

Tours de Hanoi OO [cm08] - Exercice

Karine Zampieri, Stéphane Rivière, Béatrice Amerein-Soltner

Unisciel

sciel

algotprog

UNIVERSITÉ
HAUTE-ALSACE

Version 20 mai 2018

Table des matières

1	Tours de Hanoi OO / pgtourshanoi	2
1.1	Classe HDisque	2
1.2	Classe HTour	3
1.3	Classe ToursHanoi	3
1.4	Programme de test	4

C++ - Tours de Hanoi OO (TP)



Mots-Clés Classes ■

Requis Structures de base, Structures conditionnelles, Algorithmes paramétrés, Structures répétitives, Tableaux, Récursivité des actions, Classes, Classes (suite) ■

Difficulté ●○○ (40 min) ■



Objectif

Cet exercice réalise une version OO des Tours de Hanoi permettant de visualiser les différents piliers et déplacements.

1 Tours de Hanoi OO / pgtourshanoi

Voici un extrait d'exécution du résultat attendu :

```
# de disques? 3
-      |      |
---    |      |
-----|      |
*****

      |      |      |
---    |      |
-----|      -
*****

      |      |      -
      |      |      ---
      |      |      -----
*****

Nombre de déplacements = 7
```

1.1 Classe HDisque



Écrivez une classe `HDisque` qui représente un disque de HANOÏ. Il nécessite de stocker la taille du jeu `szjeu` (entier) et la `taille` (entier) du disque.



Écrivez un constructeur par défaut initialisant la taille du jeu à 1 et la taille du disque à zéro.



Écrivez un constructeur à deux paramètres entiers `s` et `t` initialisant les attributs.



Écrivez un accesseur `getTaille` de la taille du disque.



Écrivez une méthode `toString` qui calcule et renvoie l'équivalent « chaîne » du disque. Exemple : Pour une taille de jeu `szjeu` de 3, il faut retourner la chaîne (« . » représente l'espace) :

- "...|..." si la `taille` du disque vaut 0
- "...-..." pour 1
- "...---..." pour 2
- "...-----" pour 3



Aide simple

Le nombre d'espaces `nespaces` d'un côté du pilier vaut `szjeu-1` si `taille` vaut 0, `szjeu-taille` sinon. Le nombre de tirets est `2*taille-1`.

1.2 Classe HTour

Définissez la constante `NMAXDISQUES=10` (nombre maximum de disques).



Écrivez une classe `HTour` qui représente une tour de HANOÏ. Par conséquent, elle contient un tableau `pilier` de `NMAXDISQUES` disques ainsi que l'indice `top` (entier) du sommet.



Écrivez un constructeur par défaut.



Écrivez un constructeur de paramètres `n` (entier) et `vide` (booléen) initialisant les attributs.



Écrivez des accesseurs `getDisque(k)` du disque `k` du pilier et `sommet` de l'indice du sommet.



Écrivez une méthode `push(d)` qui met un `Disque d` au sommet du pilier.



Écrivez une méthode `pop` qui dépile le disque au sommet



Écrivez une méthode `assign(taille, vide)` qui redimensionne la tour et empile les disques.

1.3 Classe ToursHanoi

Définissez la constante `NMAXTOURS=3` (nombre de tours).



Écrivez une classe `ToursHanoi` qui contient un tableau `tours` de `NMAXTOURS` tours ainsi que la `hauteur` (entier) de la tour (nombre de disques). Comme l'on souhaite également compter le nombre de déplacements, ajoutez un attribut `deplts` (entier) pour le comptage.



Écrivez un constructeur de paramètre entier `n` (nombre de disques) initialisant les attributs.



Écrivez une méthode `afficherTours` qui affiche l'état du jeu.



Écrivez une méthode **interne** `deplacer(o,d)` qui déplace le disque au sommet de la tour numéro `o` vers la tour numéro `d`.



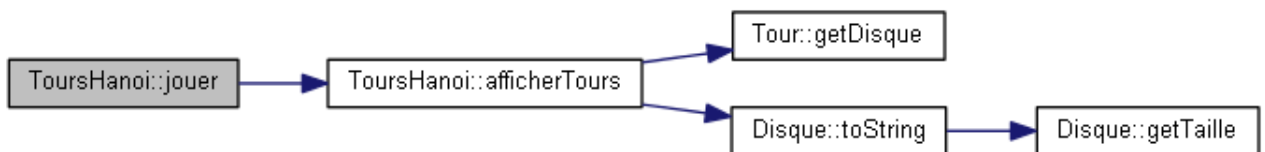
Écrivez une méthode **interne** `hanoi(n,o,d,x)` qui est la procédure récursive principale des tours de Hanoi pour `n` disques de la tour d'origine numéro `o` vers la tour destination numéro `d` en passant par la tour intermédiaire numéro `x`.



Écrivez une méthode **interne** `initialiser` qui initialise le jeu.



Jouer consiste à initialiser le jeu, afficher les tours puis à lancer la procédure `hanoi(n,o,d,x)` classique. Déduisez une méthode `jouer` qui lance le jeu.

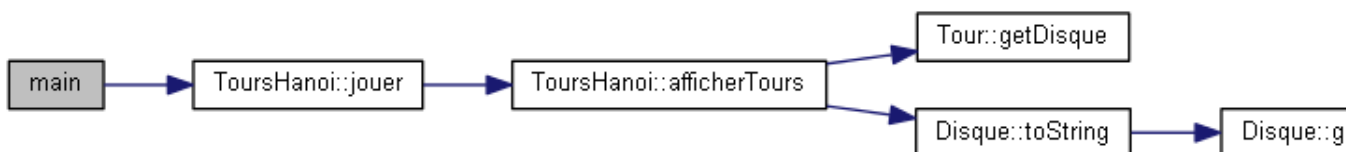


1.4 Programme de test



Écrivez un programme qui :

- Demande et saisit le nombre de disques dans un entier `n`.
- Instancie une `ToursHanoi` avec `n` disques.
- Puis lance le jeu.



Testez. Exemple de l'extrait d'exécution ci-dessus :
@[rstourshanoi.txt]