

## **Exercice 09**

---

### **Un Provider complet (avec file de message)**

<b><u>1.</u></b>	<b><u>ENONCES</u></b>	<b><u>2</u></b>
<b><u>2.</u></b>	<b><u>CORRECTION</u></b>	<b><u>4</u></b>

## 1. Enoncés

Il s'agit de faire évoluer l'exemple : ExempleCh06\_02\_Provider

[http://coursjava.fr/NSY102\\_Exemples.php?repertoire=ExempleCh06\\_02\\_Provider](http://coursjava.fr/NSY102_Exemples.php?repertoire=ExempleCh06_02_Provider)

Le provider de cet exemple ne gère pas de file de message.

Cela veut dire que le mode de communication entre un producteur et ses consommateurs est en mode push-and-push (plus précisément push asynchrone-and-push synchrone).

L'évolution consiste à réaliser aussi le mode push-and-pull :

- le producteur pousse ces événements dans une file d'évènement
- chaque consommateur tire (à son rythme) les événements de la file
- chaque consommateur à son itérateur permettant de tier les événements à son propre rythme

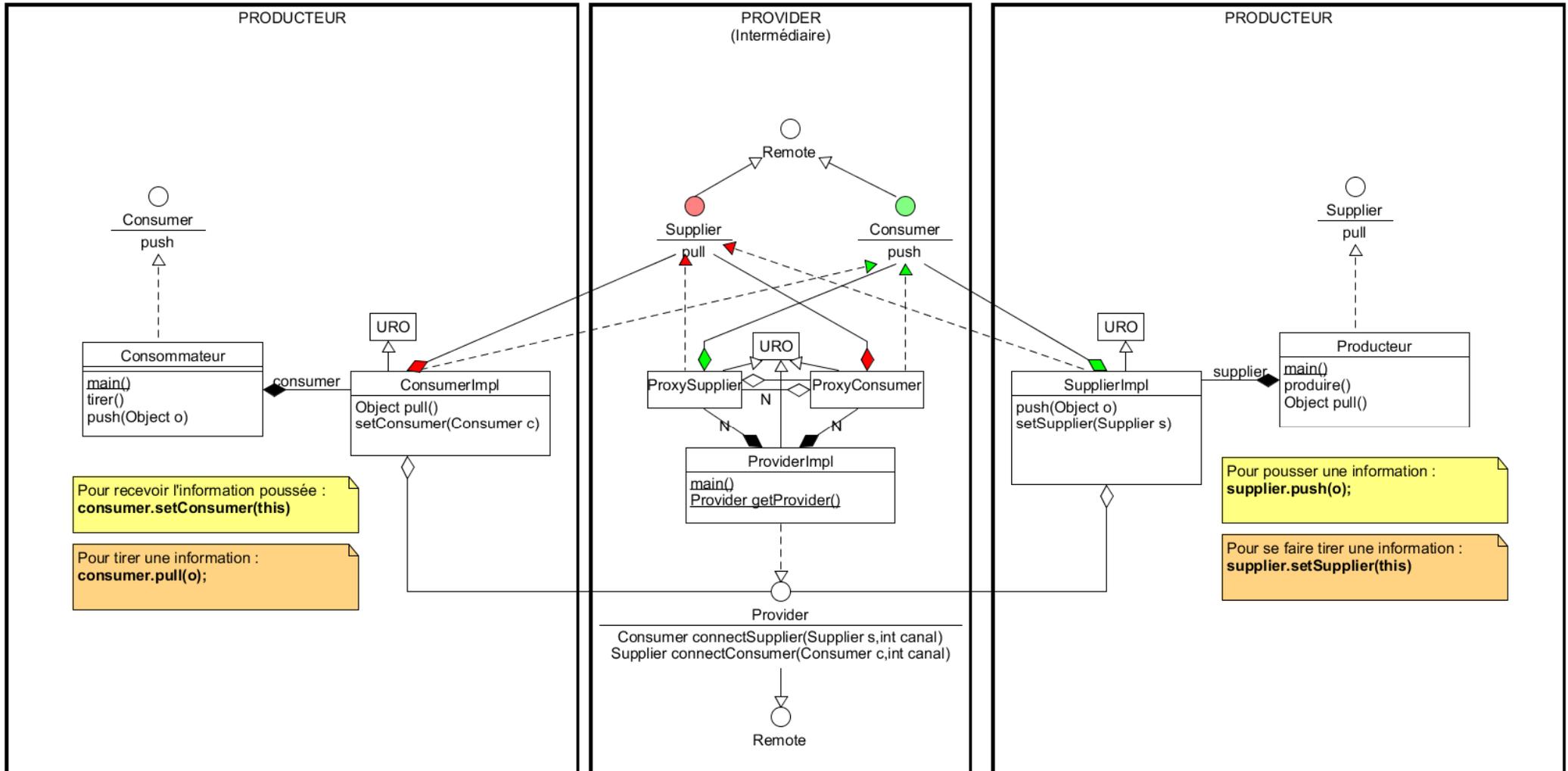
Afin de rendre compatible le mode push-and-push, et le mode push-and-pull, on considère dorénavant que le producteur pousse toujours son événement dans la file.

Si on veut garder le mode pull-and-pull (que permet le provider de l'exemple 06), il faut avoir 2 méthodes différentes de pull utilisé par le consommateur :

- pull-synchrone (pour le mode pull-and-pull) qui ne passe pas par la file d'évènement
- pull-asynchrone (pour le mode push-and-pull) qui passe par la file d'évènement.

L'exercice consiste à modifier le diagramme de classe de l'exemple ExempleCh06\_02\_Provider présentation dans le cours afin d'y injecter cette évolution.

On rappelle ci-dessous ce diagramme de classe :



PUSH-AND-PUSH  
PULL-AND-PULL

## **2. CORRECTION**