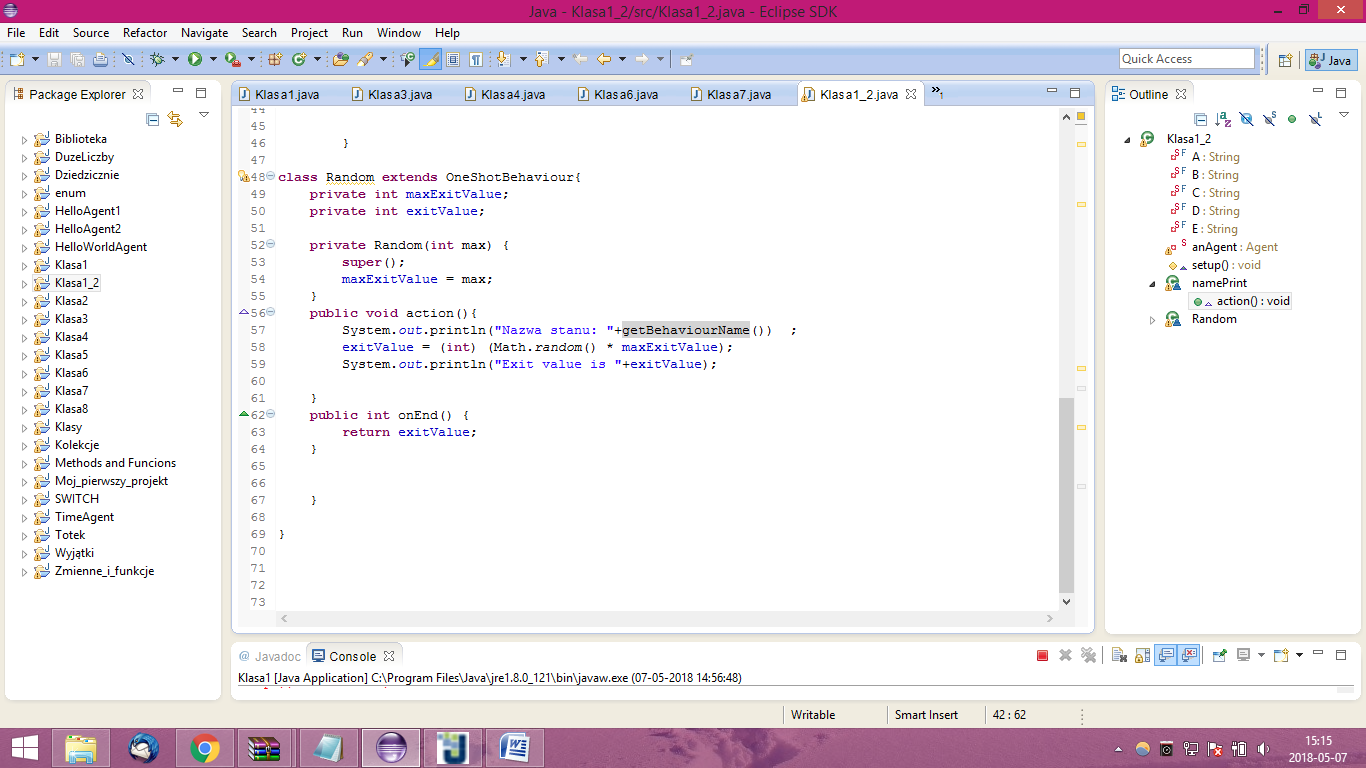
Argumentami są stan pierwszy , stan który ma się wykonać i ewentualny wynik warunku. Wynik programu wywołany dwukrotnie przedstawia się następująco:



