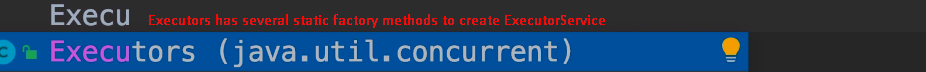
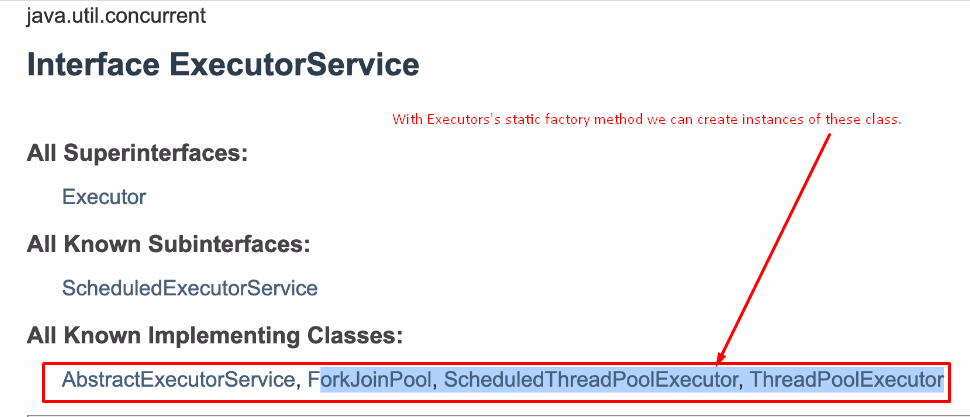
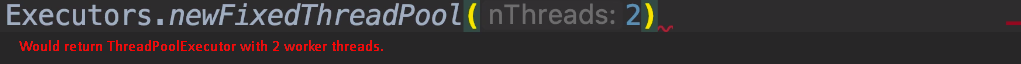
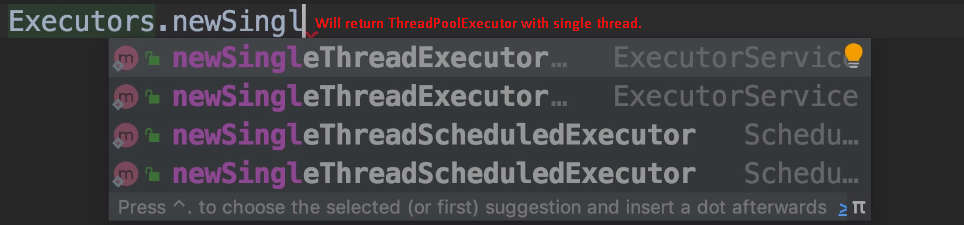
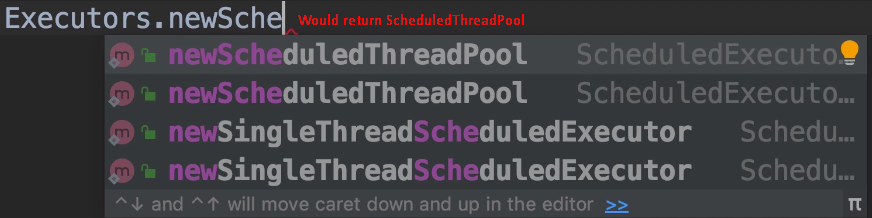
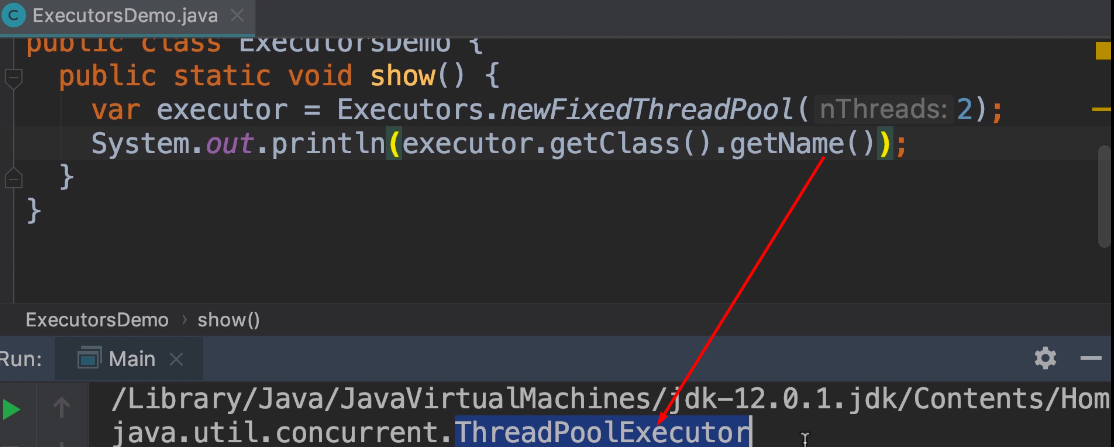
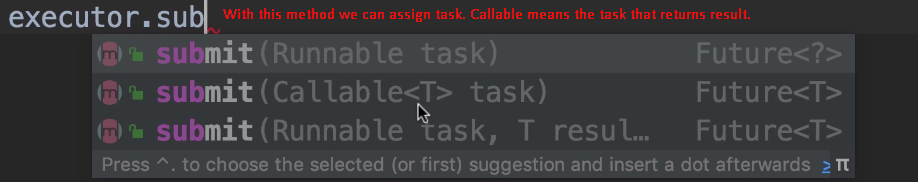
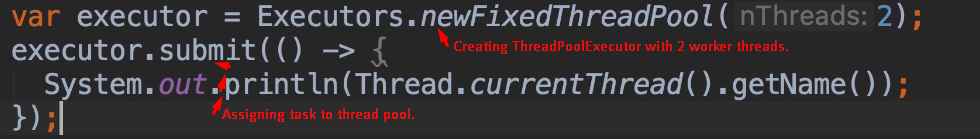
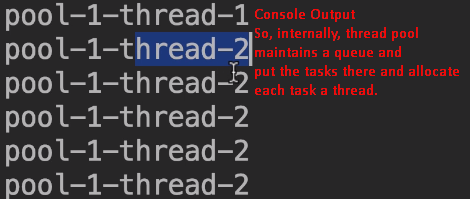
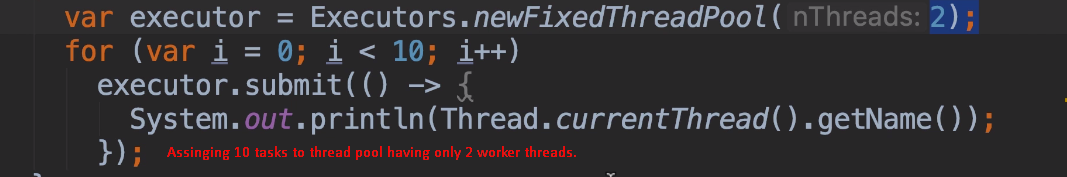
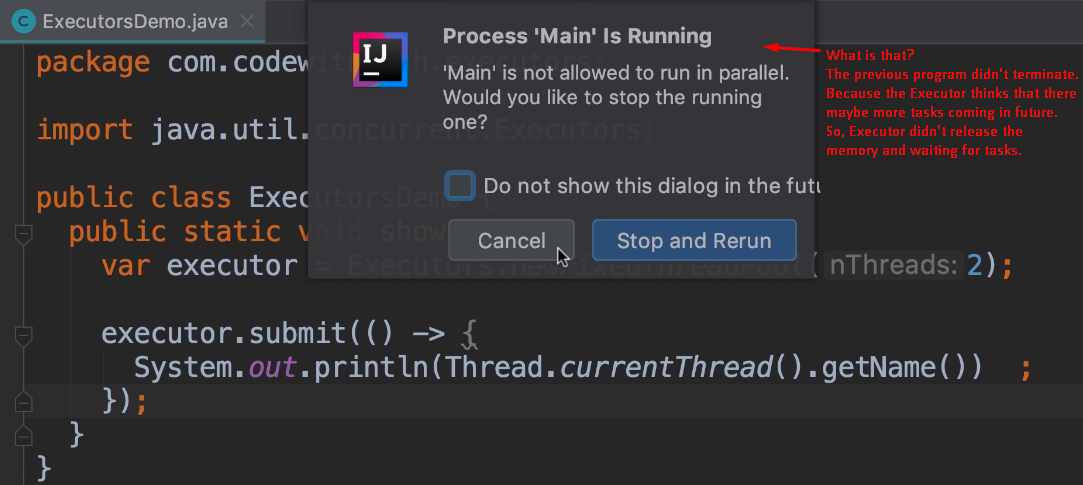
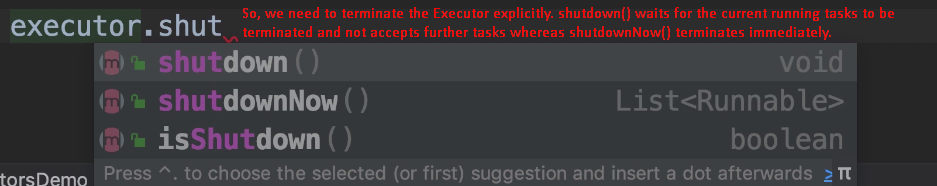
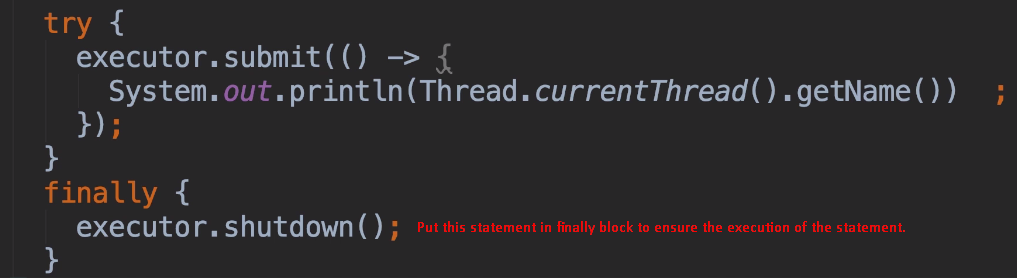
1. 
2. In Java, the concept of thread pool is represented using **ExecutorService** and its implementation.
3. **Implementations for Executor:**
   1. **ThreadPoolExecutor**: Typical implementation for thread pool that we will use most of the time.
   2. **ScheduledThreadPoolExecutor:** With this, we can schedule our tasks after a delay or periodically.
      1. **For** example: we can schedule our task to run 5 hours from now or every 2 hours.
   3. **ForkJoinPool:** Special type of pool designed to **recursively split our task into smaller tasks** and then combine the results from each sub-task to produce the overall result. It is like “Divide and Conquer” Algo.
4. Let’s create **ExecutorService**.
5. 
6. 
7. 
8. The following methods would return ThreadPoolExecutor:  
   
9. The following method returns ScheduledThreadPoolExecutor. 
10. 
11. 
12. 
13. 
14. Let’s remove the for loop in the above program.
15. 
16.   
    Now, we will not get any warning.
17. 
18. **NOTE**: Executor framework doesn’t help us to deal with concurrency problems. It just simplifies thread manipulation.