



Licence Informatique Semestre 5 Conception Programmation Objet Avancée

Objectif : comparer les DP Interpreteur et Visiteur sur un exemple simple.

Un extrait de l'application **Calculette** vous est fourni : les classes **Expression**, **Binaire**, **Somme** et **Nombre**, ainsi qu'une fonction **main** rudimentaire qui crée une expression et calcule sa valeur. Ces classes sont très pauvres et nous allons les compléter petit à petit, en cherchant à chaque étape à prendre du recul par rapport au code pour estimer la qualité de ce que l'on fait et le temps que cela prend. L'objectif n'est pas de programmer, mais de comprendre et de comparer les deux patterns. Pour cela, il est indispensable de répondre à chaque question dans l'ordre et ne pas chercher à écrire du code qui n'est pas demandé.

1. Quels DP sont appliqués dans le code fourni ?
2. Créez un nouvel projet IntelliJ avec deux répertoires **interpreteur** et **visiteur**. Placez une copie de toutes les classes dans les deux répertoires. Vérifiez que l'exécution des deux **Main** se passe sans problème (sans oublier de mettre l'option **-ea** dans les options de la machine virtuelle, dans les configurations d'exécution).
3. Dans le package **interpreteur**, complétez les classes **Somme** et **Nombre** avec la définition de la fonction **toString** (notation complètement parenthésée) ; complétez la fonction **main** pour tester votre code.
4. Complétez le package **visiteur** avec une classe abstraite **Visiteur** et une sous-classe **VisitToString**. Ce visiteur est capable de visiter **Nombre** et **Somme**. Complétez l'interface **Expression** avec la fonction **accept**, implémentée dans chaque sous-classe. Complétez la fonction **main** pour créer un **VisitToString** pour tester votre code.
5. Un moment de pause et de réflexion : êtes-vous sûr d'avoir assimilé la différence entre les deux solutions ? quelle solution avez-vous choisie pour récupérer le résultat d'une visite ? existe-t-il d'autres solutions ?
6. Il est temps maintenant d'ajouter de nouveaux opérateurs : **Produit**, **Division** et **Difference**. Complétez les packages, en respectant le DP correspondant. Complétez les classes de test. Comment estimez-vous l'effort à fournir pour intégrer cette évolution, dans les deux cas ?
7. L'évolution de notre application se poursuit. On souhaite maintenant ajouter deux nouvelles fonctionnalités : obtenir les formes postfixée et préfixée d'une expression (sans parenthésage).

https://fr.wikipedia.org/wiki/Notations_infix%C3%A9e,_pr%C3%A9fix%C3%A9e,_polonaise_et_postfix%C3%A9e

Complétez les packages, en respectant le DP correspondant. Complétez les classes de test. Comment estimez-vous l'effort à fournir dans les deux cas ?

8. Pour être définitivement assuré(e) de comprendre la différence entre ces deux approches, ainsi que leurs avantages et inconvénients respectifs, complétez l'application en ajoutant des opérateurs n-aires (somme, moyenne, ...) et une fonction permettant de compter le nombre de **Nombre** d'une expression.

Les DP Interpreteur et Visiteur ne devraient plus avoir de secrets pour vous.