**DeadLock Philosophers**

**By**

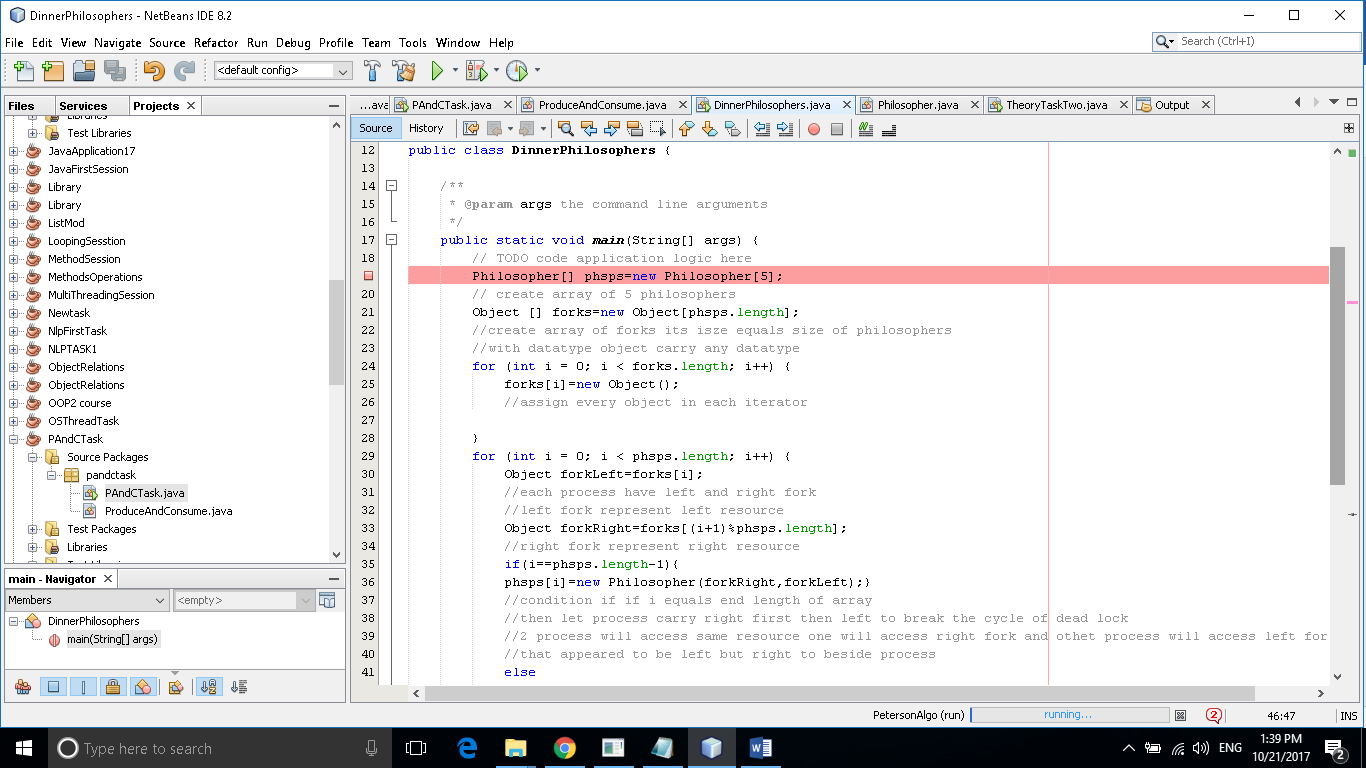
**Alhaitham Gamal**

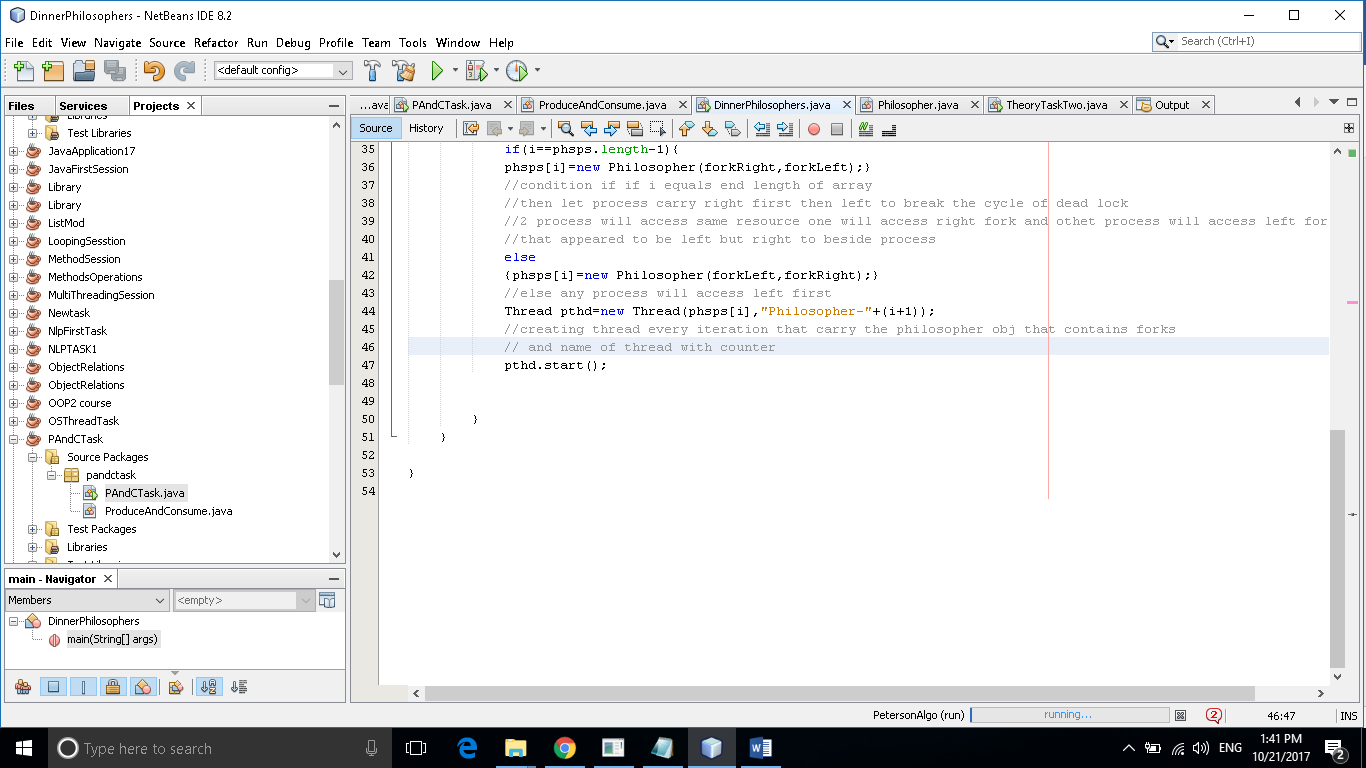
**20143114**

**Problem**: Suppose you have 5 dishes (resources)owned by 5 philosophers (process) each philosopher want to think, eat and finish without interfering with other process.