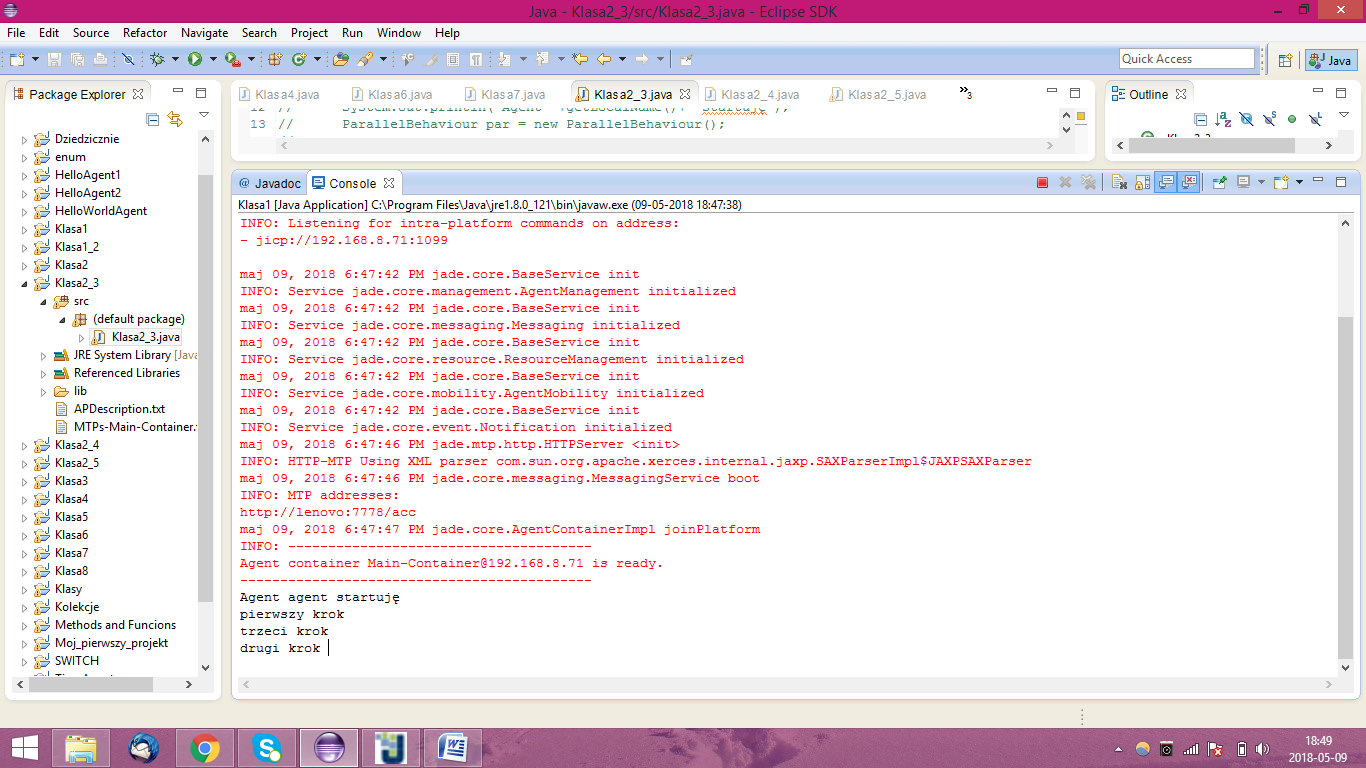
Listing kodu:





