

Feuille 5 - ADTs

1 Les Tours de Hanoï

Rappel : on a défini la classe abstraite suivante :

```

1 abstract class Stack {
2     abstract public void push(Object o);
3     abstract public Object pop();
4     abstract public String toString();
5     abstract public Object top();
6     abstract public boolean isEmpty();
7 }

```

Pour cet exercice vous pouvez supposer qu'on a également déjà écrit une implémentation concrète :

```

1 class ArrayStack extends Stack{
2     ...
3 }

```

Les Tours de Hanoï sont un jeu de réflexion imaginé par le mathématicien français Édouard Lucas, et consistant à déplacer des disques de diamètres différents d'une tour de "départ" à une tour d' "arrivée" en passant par une tour "intermédiaire", et ceci en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer plus d'un disque à la fois
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

On suppose que cette dernière règle est également respectée dans la configuration de départ. (Source : Wikipedia)

Le but de cet exercice est de programmer une class `Hanoi` qui va nous permettre de représenter les configurations de ce jeu. Un objet de cette class va contenir trois piles (`Stack`) représentant les trois tours du jeu. Votre classe doit aussi contenir :

1. Un constructeur `Hanoi(int n)` qui initialise le jeu avec n disques sur la première tour, et rien dans les deux autres tours.
2. Une méthode `public String toString()`
3. Une méthode `public void doSimpleMove(int from, int to)`. Cette méthode modifie l'état de l'objet en déplaçant un disque de la tour `from` vers la tour `to`. Attn : cette méthode doit vérifier que ce déplacement est légale, selon les règles du jeu. La méthode affiche l'état du jeu après son exécution.
4. Une méthode récursive `public void doComplexMove(int from, int to, int num)`, qui permet de faire un déplacement de `num` disques à la fois. Étant donné que les règles ne permettent de déplacer qu'un seul disque, cette méthode devra appeler `doSimpleMove` plusieurs fois.

2 Les Tours de Hanoï – version itérative

Dans l'exercice précédent, on a programmé une méthode récursive `doComplexMove`. Chaque appel de cette méthode utilise de manière implicite une pile (la pile d'appels). Pour cet exercice, programmez une boucle `while` qui permet de simuler la fonctionnalité de la méthode `doComplexMove`. Pour ceci, vous devez utiliser une pile (`Stack`) de manière explicite.

1. Comment représenter un mouvement complexe du jeu ? Vous pouvez soit utiliser une nouvelle classe, soit utiliser un simple tableau avec trois ints : la tour de départ, la tour d'arrivée, et le nombre de disques déplacés.
2. Programmer une boucle qui dans chaque itération, dépile un mouvement et, s'il s'agit d'un mouvement simple, l'exécute, sinon, remplace ce mouvement avec quelques mouvements plus simples sur la pile.