

TD patterns de comportement

L'objectif de ce TD est de se familiariser avec les différents patterns de comportement.

Commande

Une célèbre énigme mathématique consiste à chercher à obtenir un volume d'eau donné à partir de plusieurs bidons dont les volumes sont connus, et d'une source d'eau (supposée inépuisable). Pour cela, il est possible de remplir un bidon à ras bord, de le vider complètement, ou de transvaser son eau dans un autre. Dans ce cas, si c'est possible, la totalité du volume d'eau du premier bidon est transvasée dans le second, sinon seul le volume exact nécessaire pour remplir complètement le second bidon est transvasé. On suppose qu'il n'y a jamais de perte d'eau durant les transferts.

Nous souhaitons réaliser un programme afin de nous aider à répondre à cette énigme pour n bidons donnés. Le programme demande à l'utilisateur de choisir une action à réaliser avec les bidons jusqu'à obtenir le volume d'eau recherché. Lors de la simulation, l'utilisateur doit pouvoir revenir en arrière (*i.e.*, annuler la dernière action effectuée). *In fine*, la séquence des actions à réaliser pour arriver au volume recherché est donné (évidemment sans les retours en arrière).

1. Proposez un diagramme de classes qui modélise le problème en utilisant le pattern *Commande*.
2. Écrivez une implémentation des classes du diagramme¹.
3. Écrivez un programme de test qui demande à l'utilisateur d'obtenir exactement 4L d'eau avec deux bidons, un de 3L et un autre de 5L.

Observateur

Les fleurs peuvent fleurir à différentes saisons. Une fois ouvertes, les abeilles et les colibris viennent les butiner.

1. Proposez un diagramme de classes qui modélise le problème en utilisant le pattern *Observateur*.
2. Écrivez une implémentation des classes du diagramme.
3. Écrivez un programme de test qui crée des fleurs fleurissant à une saison donnée, une abeille et un colibri, et qui fait défiler les saisons.

1. Une classe `Can.java` modélisant un bidon est fournie.