TP 8b : Mise en œuvre d'une liste

Objectif pédagogique :

Passage du Type Abstrait de données à la mise en œuvre, utilisation des chaînages.

Prérequis :

Travaux pratiques et cours d'initiation à la POO, cours de SDD.

Sujet :

Nous allons réaliser la mise en œuvre d'une liste pouvant contenir des objets avec double chaînage à partir du TAD liste suivant qui est PROCHE de celui vu en cours. Remarque : dans ce TAD on ne peut pas être positionné en dehors de la liste. On veillera à respecter les signatures, préconditions et axiomes donnés. L'implémentation se fera par double chaînage avec éléments fictifs en tête et en queue.

```
TAD Liste
fonctions
  listeVide : Liste  → booléen
  aSuivant  : Liste  → booléen (vrai si l'elt courant a un successeur)
  aPrecedent: Liste  → booléen (vrai si l'elt courant a un prédecesseur)

  element   : Liste  → Objet (rend l'elt courant / fonction partielle)
  ajouter   : Liste x Objet → Liste (ajoute un elt après l'elt courant
                                     ou en tête dans une liste vide)
  enlever   : Liste  → liste (enlève l'elt courant / fonction partielle)

  suivant   : Liste  → Liste (fonction partielle)
  precedent : Liste  → Liste (fonction partielle)

  new :              → Liste (crée une liste vide)
préconditions
  element(L)  : non listeVide(L)
  enlever(L)  : non liste vide(L)
  suivant(L)  : aSuivant(L)
  precedent(L): aPrecedent(L)
axiomes
  listeVide(new())          non listeVide(ajouter(L,e))

  aSuivant(precedent(L))  // si précondition sur precedent vérifiée
  non aSuivant(enqueue())   non aSuivant(ajouter(new,e))

  aPrecedent(suivant(L)) // idem
  non aPrecedent(entete())  non aPrecedent(ajouter(new,e))

  element(ajouter(L,e))=e // l'elt ajouté devient l'elt courant
  enlever(ajouter(L,e))=L // idem vis à vis du parcours

  element(precedent(ajouter(L,e))) = element(L) // signifie qu'on ajoute
                                                   après l'élément courant

  precedent(suivant(L)) = L // si précondition vérifiée
  suivant(precedent(L)) = L // idem
```

TRAVAUX PRATIQUES Initiation à la Programmation Orientée Objet en Java et structures de données [2004]

Q0) Définition des éléments

Définir une classe Maillon qui servira à stocker les éléments et les chaînages. Cette classe doit posséder 3 attributs (**`Maillon suivant, Maillon precedent`** et **`Maillon valeur`**) ainsi que les méthodes d'accès à chacun de ces attributs (**`Maillon getSuivant(), setSuivant(Maillon m), Maillon getPrecedent(), setPrecedent(Maillon m), setValeur(Object e)`** et **`Object getValeur()`**)

Q1) Définition de l'interface de la liste

Définir une interface Liste à partir du TAD ainsi que les classes Exception que vous utiliserez pour gérer les préconditions.

Q2) Mise en œuvre de la liste

Mettre en œuvre une Classe ListeDbleImpl à double chaînage en veillant à respecter les axiomes et les préconditions donnés ici

Q3) tester la mise en œuvre de la liste

On écrira un programme (méthode main) qui vérifiera que la mise en œuvre est correcte vis à vis du TAD.

Remarques

Ce document se trouve sous `/home/clyde/d1/insa3/commun/java/tp8b`