
Programmation orientée objet

TD n° 1

Exercice 1. Liste chaînée générique

Nous souhaitons réaliser une liste simplement chaînée générique permettant de stocker des éléments d'un type quelconque `T`. La liste générique implémentera les fonctions suivantes :

- ▶ `int size()` qui renvoie la taille de la liste,
- ▶ `T front()` qui renvoie l'élément en tête de liste,
- ▶ `T back()` qui renvoie l'élément en queue de liste,
- ▶ `void push_front(T e)` qui insert l'élément `e` en tête de liste,
- ▶ `void push_back(T e)` qui insert l'élément `e` en queue de liste,
- ▶ `void insert(int pos, T e)` qui insert l'élément `e` après la position `pos`,
- ▶ `void erase(int pos)` qui supprime l'élément situé à la position `pos`,
- ▶ `void clear()` qui supprime tout les éléments de la liste.

Q- 1.1 Modélisation

Modélisez cette liste générique à l'aide de classes.

Q- 1.2 Implémentation

Implémentez cette liste générique.

Q- 1.3 Test

Testez la liste générique avec des entiers.

Exercice 2. Répertoire téléphonique

Nous souhaitons réaliser une application de gestion d'un répertoire téléphone simple. Une entrée du répertoire sera appelée contact et contiendra 4 champs : nom, prénom, téléphone et email. Les entrées du répertoires seront stockées dans une liste (nous utiliserons la liste générique de l'exercice précédent).

Différentes fonctions permettront d'éditer un contact :

- ▶ `void edit_firstname(...)` pour éditer le prénom,
- ▶ `void edit_lastname(...)` pour éditer le nom,
- ▶ `void edit_number(...)` pour éditer le numéro de téléphone,
- ▶ `void edit_email(...)` pour éditer l'email,

Q- 2.1 Modélisation

Proposez une modélisation des entrées du répertoire.

Q- 2.2 Implémentation

Implémentez la modélisation à l'aide de classe.

Q- 2.3 Répertoire

Le répertoire doit permettre de gérer une liste de contact. Utilisez la liste chaînée générique pour implémenter un répertoire téléphonique. Le répertoire devra permettre de réaliser les fonctions suivantes :

- ▶ `void add(...)` ajouter un contact,
- ▶ `void sup(...)` supprimer un contact,
- ▶ `void clear()` supprimer tout les contacts.

Q- 2.4 Répertoire bis

Utilisez le conteneur `list` de la STL pour implémenter un nouveau répertoire téléphonique.

Q- 2.5 Gestion de fichiers

Ajouter les fonctions suivantes au répertoire :

- ▶ `void save(...)` permettant d'enregistrer les contacts dans un fichier,
- ▶ `void load(...)` permettant de lire une liste contacts à partir d'un fichier.

Q- 2.6 Tests

Réaliser des programmes de tests afin de tester l'ensemble des fonctionnalités de votre production.