TRAVAUX PRATIQUES : série n°2

Implémenter les types abstraits **VECTEUR ,LISTE, PILE** et **FILE**

L'objet de ce TP est de fournir une implémentation **correcte** pour les types abstraits **VECTEUR, LISTE, PILE** et **FILE** en s'appuyant sur les spécifications Casl proposées.

Le travail doit obligatoirement se dérouler en suivant les **5** étapes du cycle de développement initié lors des Tp série n°1 (type abstrait des polynômes), à savoir :

étape 1: **Spécification** du type abstrait (elle est fournie et doit être éditée sous emacs)
étape 2: **Validation** de la spécification sous **hets.**
étape 3: Spécification des opérations (constructeurs) du type
étape 4: **Implémentation** de la spécification.
étape 5: **Vérification** de l'implémentation.

Chaque binôme est invité à choisir d'implémenter **un seul** parmi les types abstraits suivants:

- type **VECTEUR**,
- type **LISTE**,
- type **PILE** et **FILE**.

I- TYPE ABSTRAIT « VECTEUR »

Un **vecteur** est un ensemble **dynamique** d'objets occupant des rangs entiers successifs permettant :

- la **consultation**,
- la **modification**,
- l'**insertion**
- et la **suppression**

d'éléments à des **rangs arbitraires**.

1-Définition

Un vecteur V est une **suite finie** d'objets e_i repérés selon leur **rang**:

$$v = [\, e_1, \ldots, e_n \,]$$

Ordre dans un vecteur

L'**ordre** dans un vecteur est fondamental.

Cet ordre ne porte pas sur les **valeurs** des objets e_i mais sur les **rangs** occupés par ces objets.

2- Opérations de base

L'objet occupant le **premier rang** d'un vecteur est sélectionné par la fonction **premier**():

premier: Vecteur[Elem] $\rightarrow$? Elem

Par ailleurs, il existe une fonction de **succession** notée **succ**:

succ: Vecteur[Elem] x Elem $\rightarrow$? Elem

Le nombre d'objets dans un vecteur v est appelé la **taille** de v :

taille: Vecteur[Elem] $\rightarrow$ Nat

A la différence d'un tableau, un vecteur est de **taille variable.**

La taille d'un vecteur **varie** lorsqu'on y **insère** ou on y **supprime** des objets

La **taille** est nulle lorsqu'on a un **vecteur vide**.

3- Autres opérations de base

Les autres opérations de **base** que l'on peut effectuer sur les **vecteurs** sont :

- **créer** un vecteur vide:

vecteurVide : Vecteur[Elem]

- **insérer** un nouvel élément qui sera de rang i:

insérer:Vecteur[Elem] x Nat x Elem $\rightarrow$? Vecteur[Elem]

- **modifier** un élément de rang i:

modifier: Vecteur[Elem] xNat x Elem$\rightarrow$? Vecteur[Elem]

-**supprimer** l'élément de rang i:

supprimer: Veteur[Elem] x Nat $\rightarrow$? Vecteur[Elem]

- **accéder** à l' élément de rang i :

ieme : Vecteur[Elem] x Nat $\rightarrow$? Elem

II- TYPE ABSTRAIT « LISTE »

1-Définition

Une liste linéaire λ est un ensemble d'objets :

- **dynamique**,
- **ordonné**,

dont les objets sont accessibles **relativement les uns aux autres**, sur la base de leur position.

2-Opérations sur une liste

L'objet occupant le **premier rang** d'une liste est sélectionné par la fonction **tête**(v):

tête: Liste[Elem] $\rightarrow$? Elem

Le nombre d'objets d'une liste λ est appelé la **taille** de λ :

taille: Liste[Elem] $\rightarrow$ Nat

La liste qui ne contient aucun élément est la **liste vide**:

listeVide: $\rightarrow$ Liste[Elem]

L'opération **cons** **construit** une liste λ en insérant un objet en **tête** d'une autre liste λ':

cons: Elem x Liste[Elem] $\rightarrow$ Liste[Elem]

L'opération **fin** qui retourne la liste amputée de son premier objet:

fin: Liste[Elem] $\rightarrow$ Liste[Elem]

- **tête** qui retourne le **premier objet** d'une liste :

tête: Liste[Elem] $\rightarrow$ Elem

III- Types abstraits «Pile» et «File»

Pour beaucoup de traitements, les seules opérations à effectuer sur les **listes** sont:

- des **insertions aux extrémités**,
- des **suppressions aux extrémités**.

D'où l'importance particulière accordée aux notions de **pile** et de **file**.

1-Type abstrait Pile

Dans les piles les **insertions** et les **suppressions** se font à une **seule** extrémité appelée **sommet** de la pile.

Les opérations de base sur les piles sont:

- créer une **pile vide** :

  pileVide : Pile

- **empiler** un objet:

  empiler : Pile x Elem $\rightarrow$ Pile

- **retirer** l'objet qui se trouve au sommet:

dépiler : Pile $\rightarrow$? Pile

-**tester** si une pile est vide :

estVide : Pile $\rightarrow$Booléen

- **accéder** au sommet d'une pile :

sommet: Pile $\rightarrow$? Elem

2-Type abstrait File

Une **file** est une **liste** où on fait:

-les **adjonctions** à une extrémité,

-les accès et les **suppressions** à l'autre extrémité.

Par analogie avec les **files d'attente** on dit que l'objet présent depuis le **plus longtemps** est le **premier**.

Les opérations de base sur les files sont:

-créer une file vide

fileVide : File

- **ajouter** un élément dans la file :

enfiler: File x Elem $\rightarrow$ File

- **retirer** le premier élément de la file

defiler: File $\rightarrow$? File

-accéder au premier élément de la file :

premier : File $\rightarrow$? Elem

-tester si une file est vide

estVide : File $\rightarrow$ Booleen