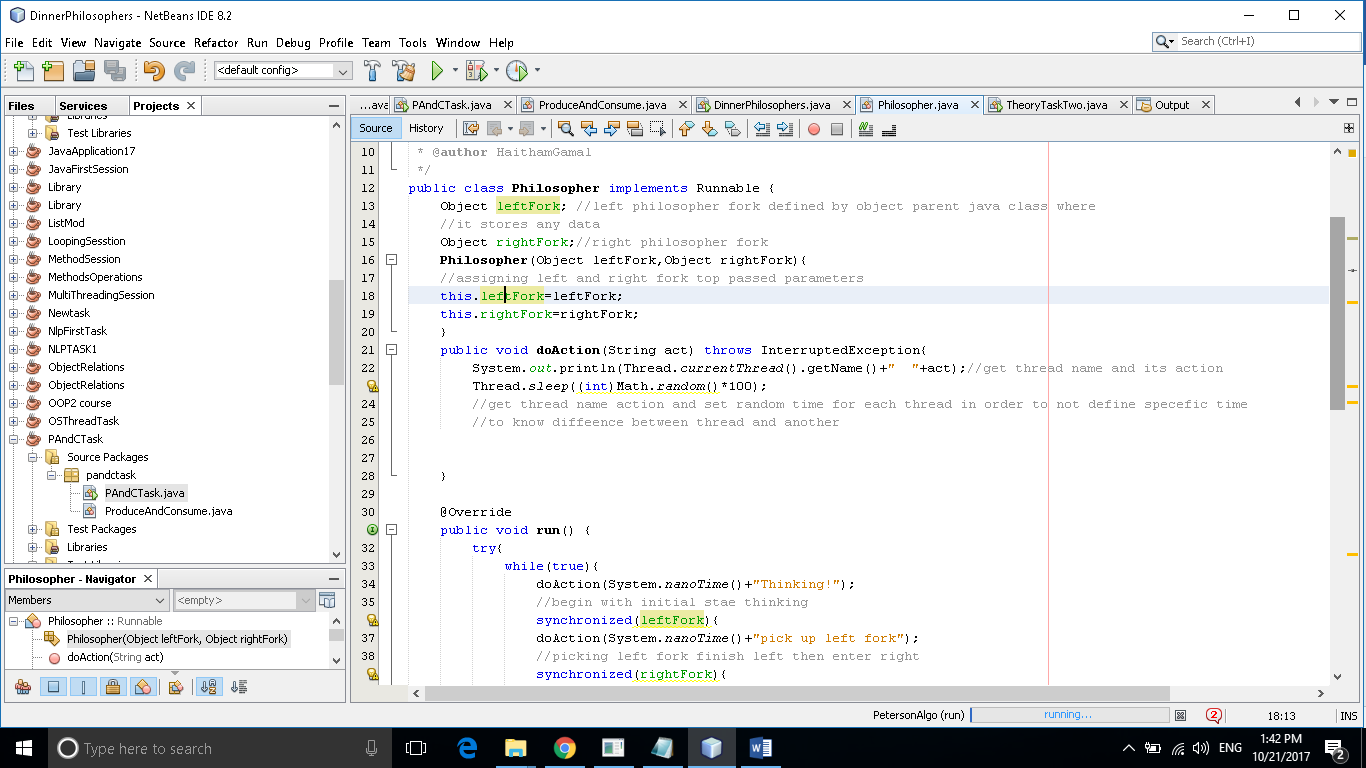
**Solution:**

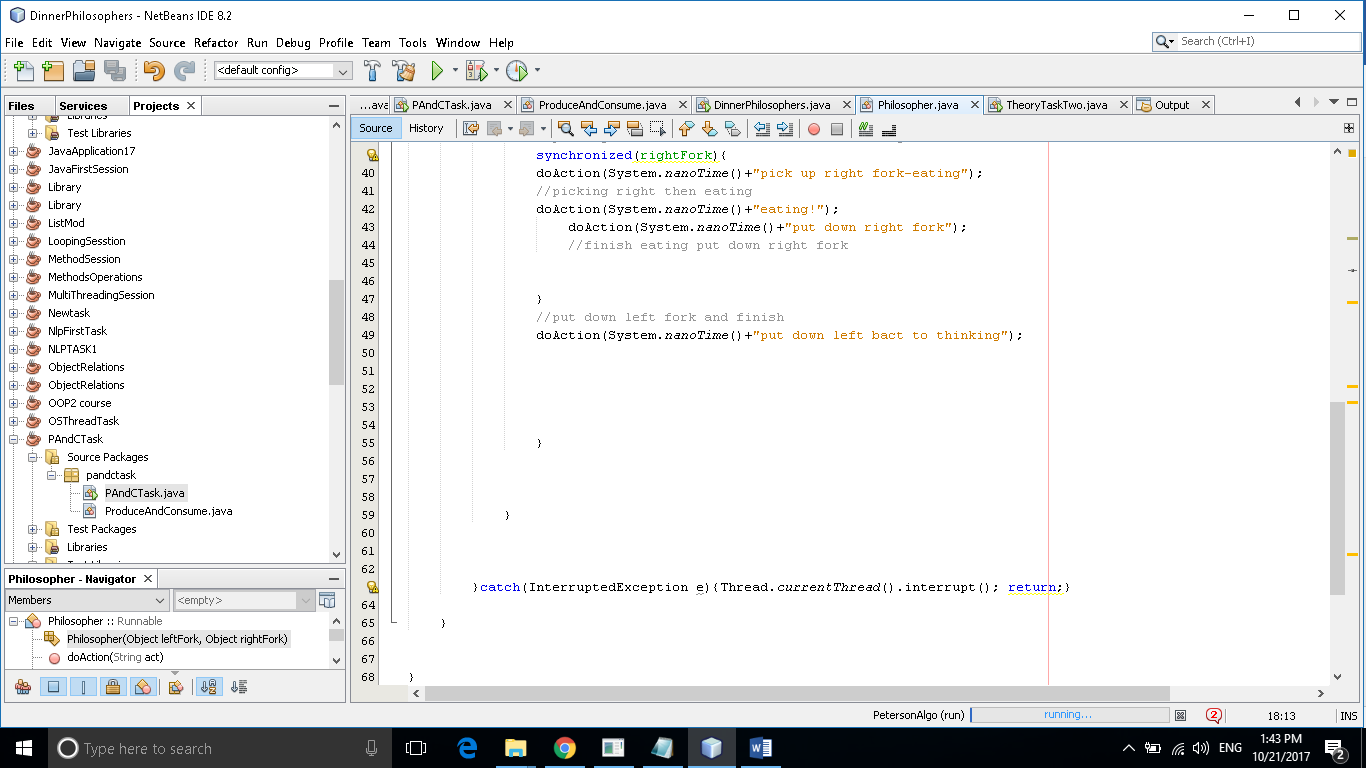
We define 5 objects of philosopher class carry two forks left and right each iteration thread object created and doing action according to passing forks right or left then we access do Action() method that prints its state after executing each action there is random waiting state for other resource to set free to use it in order to not enter deadlock cycle we made condition in for loop when I iterate reaches size of philosopher array -1 we let the last philosopher use right fork first then left fork while the beside philosopher will use left fork that is appeared right for the last so when last access right fork first it blocked to beside and cycle will be broke.

