

R1.01 : Initiation au développement (partie 2) Feuille TP n° 5

version 1

Utilisation de Piles – suite et fin

OBJECTIFS PÉDAGOGIQUES :

- 3.- Apprendre les manipulations de base d'un Type Abstrait de Données Pile.
- 1.- Coder des algorithmes sous forme modulaire : création et utilisation de sous-programmes
- 2.- S'exercer à l'écriture progressive de programmes.

RESSOURCES À VOTRE DISPOSITION POUR RÉALISER CE TP :

- **tp5.pdf** : le présent sujet de TP
- **ressourcesTP5.txt** : un fichier texte contenant :
 - La déclaration et le corps du sous-programme **resoudreToursHanoiAutomatique()** résolvant les tours de Hanoi analysé en TD.
 - La déclaration et le corps d'un sous-programme **sontVides()**, utilisé par **resoudreHanoiAutomatique()** :

EXERCICES À CODER

Vous coderez les sous-programmes suivants :

```
void deplacerPetit(UnePile lesTours[], unsigned int& posTourAvecPlusPetitDisque);  
/* Déplace :  
- le plus petit disque, situé dans la tour indiquée par posTourAvecPlusPetit  
- vers la tour suivante, toujours dans l'ordre t1 -> t2 -> t3 -> t1 ...  
Et met à jour posTourAvecPlusPetit pour préparer le prochain tour  
*/  
  
void deplacerAutre(UnePile lesTours[], unsigned int posTourAvecPlusPetitDisque);  
// But : déplace le seul disque déplaçable autre que le plus petit  
// pré-condition : le déplacement est possible
```

Ces sous-programmes utilisent eux-mêmes d'autres sous-programmes. Une des stratégies préconisées en TD a mis en lumière :

- un sous-programme nécessaire à la mise en œuvre de **deplacerPetit()** :

```
unsigned int positionTourSuivante(unsigned int positionTour);  
// But : retourne la position suivante du paramètre positionTour, à savoir :  
//      1 si positionTour = 0  
//      2 si positionTour = 1  
//      0 si positionTour = 2
```

- deux sous-programme nécessaires à la mise en œuvre de **deplacerAutre()** :

```
bool disqueEstDeplacableEntre(const UnePile& tourOrigine, const UnePile& tourDestination);  
// But : retourne true s'il est possible de déplacer un disque depuis la tourOrigine vers  
// la tourDestination  
// retourne false sinon  
  
unsigned int positionTourSuivante(unsigned int positionTour);  
// But : retourne la position suivante du paramètre positionTour, à savoir :  
//      1 si positionTour = 0  
//      2 si positionTour = 1  
//      0 si positionTour = 2
```

Vous devrez aussi coder les algorithmes de ces sous-programmes.

Si la mise en œuvre de **deplacerPetit()** et **deplacerAutre()** vous ont amené à utiliser d'autres sous-programmes que ceux cités ci-dessous, vous les programmerez en complément de **deplacerPetit()** et **deplacerAutre()**.

PRÉPARATION AU TRAVAIL

1. Dans votre espace de travail, renommer le dossier tp4 en tp4tp5.
2. Sur eLearn, télécharger le fichier **ressourcesTP5.txt** et le déplacer dans le dossier tp4tp5

NEUTRALISER LA FONCTION `main()`

3. Commenter l'appel au sous-programme `resoudreToursHanoiManuel()`
4. Compiler

CODER LES SOUS-PROGRAMMES

Coder **un par un** les sous-programmes identifiés plus haut, en commençant par les sous-programmes appelés, **puis** le sous-programme appelant.

Par exemple :

- `positionTourSuivante()` puis `deplacerPetit()`
- `positionTourSuivante()`, `disqueEstDeplacableEntre()` puis `deplacerAutre()`

1. **Créer d'abord** la déclaration, **puis** un corps vide, **puis** compiler.
Placer les déclarations et corps dans les sections appropriées du fichier `main.cpp`. Par exemple, la section
`/*----Observateurs pour les 2 fonctions, et`
`/*----- Boîte à outils pour resolutionToursHanoiAutomatique () pour les 2 procédures`
2. Compléter le corps vide avec les instructions de l'algorithme. Compiler régulièrement.

AJOUTER LES RESSOURCES DANS LE FICHIER `main.cpp`

5. Ajouter la déclaration de `sontVides()` au fichier `main.cpp`, par exemple dans la section `--- Observateurs`
6. Ajouter le corps de `sontVides()` au fichier `main.cpp` à un endroit cohérent par rapport au placement de sa déclaration.
7. Compiler
8. Ajouter la déclaration de `resoudreToursHanoiAutomatique()` dans le fichier `main.cpp`, sous la déclaration de `resoudreToursHanoiManuel()`
9. Ajouter le corps de `resoudreToursHanoiAutomatique()` dans le fichier `main.cpp`, sous le corps de `resoudreToursHanoiManuel()` à un endroit cohérent par rapport au placement de sa déclaration.
10. Compiler
11. Supprimer le fichier `tp4tp5/ressourcesTP5.txt`

COMPLÉTER LA FONCTION `main()`

12. Compléter la fonction `main()` avec un appel au sous-programme `resoudreToursHanoiAutomatique()` :

`resoudreToursHanoiAutomatique(nbDisques) ;`

13. Compiler
14. Exécuter la fonction `main()` pour `nbDisques = 3` puis pour `nbDisques = 4`
15. Comparer les résultats obtenus avec les résultats montrés sur la feuille de TD n°6.
16. Corriger les programmes codés si nécessaire

RAPPELS

Avant de coder, poser le clavier et faire un algorithme

Lors du codage : appliquer toutes les recommandations vues dans la première partie de R1.01-Initiation au développement (partie 1).