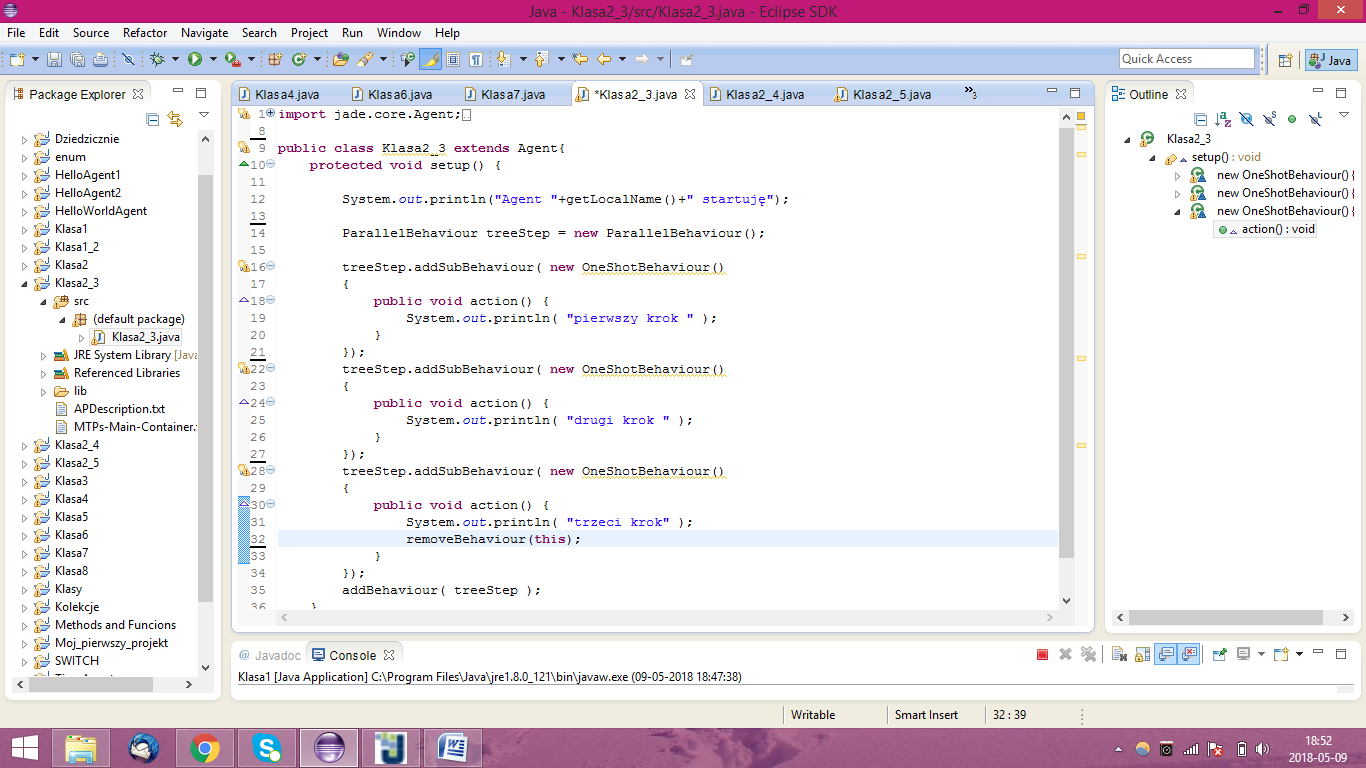


Kolejnym zadaniem było dodanie zachowania polegającego na równoległym wykonaniu trzech zachowań „generycznych” do Klasy 4 tworzonej w poprzednim zagadnieniu, która w pierwszym kroku wypisuje „krok pierwszy” w drugim”krok drugi” a w trzecim”krok pierwszy” i zadanie zostaje usunięte. Do tego celu utworzyłam obiekt ParallelBehaviour, a nastepnie wywołałam trzy razy funkcję addSubBehaviour jako argument kroki w zachowaniu generycznym. Wynik programu jest następujący:

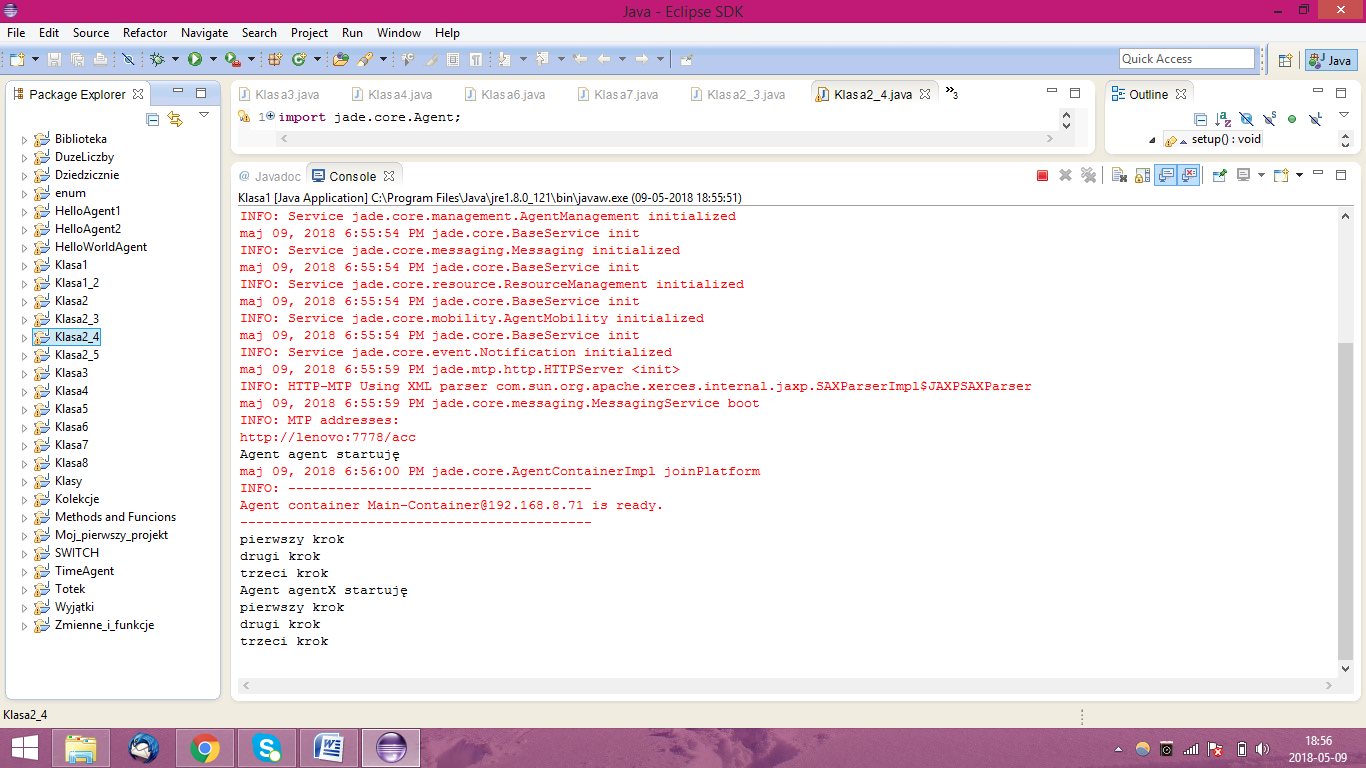


Widać, że zachowania nie wykonują się po kolei ale mieszają się.