**Figure(1)**

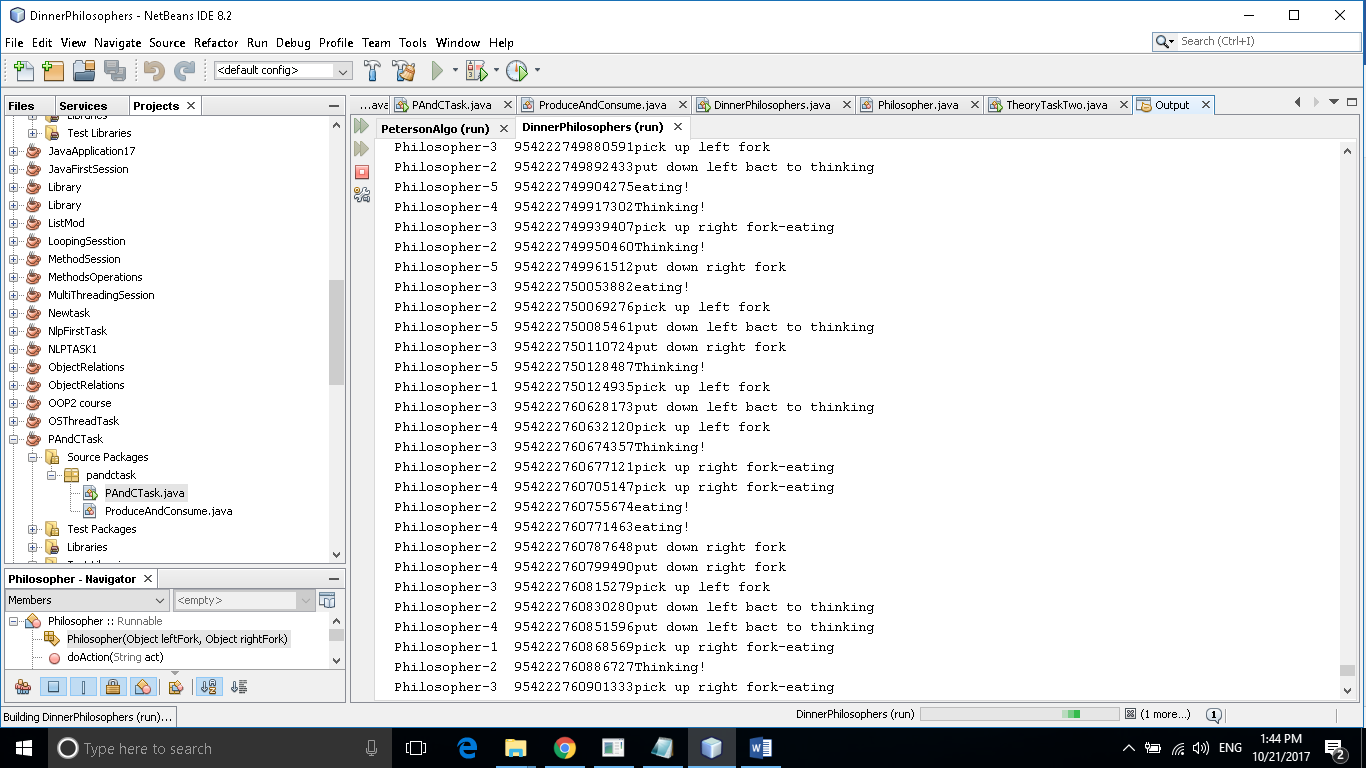
**Figure(2)**

**Figure(4)**



**Figure(5)**

Figure(6)



References:

http://www.baeldung.com/java-dining-philoshophers