TP5 : listes et itérateurs

Le but du TP est l'implémentation de listes avec itérateurs. Les listes utilisées seront des listes doublement chaînées. L'unité `U_ListeDC` est donnée avec une fonction d'ajout en tête de liste.

Exercice 1 : Création des itérateurs

Q 1.1 Déclarer le type `ITERATEUR` dans l'unité `U_ListeDC.pas`.

Q 1.2 Compléter le code des fonctions :

```
function iterateurEnDebut (l : P_LISTE) : ITERATEUR;
function iterateurEnFin (l : P_LISTE) : ITERATEUR;

function estEnFin (it : ITERATEUR) : BOOLEAN;
function estEnDebut (it : ITERATEUR) : BOOLEAN;

procedure avancer (var it : ITERATEUR);
procedure reculer (var it : ITERATEUR);
function valeur (it : ITERATEUR) : ELEMENT;
```

Dans les fonctions `iterateurEnFin` et `iterateurEnDebut` on passe un pointeur sur la liste pour s'en souvenir dans l'itérateur (sert lors des modifications de liste). Ces deux fonctions lèvent des exceptions si la liste est vide (on considère qu'un itérateur ne peut pas être défini sur une liste vide). Les procédures `avancer` et `reculer` lèvent des exceptions si on déborde de la liste.

Q 1.3 Tester en implantant dans un programme de test `testListeDC.pas` une fonction d'affichage de la liste à l'endroit et à l'envers qui utilisent les primitives sur les itérateurs :

```
procedure afficherEndroit (l : LISTE);
procedure afficherEnvers (l : LISTE);
```

Exercice 2 : Insertion avec des itérateurs

Q 2.1 Compléter le code des fonctions :

```
procedure insererAvant (var it : ITERATEUR);
procedure insererApres (var it : ITERATEUR);
```

Pour ces deux procédures, une fois l'insertion réalisée, l'itérateur sera placé sur le nouvel élément.

Q 2.2 Tester les nouvelles fonctionnalités (comme un itérateur n'est défini que sur des listes à au moins un élément, il faudra ajouter le premier élément avec la procédure `ajouterEnTete`).

Q 2.3 Créer, dans le programme de test, une procédure :

```
procedure insererTrie (const x : ELEMENT; var l : LISTE);
```

qui ajoute `x` à la liste `l` en conservant la liste dans un ordre croissant, puis la tester.