

Concurrent Programming

Exercice

Reader/Writer

Le but de cet exercice est de réaliser une hiérarchie de classes permettant de résoudre le problème des *lecteurs/rédacteurs* avec plusieurs variantes pour gérer la priorité d'accès aux données. Les classes *SharedData* (qui implémente les données), *SomeWriterThread* (qui implémente les rédacteurs) et *SomeReaderThread* (qui implémente les lecteurs) vous sont données.

Il s'agit donc d'écrire une classe *ReaderPriority* (puis *WriterPriority*) gérant l'accès aux données. La classe *ReaderPriority* devrait donner la priorité aux lecteurs lorsqu'à la fois des lecteurs et des rédacteurs sont en attente. Dans la même situation, la classe *WriterPriority* devrait donner la priorité aux rédacteurs.

Pour ce faire, chacune de ces classes devra implémenter l'interface *ReaderWriter* suivante :

```
public interface ReaderWriter {  
    public void startRead();  
    public void endRead();  
    public void startWrite();  
    public void endWrite();  
}
```

Les classes *SomeReaderThread* et *SomeWriterThread* devront appeler les méthodes *startRead* et *endRead* (resp. *startWrite* et *endWrite*) pour pouvoir accéder aux données.