Task 2: States and Transitions

Create a Java class that simulates a thread going through different lifecycle states: NEW, RUNNABLE, WAITING, TIMED_WAITING, BLOCKED, and TERMINATED. Use methods like sleep(), wait(), notify(), and join() to demonstrate these states..

```
package com.wipro.model;
public class ThreadLifecycleDemo {
public static void main(String[] args) throws InterruptedException {
Thread thread = new Thread(new Runnable() {
@Override
public void run() {
System.out.println("Thread is in runnable state.");
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
e.printStackTrace();
}//catch
synchronized (this) {
try {
System.out.println("Thread is in waiting state.");
wait();
System.out.println("Thread has been notified and is active again.");
} catch (InterruptedException e) {
e.printStackTrace();
}
}//synchronized
try {
System.out.println("Thread is in TIMED_WAITING state for 2 seconds.");
Thread.sleep(2000); // Thread will sleep for 2 seconds
} catch (InterruptedException e) {
e.printStackTrace();
}
System.out.println("Thread is trying to enter synchronized block in BLOCKED state.");
synchronized (ThreadLifecycleDemo.class) {
System.out.println("Thread entered synchronized block and is no longer BLOCKED.");
}
System.out.println("Thread is terminating...");
}//run
});
thread.start();
Thread.sleep(500);
synchronized (thread) {
System.out.println("Main thread notifying the waiting thread.");
thread.notify();
}
thread.join();
System.out.println("Thread has terminated completely.");
```

```
}
}
```