

## バグ退治

ずいぶん前に書いたコードのバグを指摘されたので修正。  
思い出すのにえらく時間がかかってしまったけど、  
バグ自体はクリティカルな割に、簡単な修正で解決。やれやれ。

## 掃除

おろそかになっていた部屋の片付けを。少しすっきり。

## JavaRock 開発日誌

wait-notify ができるようにと、下のようなコードを考えてみるけど ....

```
public class Consumer extends Thread{
    private int counter;
    private synchronized void exec(){
        try{
            wait();
        }catch(InterruptedException e){
            e.printStackTrace();
        }
        System.out.println("consume: " + counter);
        counter++;
    }
    public synchronized void wakeup(){
        notify();
    }
    public void run(){
        while(true){
            exec();
        }
    }
}
```

むー、今の実装では、いろいろと厳しいぞ。

## JavaRock 開発日誌 (2)

ごによごによやって、アドホックな解決策を思いついた  
... と思ったけど、synchronized の扱いが下手だったので、上のコードでは上手くいかない。  
とりあえず下のように、notify は外から叩くことにする。

## コンシューマ側

```
public class Consumer extends Thread{
    private int counter;
    private synchronized void exec(){
        try{
            wait();
        }catch(InterruptedException e){
            e.printStackTrace();
        }
        counter++;
    }
    public void run(){
        while(true){
            exec();
        }
    }
}
```

## プロデューサ側

```
public class Producer extends Thread{
```

```
private final Consumer consumer = new Consumer();
int counter;
public void run(){
    consumer.start();
    while(true){
        counter++;
        synchronized(consumer){
            consumer.notify();
        }
    }
}
```