

TP1
Producteur / Consommateur

Vous utiliserez pour ce Tp le package `java.util.concurrent`

Un thread producteur crée des messages (e.g. la date courante) et les place dans une file d'attente (e.g. un `Vector`). Un thread consommateur retire un message de la file d'attente et l'affiche. On impose:

- que la file d'attente soit bornée: lorsque la file d'attente est pleine le producteur se met en attente et lorsqu'elle est vide le consommateur se met en attente.
 - que l'on puisse imposer un débit différent au producteur et au consommateur (chaque thread s'endormira (`sleep`) entre chaque itération)
1. En utilisant les méthodes sur les objets(`synchronized`, `wait`...), écrivez les classes `FileAttente`, `Producteur` et `Consommateur` ainsi qu'une classe `Main` qui permet de tester ces classes. Faites les modifications nécessaires pour que l'on puisse lancer plusieurs consommateurs simultanément.
 2. Réalisez une deuxième implémentation en utilisant des sémaphores.
 3. On définit l'interface `verrou` utilisable par deux processus `pid=0` et `pid=1`:

```
public interface verrou{
public void prise{int pid}
public void relache( int pid)
}
```

- En utilisant l'algorithme de Peterson réaliser une implémentation de l'interface `verrou`
- Donnez une solution du problème du producteur consommateur en utilisant un verrou.