

## **Thread Java TD/TP N°2**

### ***Illustration d'un modèle producteur consommateur simple***

#### **Exercice I**

Cet exercice cherche à simuler la gestion d'une boîte à lettres (BAL). Une tâche, "le producteur" crée une lettre et dépose celle-ci dans la BAL. Parallèlement, une autre tâche consulte la BAL de façon asynchrone et retire la "lettre" qu'elle contient puis l'affiche à l'écran; les deux tâches s'ignorant complètement.

Le principe d'une BAL est défini par les contraintes suivantes:

- le producteur ne dépose sa "lettre" dans la BAL que si elle est vide.
- le consommateur ne retire une "lettre" de la BAL que si elle en contient une.

**I1** - Proposer une solution logicielle à cette simulation, cette solution devra comporter 2 tâches: un producteur et un consommateur. Une classe BAL propose les services DEPOSER et RETIRER.

**I2** - Modifier la tâche "le producteur" afin qu'elle saisisse la "lettre" au clavier. Protocole entre les deux tâches: lorsque la lettre est 'Q', les deux tâches décident de s'arrêter.

ne pas faire **I3** – Définir une interface graphique pour les parties **I1** et **I2** de cet exercice.

#### **Exercice II**

### ***Généralisation de la BAL: tampon à N lettres***

On suppose maintenant que la BAL peut contenir N lettres. Les deux tâches (le producteur et le consommateur) sont inchangées dans l'esprit par rapport à l'exercice précédent. Le producteur doit générer dans la BAL des lettres de 'A' jusqu'à 'Z' et finir par le caractère spécial '\*'. Pendant ce temps, de façon un peu moins rapide, le consommateur lit les lettres dans la BAL. Il doit donc apparaître rapidement la saturation de la boîte (le tampon).

Proposer une solution logicielle au problème décrit ci dessus, le tampon aura une structure de tableau où les lettres disponibles seront repérées par deux indices "tete" et "queue". Le nombre de ces lettres contenu dans une variable "nbLettres" ne devra donc pas dépasser la longueur du tableau.