Lock Hold Count

The hold count of a lock counts the number of times that a single thread, the owner of the lock, has acquired the lock.

- When a thread acquires a lock for the first time, the lock's hold count is set to one.
- If a lock is re-entrant, and a thread, reacquires the same lock, the lock's hold count will get incremented.
- When a thread releases a lock, the lock's hold count is decremented.
- **A lock is only released when it's hold count becomes zero**.

Because of this, it's really important to include a call to the unlock method in a finally clause, of any code that will acquire a lock, even if it's re-entrant.

Review of the Advantages of using Lock implementations

- **Explicit Control** over when to acquire and release locks, making it easier to avoid deadlocks, and manage other concurrency challenges.
- **Timeouts** allow you to attempt to acquire a lock without blocking indefinitely.
- Along with timeouts, **Interruptible Locking** lets you handle interruptions during acquisition more gracefully.
- **Improved Debugging** methods let you query the number of waiting threads, and check if a thread holds a lock.