

Lecteurs – Rédacteurs en Java – Une solution

```
class LecteursRedacteurs {  
  
    private boolean unRedacteurActif = false;  
    private int nbRedacteursEnAttente = 0;  
  
    private int nbLecteursEnAttente = 0;  
    private int nbLecteurs = 0;        // Nombre de lecteurs en train de lire  
  
    public synchronized void JeVeuxLire() {  
        nbLecteursEnAttente++;  
        while (unRedacteurActif or nbRedacteursEnAttente > 0 ) {  
            // Priorité aux rédacteurs  
  
            try {this.wait();} catch (InterruptedException e) {}  
        }  
        nbLecteursEnAttente--;  
        nbLecteurs++; // Un lecteur supplémentaire  
    }  
  
    public synchronized void JaiFinideLire() {  
        nbLecteurs--;  
        if (nbLecteurs == 0) this.notifyAll();  
  
        Pourquoi "all"? --> Il peut y avoir simultanément des lecteurs et des rédacteurs en attente.  
  
        En effet, s'il existe un rédacteur en attente, les lecteurs sont bloqués (en raison de la priorité  
        donnée aux rédacteurs)  
  
        --> Il faut être sûr de réveiller au moins un rédacteur en attente  
  
        Pourquoi le test if (nbLecteurs == 0) ? -> inutile de réveiller un quelconque rédacteur en  
        attente.  
  
        Ce dernier devrait de toute façon attendre que tous les lecteurs aient terminé leur travail.  
  
        Inutile non plus de réveiller des lecteurs en attente. Si c'était le cas, cela voudrait dire  
        qu'un rédacteur prioritaire est en attente.  
    }  
  
    public synchronized void JeVeuxEcrire() {  
        nbRedacteursEnAttente++;  
        while ((nbLecteurs > 0 or unRedacteurActif) ) {  
            try {this.wait();} catch (InterruptedException e) {}  
  
            Pourquoi if (nbLecteurs > 0) ? --> Tant qu'il y a des lecteurs en train de lire, pas le  
            droit d'écrire  
  
        }  
        unRedacteurActif=true;  
        nbRedacteursEnAttente--;  
    }  
  
    public synchronized void JaiFinideEcrire() {  
        unRedacteurActif=false;  
        this.notifyAll();  
        Idem que pour "JaiFinideLire": on réveille tout le monde.. Rédacteurs et lecteurs..  
        De toute façon, grâce au "while", tous les threads réveillés testent à nouveau la condition  
    }  
}
```

