es/74tdof371b802063647a4ac2c06cd93b86474" class="notecardImageClass" d="74tdof371b802063647a4ac2c06cd93b86474"></div>
</div><div>
</div></div><div>
</div>/div><div>
</div></div> </div> </div> </div>
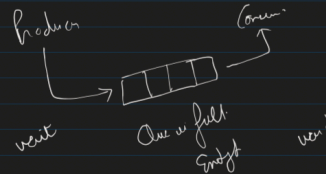
Download as PDF

Java:Thread Joining, Daemon Thread, Thread Priority
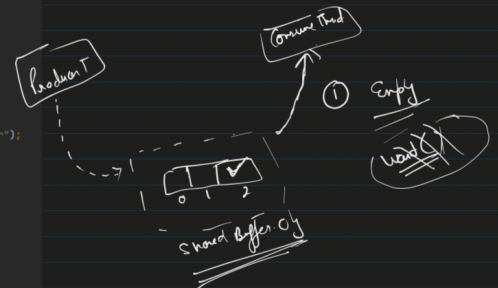
"Concept && Coding" YT Video Notes

wait() → relese Lock
notify() sleep()

TERMINAL

Why Stop, Resume, Suspended method is deprecated?

STOP : Terminates the thread abruptly, No lock release, No resource clean up happens.

SUSPEND: Put the Thread on hold (suspend) for temporarily. No lock is release too.

RESUME: Used to Resume the execution of Suspended thread.

Both this operation could led to issues like deadlock.

lets see an example of it

T1 → Lock R1
T2 → waits for R1
Suspend on STop

T1 → Lock R1
T2 → waits for R1
T1 → suspend

Main	Th1	Th2
t1: Started	Created	Created
t2: Th1 & T2 Created	Acquired Lock R1	Sleep (1000)
t3: Start Th1 & T2		To do Acquired lock R1

T4: | Suspended | Th2 waits for Lock to relese
T5 Suspend Th1 | R1 | Lock! :- Th1

JOIN:

- When JOIN method is invoked on a thread object. Current thread will be blocked and waits for the specific thread to finish.

- It is helpful when we want to coordinate between threads or to ensure we complete certain task before moving ahead.

Main

Create start T1

T2

T3

level
T1 : 5
T2 : 4
T3 10
T4
10

Join for
T2 after

T3
T1
T2
T4

THREAD PRIORITY:

- Priorities are integer ranging from 1 to 10.

1 -> low priority
10 -> highest priority

- Even we set the thread priority while creation, its not gurantted to follow any specific order, its just a hint to thread scheduler which to execute next. (but its not strict rule)

- When new thread is created, it inherit the priority of its Parent thread.

- we can set custom priority using *"setPriority(int priority)"* method

Main (5)
└ T1 (10)

DAEMON THREAD:

X on this which Runs
ASYNC

Main thread (master thread)