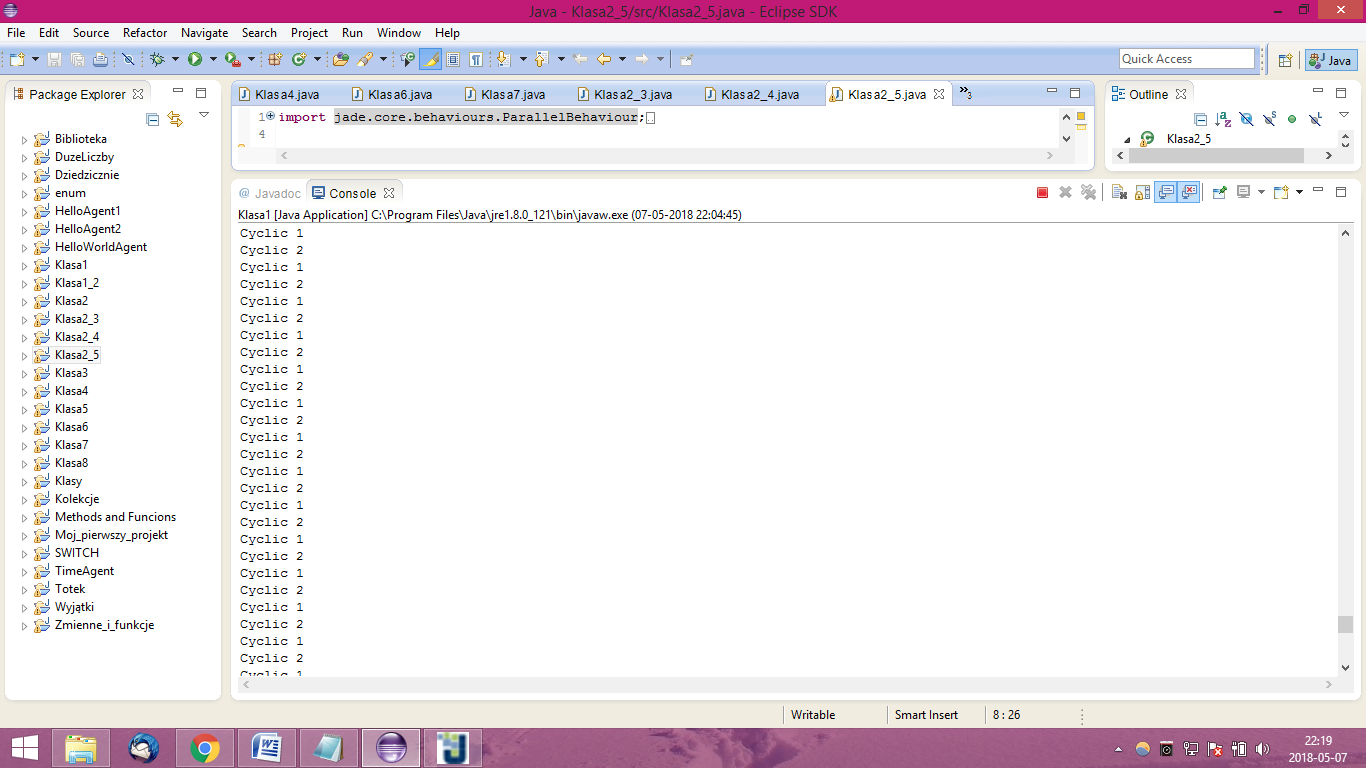
Listing kodu:



W kolejnym kroku utworzyłam klasę Klasa2\_4, gdzie kod przekopiowałam z klasy poprzedniej zmieniając jedynie obiekt z typu ParallerBehaviour na SequentialBehaviour, co spowodowało sekwencyjne wykonanie kodu:



Ostatnim krokiem było utworzenie agenta który wykonuje dwa zadania cyklicznie wypisując „Cyclic 1” i „Cyclic2” w dwóch osobnych watkach. W tym celu stworzyłam obiekt klasy ParallelBehaviour i dodałam do niej zachowania cykliczne. Jak widać na poniższym screenie zadania mieszają się ze sobą.



Listing kodu:

