CS 307 – HW2 – Dining Philosophers Eylül ONAR 25320

In this homework, it is asked to implement a solution for a modified version of dining philosophers problem by using Java threads and semaphores and to create a barrier so that philosophers wait for each other to start dining. When they arrive to the table, they will reach to a barrier that will block them to start dining before all philosophers are ready to dine. Each philosopher has a barrier parameter and will be acquired to be locked and will be released when all of them arrive. We have three states eating, thinking and being hungry. For the eating state, we create a function where "full" variable is initially set to 0 that refers to philosophers' not eating yet. If philosopher has not eaten, and in the hungry state, then he should eat. We first check whether his right fork is available or not, then look for the left fork. If both forks are not being used, then the philosopher can go the eating statement. We should call the GUI functions in order to change the colors of the forks and plate that particularly shows in which state the philosopher is and the availability of the forks. After done eating, the forks should be released so that others can have access. For the thinking state, we randomly generate integers for the thinking period as all philosopher will think for a distinct period of time. After a philosopher is in the thinking state, he should pass to the hungry state after a certain amount of time (msec). The plate will turn to red and the state will be changed to hungry. In the running function, we display philosophers' arrival times and put the black-white plate in front of them when they reach to the table. In this part, the barrier will ensure that philosopher who arrives, will have to wait for others to come. When all are ready, the dining will be initiated and thinking and eating state functions will be called.