

Proxy Pattern

1 Gestion des données transactionnelles d'un e-commerce

Dans un e-commerce, un commerçant enregistre le comportement d'achat avéré de ses clients dans une base de données transactionnelles. Les données enregistrées représentent le panier d'achat des clients (les transactions).

Exemple :

D_1				
Transaction	Articles			
t_1	A	B	C	D
t_2		B	C	E
t_3	A	B	C	E
t_4		B		E

Ici, nous avons un exemple d'une base transactionnelle D_1 d'un e-commerce qui propose 5 articles (de A à E) et les paniers de 4 clients (t_1 à t_4). Nous pouvons lire qu'un client a eu une transaction t_3 où il a acheté les articles $\{A, B, C, E\}$. Un article est décrit par un numéro unique, une description et un prix. Une transaction est composée d'un ensemble d'Articles.

Question 1 • Proposer une conception avec les classes **Article**, **Transaction** et **Dataset**. La classe **Dataset** permet de représenter une base transactionnelle et d'avoir, à l'instant t , une et une seule base en mémoire.

Un motif ensembliste est un sous-ensemble d'articles possible. A , BC , AD , BCE et DE sont des motifs possibles. On souhaite représenter un motif comme étant une spécialisation de la notion de transaction. On parle de fréquence relative d'un motif, noté $freq$, comme étant le nombre d'occurrence du motif dans la base. Un motif est fréquent si et seulement si sa fréquence est supérieure ou égale à un certain seuil θ que nous allons fixer dans ce TD à 50%.

- $freq(A) = 2/|D_1| = 2/4 = 50\% \Rightarrow A$ est fréquent !
- $freq(BC) = 3/4 = 75\% \Rightarrow BC$ est fréquent !
- $freq(AD) = 1/4 = 25\% \Rightarrow AD$ n'est pas fréquent !
- $freq(BCE) = 2/4 = 50\% \Rightarrow BCE$ est fréquent !
- $freq(DE) = 0/4 = 0\% \Rightarrow DE$ n'est pas fréquent !

Question 2 • Ajouter à votre conception la classe **Motif** qui implémente la méthode `isFreq() : boolean`.

Question 3 • La fonction `isFreq()` fait appel à la méthode privée `freq():float`. Donner le code des deux fonctions.

Étant donnée notre base transactionnelle D_1 , nous souhaitons connaître si les motifs de la liste suivante sont fréquents ou pas :

$$L = \langle BE, B, E, ABC, AB, AC, BC, A, C, D, AD, BD, CD, BCD, ACD, ABCD \rangle$$

Question 4 • Classer les motifs comme fréquent/non-fréquent. Combien de fois la fonction privée `freq` est-elle appelée ?

Question 5 • Quelle est l'observation et les propriétés à retenir de :

- BE et $\{B, E\}$
- ABC et $\{AB, AC, BC, A, B, C\}$
- D et $\{AD, BD, CD, BCD, ACD, ABCD\}$

1.1 Le Proxy-Cache

Afin d'éviter le calcul inutile de la fréquence des motifs dominés, vous allez mettre en place un proxy-cache.

Question 6 • Réviser votre conception en ajoutant un proxy à la notion de `Motif` avec l'interface `IMotif` et le proxy `ProxyMotif`. Afin que le test `isFreq` utilise le proxy.

Question 7 • Donner le code de la fonction `isFreq` du proxy.

Dans le dossier `data`, nous vous avons fourni un exemple d'une base transactionnelle `ecommerce.txt` (129 articles et 68 000 transactions) et un fichier de test `motifs.txt` (346 962 motifs). Dans `ecommerce.txt`, chaque ligne représente une transaction et chaque numéro dans la transaction représente le numéro d'un article.

Question 8 • Implanter l'ensemble des classes et des méthodes sans l'utilisation du proxy.

Question 9 • Tester votre réalisation sur le fichier `motifs.txt` en affichant à la fin le nombre de motifs fréquent. Mesurez le temps de calcul en utilisant `System.currentTimeMillis()` qui retourne le nombre de millisecondes écoulées depuis EPOCH (01/01/1970 à 00H00). Mesurer le temps de calcul nécessaire pour tester l'ensemble des motifs.

Question 10 • Implanter enfin le proxy et renouveler les tests et les mesures de temps. Donner vos conclusions.