



HLIN302 – TP noté

Programmation impérative avancée
Clément AGRET, Pascal GIORGI et Alban MANCHERON
[Durée 1h30]

1 On enfile des rendez-vous

Dans cet exercice, nous souhaitons gérer une file de rendez-vous patients d'un cabinet médical en utilisant une structure de données non-linéaire permettant l'ajout en tête ou le retrait en queue de file en temps constant. Pour cela, nous définissons les deux classes suivantes :

- La classe *Rdv* contient un nom de fichier (chaîne de caractères), un type de rendez-vous (CONTROLE, MALADIE, VACCIN, CERTIFICAT, AUTRE) ainsi que l'adresse mémoire du prochain rendez-vous. Un rendez-vous peut être le dernier et donc ne pas avoir de rendez-vous suivant.
- La classe *FileDeRdv* contient uniquement les adresses mémoires des objets *Rdv* se trouvant au début et à la fin de la file. Bien entendu, chaque objet *Rdv* permettra de récupérer l'objet suivant dans la file. La file doit permettre de ne contenir aucun rendez-vous.

1.1 La classe *Rdv*

1. Cette classe doit permettre de construire uniquement un *Rdv* à partir d'une chaîne de caractère (le nom du patient) et de sa catégorie (CONTROLE, MALADIE, VACCIN, CERTIFICAT, AUTRE). Bien évidemment, la mise en file des objets de la classe *Rdv* n'est pas gérée par cette classe, ce sera le rôle de la classe *FileDeRdv*.



Cette classe ne gère aucune donnée dynamique, donc certaines méthodes ne sont pas obligatoires.

2. Cette classe doit également permettre de modifier ou de récupérer l'objet *Rdv* suivant dans la file. Par contre, il doit être impossible de modifier le nom du patient ainsi que la catégorie d'un objet *Rdv* une fois que celui-ci a été créé.
3. Écrire les fichiers de déclaration et de définition de la classe *Rdv*.
4. Écrire un programme qui enfile manuellement (sans la classe *FileDeRdv*) des rendez-vous correspondant aux patients "Patient_<X>" où <X> est en entier positif allant de 1 à 100.

1.2 La classe *FileDeRdv*

1. Cette classe doit permettre *a minima* de construire une file d'objets *Rdv* vide, d'ajouter un objet *Rdv* à la fin de la file et de retirer un objet *Rdv* du début de la file.



Cette classe devra gérer l'allocation dynamique des objets *Rdv* enfilés et garantir que l'ajout (respectivement la suppression) de l'objet *Rdv* à la fin (resp. au début) de la file se fait en temps constant.

2. Comme cette classe doit gérer explicitement la mémoire des objets *Rdv* enfilés, nous allons définir deux fonctions (en dehors de cette classe) qui permettent à partir de l'adresse d'un objet *Rdv* de :

— supprimer de la mémoire cet objet *Rdv* ainsi que tous ceux qui le suivent dans la file :

fonction **void** *suppress*(*Rdv* *);

— dupliquer en mémoire cet objet *Rdv* ainsi que ceux qui le suivent :

fonction *Rdv** *duplicate*(**const** *Rdv* *).

Ces deux fonctions suivront le même schéma récursif.

- dupliquer/supprimer l'objet *Rdv* donné par son adresse.
 - si cet objet *Rdv* était suivi d'un autre objet *Rdv* alors on effectue la même action récursivement sur cet objet.
3. Donner le code du constructeur par copie, du destructeur et de l'opérateur d'affectation pour cette classe en utilisant les fonctions *duplicate* et *suppress*.
 4. Écrire les fichiers de déclaration et de définition complets de la classe *FileDeRdv*.
 5. Écrire un second programme qui crée une *FileDeRdv* contenant les rendez-vous correspondant aux patients "Patient_<X>", où <X> est en entier positif allant de 1 à 100. Ce programme devra afficher cette *FileDeRdv* sur la sortie standard *cout*, et l'écrire dans un fichier appelé *rendez-vous.cfg*.
 6. Écrire un programme *liste_rdv* qui permet de passer en paramètre de la ligne de commande un nom de fichier dans lequel on puisse récupérer la description d'une file de rendez-vous. Ce programme construira la file de rendez-vous correspondant au fichier et l'affichera sur la sortie standard *cout*. Votre programme devra accepter le fichier de configuration suivant :

```

1 # Tout ce qui suit le caractère '#' sera ignoré jusqu'à la fin de la ligne
2 Rdv: CERTIFICAT   A. Mancheron # En bonne santé
3 Rdv: CONTROLE    C. Agret      # Pour sa jambe
4 Rdv: VACCIN       P. Giorgi     # Non, pas la grippe...
5 Rdv: MALADIE      A. Nonime
6 Rdv: AUTRE        P. Janssen    # Le plaisir de lui caler un RDV
7 # Rdv: AUTRE      J. Répa       # Inutile d'insister!
```

Le résultat affiché attendu devra ressembler à :

```

nom@machine:~/HLIN302/CC$ ./liste_rdv patients.cfg
# Liste de rendez-vous générée automatiquement par la commande :
# ./liste_rdv patients.cfg
Rdv: CERTIFICAT A. Mancheron
Rdv: CONTROLE C. Agret
Rdv: VACCIN P. Giorgi
Rdv: MALADIE A. Nonime
Rdv: AUTRE P. Janssen
nom@machine:~/HLIN302/CC$
```

Bon Courage...