

## Corrigé Quiz 2

### Exercice 1 : Lecteurs-rédacteurs alternés

L'algorithme 1 permet la lecture et l'écriture avec lecteurs et rédacteurs alternés.

### Exercice 2 : Sémantique des moniteurs

- stoppé-entrée (Portal)  $\longrightarrow$  queue FIFO
- stoppé-wait (Portal)  $\longrightarrow$  queue FIFO
- stoppé-send (Portal)  $\longrightarrow$  queue LIFO
- stoppé-entrée (Java)  $\longrightarrow$  ensemble
- stoppé-wait (Java)  $\longrightarrow$  ensemble

**Figure 1** Lecteurs-rédacteurs alternés

---

```

monitor Alternés;
  defines debutLire, finLire, debutEcrire, finEcrire;
  signal lireOK, ecrireOK;

  var
    ecriture : boolean  $\leftarrow$  false                                {indique si une écriture est en cours}
    nbLecteurs : integer  $\leftarrow$  0                                {indique combien de lecteurs sont en train de lire}

  procedure debutLire
    if ecriture or ecrireOK.awaited then                            {attendre si écriture en cours ou en attente}
      lireOK.wait
    end if
    nbLecteurs  $\leftarrow$  nbLecteurs + 1
    if not ecrireOK.awaited then                                    {réveiller autres lecteurs si possible}
      lireOK.send
    end if
  end

  procedure finLire
    nbLecteurs  $\leftarrow$  nbLecteurs - 1
    if nbLecteurs = 0 and ecrireOK.awaited then
      ecrireOK.send                                {réveiller un rédacteur si pas de lecture en cours}
    end if
  end

  procedure debutEcrire
    if nbLecteurs > 0 or ecriture then                            {attendre si pas d'accès exclusif}
      ecrireOK.wait
    end if
    ecriture  $\leftarrow$  true
  end

  procedure finEcrire
    ecriture  $\leftarrow$  false
    if lireOK.awaited then                                    {alternance: réveiller lecteur si possible ...}
      lireOK.send
    else
      ecrireOK.send                                {... sinon, réveiller prochain rédacteur}
    end if
  end
end

```

---