Hulin Cédric

I.U.T. de NICE / COTE D'AZUR
Département Informatique
DUT S2

Année 2008_2009
Module APO-C++
Feuille_1

- -

Le TD est consacré à des révisions du langage C via la description de types abstraits de données (T.A.D.). L'objectif est de préparer l'étude des descriptions orientées objets analogues en langage C ++, qui seront développées dans les TP suivants.

Exercice Le T.A.D. *Pile*

Le T.A.D. ***Pile*** permet d'exécuter les opérations de base sur la structure de données standard "Pile" : création, empiler un élément, dépiler un élément, destruction.

Détailler le code source des définitions en langage C correspondant à l'interface fonctionnelle décrite ci dessous :

```
struct sPile {

    void** m_pT;
    unsigned int m_sommet;
    unsigned int m_taille;

};

typedef struct sPile Pile;

Pile* construire(int);
void* empiler(Pile*, void*);
void* depiler(Pile*);
bool  detruire(Pile*);
```

Cette interface fonctionnelle est générique : elle est indépendante du type des éléments à mémoriser dans la pile. La fonction *empiler* devra interdire de placer en sommet de pile un pointeur NULL. Le retour de chaque fonction sera le pointeur NULL en cas d'anomalie.

Exécuter avec succès tous les tests unitaires fournis.

Exercice L'ensemble ZxZ*

Le T.A.D. ***ZxZ*** permet d'exécuter les opérations de base usuelles sur les fractions : création, réduction, +, -, *, /, destruction.

Détailler le code source des définitions en langage C correspondant à l'interface fonctionnelle décrite en annexe. Le T.A.D. devra se protéger contre toute tentative de construction d'une fraction invalide.

Toute fraction créée devra être réduite préalablement à sa mise à disposition pour toute fonction du T.A.D.

La description inclut l'ensemble Z des entiers signés et une fonction *visualiser* permettant d'afficher toute fraction sous la forme mathématique usuelle : N/D. On a cet effet les possibilités de description polymorphique de l'extension C++.

Les fonctions de servitude PGCD et PPCM ont déjà été décrites dans le module API. Leur définition est fournie en annexe.

Exercice Le T.A.D. *Vecteur*

Le T.A.D. ***Vecteur*** permet de modéliser l'espace vectoriel RxR. Il a été détaillé en cours.

Construire le projet C++ correspondant.

Analyser et exécuter avec succès chacun des tests unitaires fournis.