

Systeme TP2

1. A.

Un rédacteur empêche les lecteurs d'accéder à sa ressource alors qu'en pâtisserie, le pâtissier n'attend pas de finir ces gâteaux pour les vendre. Un client peut prendre un gâteau déjà fait et mis en file d'attente pendant qu'un pâtissier fait un nouveau gâteau. Si le pâtissier est rédacteur, le client devra attendre qu'il ait fini son gâteau pour en prendre un.

B.

Ici un pâtissier et un client doivent partager la ressource Gâteau. Il faut gérer une exclusion mutuelle pour qu'un pâtissier n'ajoute pas de gâteau lorsqu'un client en prend un et inversement. L'accès à la liste est conflictuel.

Pâtisserie fera moniteur sur ces méthodes d'ajout et de retrait de gâteaux.

Un thread est défini par sa méthode `run()` et est lancé par la méthode `start()`;

2.

L'exclusion mutuelle est gérée par les méthodes `put()` et `take()` de `l'ArrayBlockingQueue`.

Ces méthodes gèrent elles même le `wait()` et le `notify()`, on a donc plus besoin de `synchronized()` et donc de moniteur

3. A.

Un pâtissier n'a pas a continuer de faire des gâteaux si il n'y a plus de clients.

C.

Un des problèmes est que l'on tue des clients.

De plus, des inégalités apparaissent, des clients peuvent prendre plus de gâteaux que d'autres.