Formation C++ - Marseille, 01/12/2018 - Achref El Mouelhi

TP 3 : Conteneurs et exceptions

Exercice 1

Considérons une classe `Personne` ayant les attributs suivants :

- `num` : un attribut privé de type `int`
- `nom` : un attribut privé de type `string`
- `prenom` : un attribut privé de type `string`

1. Créez la classe `Personne` (deux fichiers : `.h` et `.cpp`
2. Définissez les getters et setters de tous les attributs
3. Définissez des constructeurs avec et sans paramètres
4. Définissez une méthode `getNomPrenom()` qui retourne le nom concaténé au prénom
5. Dans le `main`, créez quelques instances de la classe `Personne` et ajoutez les à un dictionnaire `map <string, Personne> personnes` : la clé étant la concaténation des nom et prénom et la valeur étant l'instance de `Personne`.
6. Dans une boucle `for`, affichez la clé pour les éléments de chaque itération d'indice pair et la valeur pour les autres éléments.

Exercice 2

Considérons la classe `Personne` de l'exercice 1.

1. Créez une classe `ListePersonnes` contenant un attribut `map<string, Personne> personnes`
2. Écrivez une méthode `void ajouterPersonne(Personne personne)` qui permet d'ajouter `personne` au dictionnaire `personnes` : la clé étant la concaténation des nom et prénom et la valeur étant l'instance de `Personne`.
3. Modifiez la méthode précédente pour qu'elle lève une exception de type `PersonneException` (à créer) si les nom et prénom de l'objet `personne` à ajouter existent déjà dans le dictionnaire.
4. Dans `main` :
   - instanciez un objet de la classe `ListePersonnes`,
   - demandez à l'utilisateur de saisir un numéro de personne positif,
   - redemandez-lui de saisir un numéro tant que la saisie est négative ou nulle,
   - demandez-lui de saisir un `nom` et un `prenom`,
   - créez un objet `Personne` avec les valeurs saisies et ajoutez le dans le dictionnaire de `ListePersonnes`,
   - proposez à l'utilisateur de recommencer,
   - si l'utilisateur décide de quitter le programme, affichez tout le contenu du dictionnaire de `ListePersonnes`.

Exercice 3

Considérons une classe C++ appelée `Nombre` ayant les attributs suivants :

- `var1` : un attribut privé de type `int`
- `var2` : un attribut privé de type `int`

1. Créez la classe `Nombre`
2. Définissez les getters et setters de tous les attributs
3. Définissez un constructeur avec paramètres

Considérons une deuxième classe appelée `Operation` ayant l'attribut suivant :

- `nombre` : un attribut privé de type `Nombre`

4. Créez la classe `Operation`
5. Définissez les getter et setter
6. Définissez un constructeur avec paramètre
7. Définissez une première méthode `public int division()` qui retourne le résultat de la division de `var1` par `var2` si ce dernier est différent de zéro. Sinon, elle lève une exception `OperationException` (à créer)
8. Définissez une deuxième méthode `public int racineDeLaSomme()` qui retourne la racine carrée de la somme de `var1` et `var2` si la somme est positive. Sinon, elle lève une exception `OperationException` (à créer)
9. Créez la classe `OperationException`
10. Préparez le ou les constructeurs qui permettront de traiter les exceptions décrites dans les questions précédentes
11. Dans `main`,
    - demandez à l'utilisateur de saisir deux entiers,
    - créez un objet de la classe `Nombre`,
    - demandez à l'utilisateur de saisir `1` pour avoir le résultat de la division ou `2` pour la racine carrée de la somme,
    - utilisez la classe `Operation` pour calculer le résultat de l'opération choisie,
    - demandez à l'utilisateur s'il veut choisir une autre opération.