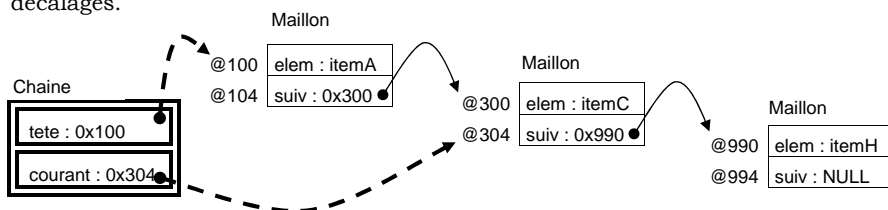


TRAVAUX PRATIQUES – Semaine n°5

Thèmes

- Chaîne, file et liste
- Implémentation de file et de liste avec un composant de chaîne

La chaîne (de maillons) est une structure concrète usuelle pour stocker une liste d'items. Son avantage, par rapport au tableau dynamique, est de diminuer le coût d'opérations telles que l'insertion ou la suppression d'un item en évitant les décalages.



Les types `Chaine` et `Maillon` choisis sont les suivants :

```
struct Chaine {
    Maillon* tete;           // Pointeur sur le maillon de tête
    Maillon** courant;      // Superpointeur (pointeur de pointeur)
                           // désignant le maillon courant
};

struct Maillon {
    Item elem;              // Élément (du maillon) de la liste chaînée
    Maillon* suiv;         // Pointeur sur le maillon suivant
};
```

Les opérations définies sur la chaîne de maillons sont les suivantes :

- (1) initialisation de la chaîne, désallocation de la chaîne, test de chaîne vide,
- (2) positionnement -en début de chaîne, -en fin de chaîne, -au maillon suivant, test de début de chaîne, test de fin de chaîne,

- (3) lecture de l'item du maillon courant, écriture d'un item dans le maillon courant, insertion d'un nouveau maillon avant le maillon courant, suppression du maillon courant.

Le but de ces exercices est (1) de compléter le composant de chaîne (`Chaine`) qui vous est donné et, à partir de cette structure de données concrète, de réécrire les composants (2) de file et (3) de liste.

Exercice 1. Composant de chaîne

- 1.1. Le but de cet exercice est de compléter le composant de chaîne.

Créez un nouveau projet de type « `Projet C++` » `sem06-tp-Cpp1`. Importez à partir des sources de TP les fichiers dans `Exo1` (`Chaine.h`, `Chaine.cpp`, `Item.h`, `Position.h`, `Position.cpp` et `testChainePositions.cpp`).

- 1.2. Ecrivez dans le fichier source `Chaine.cpp` le code de la fonction suivante :

```
void supprimer(Chaine& c);
```

- 1.3. Complétez le code de `testChainePositions.cpp` qui teste les différentes fonctions du composant de chaîne. Exécutez le programme de test.

Exercice 2. Composant de file d'items

Le but de cet exercice est de réécrire un composant de file d'items en utilisant comme structure de stockage la chaîne de maillons précédemment décrite.

- 2.1. Créez un nouveau projet de type « `Projet C++` » `sem06-tp-Cpp2`. Recopiez les fichiers `Chaine.h`, `Chaine.cpp`, `Item.h`, `Position.h`, `Position.cpp` du projet précédent. Importez à partir des sources de TP les fichiers dans `Exo2` (`File.h`, `File.cpp` et `testFilePositions.cpp`).
- 2.2. Complétez le code du fichier `File.cpp`.
- 2.3. Complétez le code du fichier `testFilePositions.cpp` qui teste votre composant de file. Exécutez le programme de test.

Exercice 3. Composant de liste d'items

Le but de cet exercice est de réécrire un composant de liste d'items en utilisant comme structure de stockage la chaîne de maillons précédemment décrite.

- 3.1. Créez un nouveau projet de type « `Projet C++` » `sem06-tp-Cpp3`. Recopiez les fichiers `Chaine.h`, `Chaine.cpp`, `Item.h`, `Position.h`, `Position.cpp`, du projet précédent. Importez à partir des sources de TP les fichiers dans `Exo3` (`Liste.h`, `Liste.cpp` et `testListePositions.cpp`).
- 3.2. Ecrivez dans le fichier source `Liste.cpp` le code de la fonction suivante :

```
void deplacer(Liste& l, unsigned int i);
```
- 3.3. Complétez le code du fichier `testListePositions.cpp` qui teste votre composant de liste. Exécutez le programme de test.