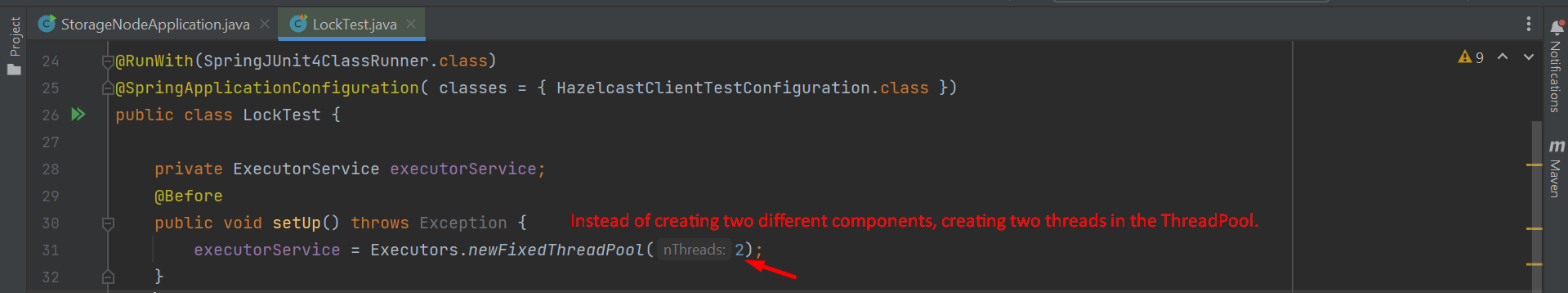
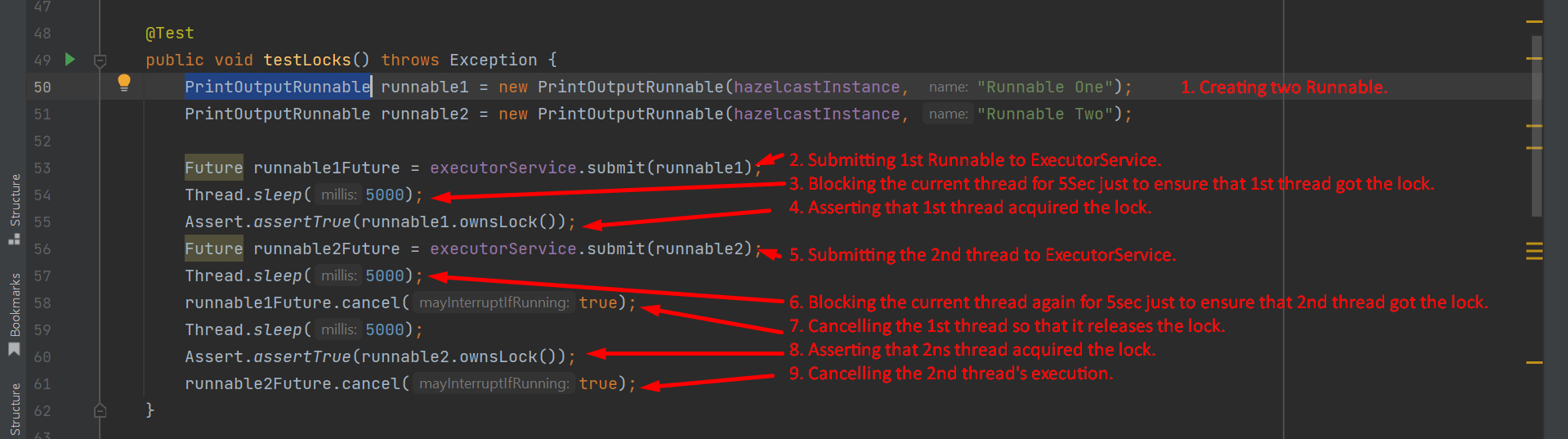
1. **Agenda**:
   1. Lock Data Structure.
2. We already looked at Key Lock in module 3 where we can lock based on key and then we have complete control over the entry for that key in the map.

But what if we’re not working on IMap and we want to have a lock on the data.

1. **Solution**: **com.hazelcast.core.ILock extends** **java.util.concurrent.locks.Lock**
   1. **com.hazelcast.core.ILock** is distributed version of **java.util.concurrent.locks.Lock.** **It allows a single thread in the entire cluster to have the sole control of a guarded piece of code.**
   2. Let’s say you have a process that can only run on a single host at any one time but you want that process to fail over to another host. Should the host currently running that process fail.  
      We can use lock for this purpose.
2. We have written other test case for demonstration purpose.
3. Rather than two different components, we’re going to use two different threads & we’re going to execute these in **ExecutorService** with **two threads** in **thread pool**.  
   
4. Chart

   Description automatically generated with medium confidence
5.   
     
   Text

   Description automatically generated  
   Graphical user interface, text

   Description automatically generated