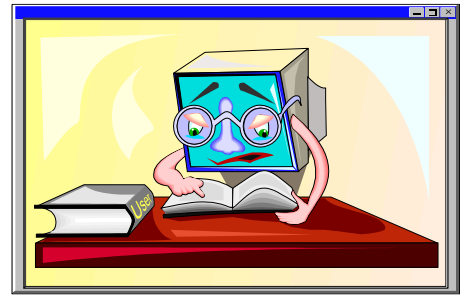


Durée: 8h

TP logiciel 7 : Répertoire de coordonnées

Objectifs:

- Utiliser Qt Creator C++ en mode console
- Produire un programme et le coder en langage C++;
- Mise en oeuvre des instructions traitant les structures de données, les pointeurs et le passage de structures aux fonctions.
- Déclaration, développement et mise en oeuvre de la classe **CIhmAgenda**



Travail à effectuer:

Coder en langage C++ un programme simple de répertoire de coordonnées..

Remarque: Commentez au maximum votre fichier pour une meilleure compréhension et servez-vous de l'aide de Qt Creator C++ pour utiliser les nouvelles fonctions.

Présentation:

On souhaite gérer un annuaire d'emails et de numéros téléphoniques. Il y a au maximum 200 noms dans ce répertoire. On veut donc un programme qui permette de gérer les noms, prénoms, adresses, email et numéros de téléphone tel que:

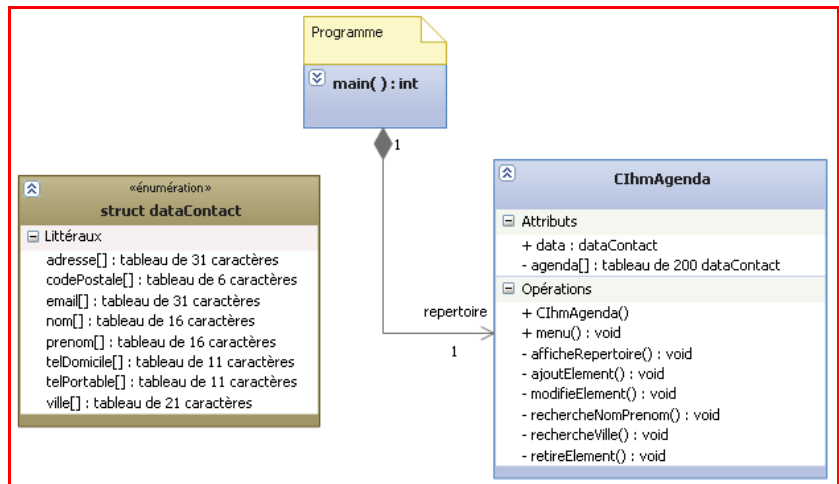
- ajouter un élément dans le répertoire,
- supprimer un élément,
- afficher l'ensemble de l'annuaire,
- rechercher un numéro par nom et prénom ou par ville,
- mettre à jour un élément du répertoire en modifiant une donnée (nom, prénom, adresse, email ou numéro téléphonique),
- ... etc ...

Chaque élément du répertoire est caractérisé par son nom (15 caractères maximum), son prénom (15 caractères), son adresse (30 caractères), son code postal (5 caractères), sa ville (20 caractères), son email (30 caractères) et ses numéros de téléphone: du domicile (10 caractères) et du portable (10 caractères).



Le programme principal est composé d'un objet **repertoire** de la classe **CihmAgenda** et exécute la méthode **menu()** de la classe. La méthode **menu()** permet de choisir la manipulation que l'on souhaite effectuer à l'objet **repertoire**.

Définir la classe **CihmAgenda** qui respecte le diagramme de classe ci-contre.



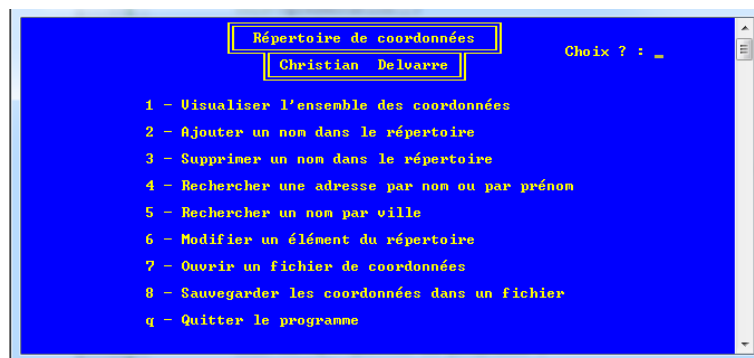
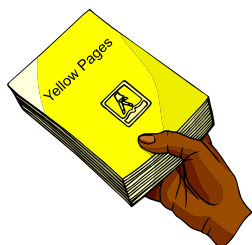
L'annuaire téléphonique comporte au maximum 200 noms. Pour représenter l'agenda dans le programme, on utilisera un tableau de la structure **dataContact**. Les noms et prénoms sont mis à partir l'indice 0 du tableau.

Un emplacement libre contient une valeur particulière (**nom[0] = '\0'** et **prenom[0] = '\0'**). On peut avoir des emplacements libres entre des emplacements occupés.

Question : quels sont les avantages/inconvénients de cette organisation dans le tableau ?

Travail à effectuer:

- 1° - Développer un nouveau programme C++, ajouter au projet, un fichier d'entête **Contact.h** pour définir la structure **dataContact** comme énoncé dans le diagramme de classe.
- 2° - Ajouter au projet, une nouvelle classe **CihmAgenda**, et compléter le fichier **cihmagenda.h** en fonction du diagramme de classe proposé.
- 3° - Développer la méthode **menu()** de la classe **CihmAgenda** en fonction de la gestion de l'annuaire présenté en première page. Cette méthode met en œuvre l'appel des différentes méthodes au travers d'un menu à choix en utilisant l'instruction **switch**.



- 4° - Déclarer dans le programme principal, **main()** un objet **repertoire** et exécuter sa procédure **menu()**.

- 4° - Ecrire la méthode **void AjoutElement()** qui permet de saisir tous les champs de la variable **data** et l'ajoute dans le tableau **agenda[]**. Si l'élément **data** n'a pas pu être ajouté, faute de place, la méthode affiche un message d'avertissement: mémoire pleine sinon on indique que l'élément est ajouté en mémoire.
- 5° - Ecrire la méthode **void AfficheRepertoire()** qui affiche le contenu de l'annuaire.
- 6° - Ecrire la méthode **void RetireElement()** qui supprime de l'annuaire un élément correspondant au nom et au prénom. La méthode affiche un message en cas de succès ou indique qu'il ne trouve pas le nom et le prénom dans la mémoire en cas d'échec.
- 7° - Ecrire la méthode **void RechercheNomPrenom()** qui affiche les informations disponibles pour le nom et prénom saisie.
- 8° - Ecrire une fonction **int RechercheVille()** qui affiche les informations disponibles pour la ville trouvée.
- 9° - Ecrire une fonction **void ModifieElement()** qui permet dans un premier temps de sélectionner un élément dans le répertoire puis dans un second temps de ne modifier que les champs souhaités de l'élément sélectionné.

