

Travaux Pratique Java – Multithreading et Synchronisation

Titre : Système de production/consommation avec panier limité et plusieurs threads

Contexte

Vous allez simuler un système où :

- **3 producteurs** fabriquent des fruits.
- **4 consommateurs** mangent ces fruits.
- Tous utilisent un **panier commun**, capable de contenir au **maximum 5 fruits**.
- Les producteurs **arrêtent de produire** lorsque le panier atteint cette **limite maximale** (il ne déborde jamais).
- Les consommateurs consomment jusqu'à ce que le panier soit **vide**.

Spécifications

- Le **panier** est **partagé** entre tous les producteurs et consommateurs.
- Un producteur **attend** si le panier est **plein**.
- Un consommateur **attend** si le panier est **vide**.
- Le programme utilise synchronized, wait(), notifyAll().
- Le système utilise sleep() pour simuler la production/consommation.
- Le système s'arrête une fois que tous les fruits sont produits et consommés (par exemple, 15 fruits au total).

Classes proposes

Classe	Description
Panier	Gère le stockage des fruits avec une limite de capacité.
Producteur	Produit des fruits (thread).
Consommateur	Consomme les fruits (thread).
MainTP	Démarre tous les threads.

Exemple d'instructions métier :

Le **panier** a une capacité de **5 fruits**. Chaque producteur doit produire **5 fruits** (soit un total de 15). Les consommateurs mangent les fruits jusqu'à ce que le panier soit vide et que tous les fruits aient été produits et consommés.

Étapes du TP à suivre

1. Implémenter la classe Panier avec synchronisation.
2. Implémenter les classes Producteur et Consommateur avec Thread.
3. Utiliser wait() et notifyAll() dans Panier pour gérer l'attente.
4. Créer 3 producteurs et 4 consommateurs dans la classe MainTP.
5. Lancer tous les threads et gérer correctement l'arrêt du programme.