

Inter-thread communication in Java

By
M. BABY ANUSHA,
ASST.PROF IN CSE DEPT.,
RGUKT,NUZVID

Introduction

- **Inter-thread communication** or **Co-operation** is all about allowing synchronized threads to communicate with each other.
- Cooperation (Inter-thread communication) is a mechanism in which a thread is paused running in its critical section and another thread is allowed to enter (or lock) in the same critical section to be executed.
- It is implemented by following methods of **Object class**:
 - wait()
 - notify()
 - notifyAll()

Inter-thread communication

1) wait() method

- Causes current thread to release the lock and wait until either another thread invokes the notify() method or the notifyAll() method for this object, or a specified amount of time has elapsed.
- The current thread must own this object's monitor, so it must be called from the synchronized method only otherwise it will throw exception.

Inter-thread communication

Syntax:

```
public final void wait()throws InterruptedException
```

- waits until object is notified.

```
public final void wait(long timeout)throws InterruptedException
```

- waits for the specified amount of time.

Inter-thread communication

2) notify() method

- Wakes up a single thread that is waiting on this object's monitor. If any threads are waiting on this object, one of them is chosen to be awakened. The choice is arbitrary and occurs at the discretion of the implementation.

Syntax:

```
public final void notify()
```

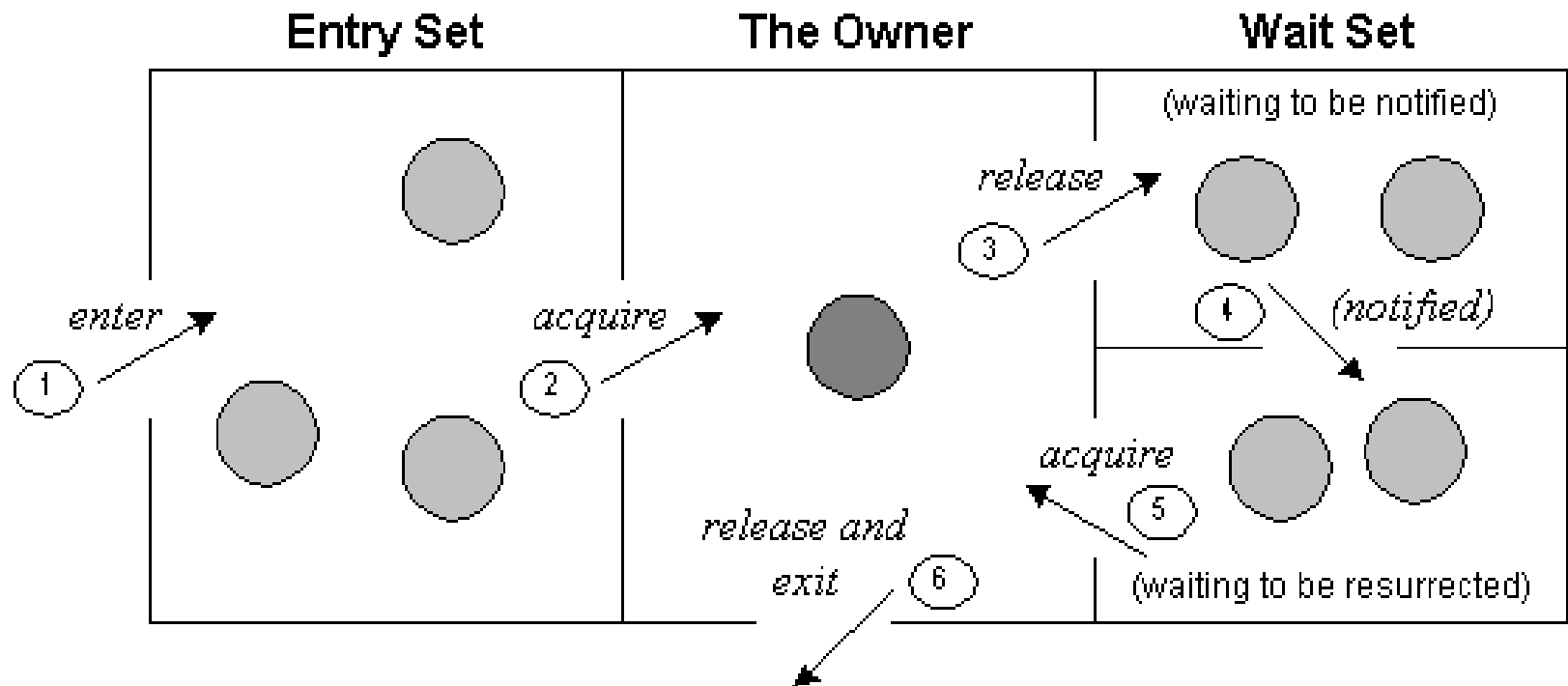
3) notifyAll() method

- Wakes up all threads that are waiting on this object's monitor.

Syntax:

```
public final void notifyAll()
```

Understanding the process of inter-thread communication



Understanding the process of inter-thread communication

The point to point explanation of the above diagram is as follows:

- Threads enter to acquire lock.
 - Lock is acquired by one thread.
 - Now thread goes to waiting state if you call `wait()` method on the object.
- Otherwise it releases the lock and exits.

Understanding the process of inter-thread communication

- If you call `notify()` or `notifyAll()` method, thread moves to the notified state (runnable state).
- Now thread is available to acquire lock.
- After completion of the task, thread releases the lock and exits the monitor state of the object.

Example of inter thread communication in java

```
class Customer
{
    int amount=10000;

    synchronized void withdraw(int amount)
    {
        System.out.println("going to withdraw...");

        if(this.amount<amount){
            System.out.println("Less balance; waiting for deposit...");
            try
            {
                wait();
            }
            catch(Exception e){}
        }
    }
}
```

Example of inter thread communication in java

```
this.amount-=amount;
```

```
System.out.println("withdraw completed...");  
}
```

```
synchronized void deposit(int amount)  
{  
    System.out.println("going to deposit...");  
    this.amount+=amount;  
    System.out.println("deposit completed... ");  
    notify();  
}  
}
```

Example of inter thread communication in java

```
class Test
{
    public static void main(String args[]){
        final Customer c=new Customer();
        new Thread(){
            public void run(){c.withdraw(15000);}
        }.start();
        new Thread(){
            public void run(){c.deposit(10000);}
        }.start();
    }
}
```

Example of inter thread communication in java

Output:

going to withdraw...

Less balance; waiting for deposit...

going to deposit...

deposit completed...

withdraw completed



Thank you!

shutterstock.com · 567687052