Feuille 4 : Piles et Files

1 Piles

On rappelle que le type abstrait `Pile de objet` est défini par les primitives suivantes :

```
fonction valeur(val P:Pile de objet):objet;
fonction pileVide(val P:Pile de objet):booleen;
fonction creerPile(ref P:Pile de objet): vide;
fonction empiler(ref P:Pile de objet, val x:objet):vide;
fonction depiler (ref P:Pile de objet):vide;
fonction detruirePile(ref P:Pile de objet):vide;
```

Exercice 4.1

Évaluer à l'aide des primitives du type abstrait `Pile de objet` la fonction suivante et donner le contenu de la pile après exécution.

```
fonction essai_pile():Pile de car;
var P: Pile de car;
var C:car;
debut
   creerPile(P);
   empiler(P,'A');
   depiler(P);
   empiler(P,'B');
   C=valeur(P);
   empiler(P,'a');
   empiler(P,C);
   retourner(P);
fin
```

Exercice 4.2

On se donne une pile P_1 contenant des entiers positifs.

1. Écrire un algorithme pour déplacer les entiers de P_1 dans une pile P_2 de facon à avoir dans P_2 tous les nombres pairs en dessous des nombres impairs.
2. Écrire un algorithme pour copier dans P_2 les nombres pairs contenus dans P_1 . Le contenu de P_1 après exécution de l'algorithme doit être identique à celui avant exécution. Les nombres pairs dans P_2 doivent être dans l'ordre où ils apparaissent dans P_1 .

Exercice 4.3 *Utiliser une pile pour*

1. Évaluer une expression arithmétique postfixée codée sur un tableau de caractères, en supposant pour simplifier que
   — tous les opérateurs sont binaires et limités à $+, -, *$ et $/$,
   — on utilise uniquement des nombres sur un caractère.
2. Transformer une expression arithmétique infixée valide avec parenthèses en une expression arithmétique postfixée codée sur un tableau de caractères, en supposant pour simplifier que tous les opérateurs sont binaires et limités à $+, -, *$ et $/$.

Exercice 4.4

Un problème fréquent d'un compilateur et des traitements de textes est de déterminer si les parenthèses d'une chaîne de caractères sont balancées et proprement incluses l'une dans l'autre. Par exemple, la chaîne ((( )) ( ))( ) est bien balancée et proprement écrite, tandis que les chaînes)( ) ou ( )) ne le sont pas. Écrire une fonction :

1. Qui retourne vrai si une chaîne de caractères est proprement écrite et bien balancée, et faux sinon.
2. Qui retourne la position de la première parenthèse qui déroge à cette règle si la chaîne n'est pas bien écrite et bien balancée.

2 Files

On rappelle que le type abstrait `File de objet` est défini par les primitives suivantes :

```
fonction valeur(val F:file de objet):objet;
fonction fileVide(val F:file de objet):booleen;
fonction creerFile(ref F:file de objet): vide;
fonction enfiler(ref F:file de objet, val v:objet):vide;
fonction defiler(ref F:file de objet):vide;
fonction detruireFile(ref F:file de objet):vide;
```

Exercice 4.5

Évaluer à l'aide des primitives du type abstrait `File de objet` la fonction suivante et donner le contenu de la file après exécution.

```
fonction essai_file():File de car;
var F: File de car;
var C:car;
debut
   creerFile(F);
   enfiler(F,'A');
   enfiler(F,'B');
   enfiler(F,'C');
   defiler(F);
   C=valeur(F);
   enfiler(F,'a');
   defiler(F);
   enfiler(F,'b');
   enfiler(F,C);
   retourner(F);
fin
```

Exercice 4.6

Un palindrome est une chaîne de caractères qui se lit de la même manière de gauche à droite et de droite à gauche. En utilisant un nombre fixe de piles et de files, des primitives du type Pile de objet et file de objet , et un nombre fixe de variables de type entier et car, écrire un algorithme qui détermine si une chaîne de caractères est un palindrome. On suppose que la chaîne de caractères est passée en paramètre dans une liste simplement chaînée qui ne sera parcouru qu'une seule fois.

L'algorithme doit donner en sortie vrai ou faux selon le cas.

Exercice 4.7

On souhaite implémenter le type abstrait File à l'aide du type Pile.

- Combien de Piles seront nécessaires ? Comment minimiser le nombre de déplacements des objets entre les différentes Piles ?
- Peut-on écrire toutes les primitives du type File avec une complexité en O(1) ? Justifiez.
- Écrire les primitives.

Problème récurrent

Exercice 4.8 Gestion d'une piste d'atterrissage

Un avion est caractérisé par un enregistrement contenant :
— un indicatif (6 caractères)
— sa destination (30 caractères)
— son autonomie résiduelle de carburant, comptée en heures de vol (entier)
— deux booléens indiquant s'il y a un pirate à bord et s'il y a le feu.

Le problème consiste à

1. définir les structures de données nécessaires à la gestion d'une piste d'atterrissage ;
2. définir et écrire une fonction calculant la priorité d'un avion pour l'utilisation de la piste ;
3. définir et écrire les fonctions nécessaires à la gestion complète de la piste (on envisagera la suppression d'un avion piraté de la file d'attente lorsque le pirate a mis sa menace de détournement à exécution).

Quelles notions vues dans ce TD peuvent permettre d'amorcer la résolution du problème ?

Annexe A **Implémentation du type abstrait** *Pile de objet* **par un tableau**

```
Pile de objet= structure
                 taille: entier;
                 sommet: entier;
                 pile: tableau[1..taille] de objet
               finstructure
```

Annexe B **Implémentation du type abstrait** *Pile de objet* **par une liste** listeSC

```
Pile de objet= listeSC de objet
```

Annexe C **Implémentation du type abstrait** *file de objet* **par un tableau**

On utilise le tableau de manière circulaire avec un pointeur donnant le premier élément et un autre donnant le dernier.

```
file de objet= structure
                 taille: entier;
                 premier: entier;
                 dernier: entier;
                 plein: booleen;
                 file: tableau [0..taille-1] de objet
               finstructure
```

Annexe D **Implémentation du type abstrait file de objet par une liste** listeDC

```
file de objet= listeDC de objet
```