

```

public class WaitNotifyExample {
    public static void main(String[] args) {
        WorkObject workObject = new WorkObject();
        ThreadA threadA = new ThreadA(workObject);
        ThreadB threadB = new ThreadB(workObject);
        threadA.start();
        threadB.start();
    }
}

```

공유 작업 객체 생성

작업 스레드 생성 및 실행

```

public class WorkObject {
    public synchronized void methodA() {
        Thread thread = Thread.currentThread();
        System.out.println(thread.getName() + ": methodA 작업 실행");
        notify();
        try {
            wait();
        } catch (InterruptedException e) {
        }
    }

    public synchronized void methodB() {
        Thread thread = Thread.currentThread();
        System.out.println(thread.getName() + ": methodB 작업 실행");
        notify();
        try {
            wait();
        } catch (InterruptedException e) {
        }
    }
}

```

동기화 메소드

다른 스레드를 실행 대기 상태로 만들

자신의 스레드는 일시 정지 상태로 만들

동기화 메소드

다른 스레드를 실행 대기 상태로 만들

자신의 스레드는 일시 정지 상태로 만들

```

public class ThreadA extends Thread {
    private WorkObject workObject;

    public ThreadA(WorkObject workObject) {
        setName("ThreadA");
        this.workObject = workObject;
    }

    @Override
    public void run() {
        for(int i=0; i<10; i++) {
            workObject.methodA();
        }
    }
}

```

공유 작업 객체를 받음

스레드 이름 변경

동기화 메소드 호출

```

public class ThreadB extends Thread {
    private WorkObject workObject;

    public ThreadB(WorkObject workObject) {
        setName("ThreadB");
        this.workObject = workObject;
    }

    @Override
    public void run() {
        for(int i=0; i<10; i++) {
            workObject.methodB();
        }
    }
}

```

공유 작업 객체를 받음

스레드 이름 변경

동기화 메소드 호출

실행 결과

```

ThreadA: methodA 작업 실행
ThreadB: methodB 작업 실행
ThreadA: methodA 작업 실행
ThreadB: methodB 작업 실행
...

```