*//*

[*https://raw.githubusercontent.com/vsaravanan/java22/master/src/main/java/com/saravanjs/java22/console/multithreading/WaitNotifyTest.java*](https://raw.githubusercontent.com/vsaravanan/java22/master/src/main/java/com/saravanjs/java22/console/multithreading/WaitNotifyTest.java)

[*https://www.digitalocean.com/community/tutorials/java-thread-wait-notify-and-notifyall-example*](https://www.digitalocean.com/community/tutorials/java-thread-wait-notify-and-notifyall-example)

 *class* Message {
 *private* String msg;

 *public* Message(String str){
 *this*.msg=str;
 }

 *public* String getMsg() {
 *return* msg;
 }

 *public void* setMsg(String str) {
 *this*.msg=str;
 }

}

*class* Waiter *implements Runnable*{
 *private static int i* = 0;

 *private* Message msg;

 *public* Waiter(Message m){
 *this*.msg=m;
 }

 @Override
 *public void* run() {
 String name = Thread.*currentThread*().getName() + ++*i*;
 *synchronized* (msg) {
 *try*{
 System.*out*.println(name+" waiting to get notified ");
 msg.wait();
 }*catch*(InterruptedException e){
 e.printStackTrace();
 }
 System.*out*.println(name+" waiter thread got notified ");
 *//process the message now* System.*out*.println(name+" processed: "+msg.getMsg());
 }
 }

}

*class* Notifier *implements Runnable* {

 *private* Message msg;

 *public* Notifier(Message msg) {
 *this*.msg = msg;
 }

 @Override
 *public void* run() {
 String name = Thread.*currentThread*().getName();
 System.*out*.println(name+" started");
 *try* {
 Thread.*sleep*(1000);
 System.*out*.println("Resumed!");
 *synchronized* (msg) {
 msg.setMsg(name+" Notifier work done");
 msg.notify();
*// msg.notifyAll();* }
 } *catch* (InterruptedException e) {
 e.printStackTrace();
 }

 }

}

*public class* WaitNotifyTest {
 *public static void* main(String[] args) {
 Message msg = *new* Message("process it");

 Scanner scanner = *new* Scanner(System.*in*);
 *while* (*true*) {
 System.*out*.println("Enter action (w=Wait, n=Notify, q=Quit):");
 String input = scanner.nextLine().trim().toLowerCase();

 *switch* (input) {
 *case* "w":
 System.*out*.println("Waiting...");
 Waiter waiter = *new* Waiter(msg);
 Thread waiterThread = *new* Thread(waiter, "waiter");
 waiterThread.start();
 *break*;
 *case* "n":
 System.*out*.println("Notifying...");
 Notifier notifier = *new* Notifier(msg);
 Thread notifierThread = *new* Thread(notifier, "notifier");
 notifierThread.start();

*// synchronized (msg) {
// msg.notify();
// }
 break*;
 *case* "q":
 System.*out*.println("Quitting...");
 scanner.close();
 System.*exit*(0);
 *break*;
 *default*:
 System.*out*.println("Invalid input!");
 }
 }
 }
}

Enter action (w=Wait, n=Notify, q=Quit):

w

Waiting...

Enter action (w=Wait, n=Notify, q=Quit):

waiter1 waiting to get notified

w

Waiting...

Enter action (w=Wait, n=Notify, q=Quit):

waiter2 waiting to get notified

w

Waiting...

Enter action (w=Wait, n=Notify, q=Quit):

waiter3 waiting to get notified

n

Notifying...

Enter action (w=Wait, n=Notify, q=Quit):

notifier started

Resumed!

waiter1 waiter thread got notified

waiter1 processed: notifier Notifier work done

n

Notifying...

Enter action (w=Wait, n=Notify, q=Quit):

notifier started

Resumed!

waiter2 waiter thread got notified

waiter2 processed: notifier Notifier work done

n

Notifying...

Enter action (w=Wait, n=Notify, q=Quit):

notifier started

Resumed!

waiter3 waiter thread got notified

waiter3 processed: notifier Notifier work done

n

Notifying...

Enter action (w=Wait, n=Notify, q=Quit):

notifier started

Resumed!