

Java Multithreading for Senior Engineering Interviews / ... / IllegalMonitorStateException

IllegalMonitorStateException

Learn the reasons that cause IllegalMonitorStateException to be thrown.

If you are interviewing, consider buying our **number#1** course for <u>Java Multithreading</u> <u>Interviews</u>.

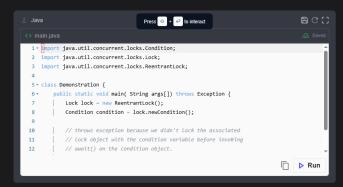
Explanation

The illegalMonitorstateException is a common programming error that can show up in concurrent programs. Depending on the structure of the program, the exception may occur consistently or only occasionally. The illegalMonitorstateException exception class extends the RuntimeException class and according to the official documentation is thrown to indicate that a thread has attempted to wait on an object's monitor or to notify other threads waiting on an object's monitor without owning the specified monitor. In other words if you invoke wait() or notify()/notifyall() without synchronizing on the object i.e. outside of a synchronized method or block (with the object as the synchronization target) then illegalMonitorstateException will be thrown. Similarly, the exception is thrown when you invoke these methods on an instance of condition class without acquiring the associated lock with the condition. The class is part of the java.lang package and not java.util.concurrent.*

Repro using Lock

The program below demonstrates generating IllegalMonitorStateException using a $\pmb{Repro\ using\ Lock}$

The program below demonstrates generating IllegalMonitorStateException using a condition object that was instantiated from a Lock object.



The fix for the above program appears below:

Repro using object

The program in the widget below causes <code>illegalMonitorStateException</code> by invoking <code>notifyall()</code> on an object without synchronizing on it.

The fix for the above program appears below:

| Object myObject - now Object();
| synchronized (myObject) {
| // invoking notifyAll(); |
| myObject.notifyAll();
| }

| Technical Quiz

| Consider the program below and state the outcome of executing the main() method.

| public class IllegalMonitorStateQuiz {
| synchronized void someFunction() throws InterruptedException {
| this.wait();
| }
| public static void main(String[] args) throws Exception {
| (new IllegalMonitorStateQuiz()).someFunction();
| }

| A | EllegalMonitorStateQuiz()).someFunction();
| C. Program is either blocked on wilt() forever or exits because of a spurious wakeup.
| D. The Larget of synchronizes and the object invoking wilt() are different and the program doesn't compile.

Submit Answer

← Back lesson

Phaser

| Wark As Completed | Next → TimeoufException