HLIN603
Feuille de TD/TP N°3 : Contrôles d'Accès Statiques et Généricité Paramétrique

Exercice 1 : Contrôle d'accès statique-1

Donnez les accès pour les programmes suivants :

a) Accès à f protected

```
class C1
{
protected: virtual void f(){}
friend class A;
friend class B;
public:
virtual void mc1();
};

class C2 : public virtual C1
{public:
virtual void mc2();
};

void C1::mc1() {C1 *c1; c1->f(); C2 *c2; c2->f();}
void C2::mc2() {C1 *c1; c1->f(); C2 *c2; c2->f();}

class A
{public:
virtual void ma() {C1 *c1; c1->f(); C2 *c2; c2->f();}
};

class B : public virtual A
{public:
virtual void mb() {C1 *c1; c1->f(); C2 *c2; c2->f();}
};

class D
{public:
virtual void md(){C1 *c1; c1->f(); C2 *c2; c2->f();}
};

int main(){C1 *c1; c1->f(); C2 *c2; c2->f();}
```

b) Accès à f private

Avec le même programme que ci-dessus.

c) Avec une redéfinition de f

La classe C2 du programme initial est modifiée ainsi.

```
class C2 : public virtual C1
{
protected: virtual void f(){}
public:
virtual void mc2();
friend class D;
};
```

d) Accès à une partie d'une classe

Chercher les directives d'accès permettant de représenter les accès et les classes de la figure 1.

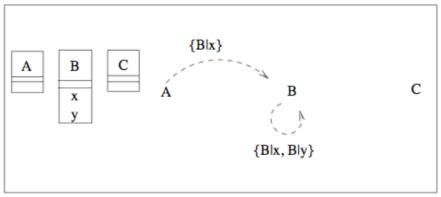

FIGURE 1 – Graphe d'accès en C++

Exercice 2 : Généricité paramétrique : Une classe paramétrée Dictionnaire

Un dictionnaire est un ensemble d'associations, c'est-à-dire de couples (clé, valeur), où la clé est d'un type donné ***TypeCle*** et la valeur d'un type donné ***TypeValeur***. Une association peut être représentée par la classe ***Assoc*** vue en cours. Un dictionnaire possède habituellement les méthodes suivantes :

- put: place une clé et une valeur associée dans le dictionnaire;
- get: prend une clé et renvoie la valeur associée;
- estVide: dit si le dictionnaire est vide;
- taille: retourne le nombre d'associations clé-valeur effectivement présentes dans le dictionnaire;
- contient: prend une clé et dit si elle est présente dans le dictionnaire;
- affiche: affiche le contenu du dictionnaire sur un flot de sortie.

1) **Spécification.** Écrivez la partie "signatures des méthodes" du fichier .h correspondant au dictionnaire générique.
   - Ajoutez la surcharge de ***l'opérateur <<*** pour l'affichage et la surcharge de ***l'opérateur =*** pour l'affectation.
2) **Implémentation par un tableau d'associations.** Le tableau d'associations est alloué dynamiquement, par défaut avec une taille de 10. L'indice d'une association est calculé grâce à une fonction de hachage appliquée à la clé (voir en annexe).
   - En utilisant les exemples de résultats de la fonction de hachage, faites quelques essais d'insertions "à la main" pour voir comment les choses se passent : par exemple put("abricot",235); put("amande",1023); put("ananas",242); put ("pomme",83); ...
   - Complétez la partie "attributs" du fichier .h, et écrivez le fichier .cc correspondant.

Pour faciliter l'écriture de certaines méthodes, on peut écrire la méthode privée :

```
void CherchCl(const TypeCle& cl, int& i, int& res);
/* cherche la cle cl dans le dictionnaire :
      si cl est presente: renvoie res=1, i indice de la case de cl dans T;
      si cl est absente et le dictionnaire non plein: renvoie res=0, i indice de case possible pour cl dans T;
      si cl est absente et le dictionnaire plein: renvoie res=2, i non significatif. */
```

3) **Utilisation.** Instanciez vos classes pour avoir un dictionnaire dont les clés sont des chaînes et les valeurs des entiers.
   - Écrivez un programme simple qui teste le fonctionnement de ce dictionnaire (ajoutez des couples, affichez le dictionnaire, rajoutez des couples pour tester l'agrandissement, affichez, etc.).
   - Utiliser la classe dictionnaire pour stocker les mots d'un texte lu sur l'entrée standard et comptabiliser le nombre d'occurrences de chacun.

Exercice 3 : Contrôle d'accès-2 (Facultatif)

a) Héritage d'implémentation : Pile

Complétez l'implémentation de la classe *Pile* vue en cours, qui hérite de manière privée (héritage d'implémentation) de *vector*.

b) Modélisation et droits d'accès : Bibliothèque

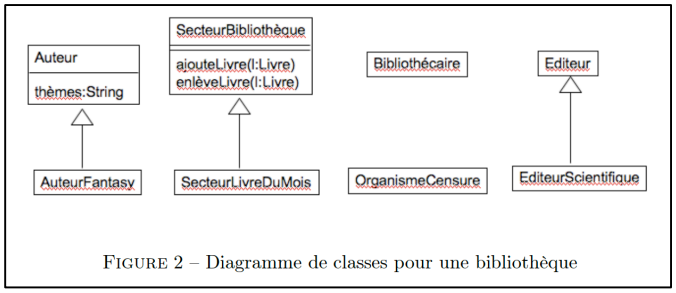

FIGURE 2 – Diagramme de classes pour une bibliothèque

Pour le diagramme de classes de la figure 2, étudiez les accès autorisés par les contraintes suivantes (représentez les dans un graphe).

- Les bibliothécaires peuvent ajouter et enlever des livres dans tous les secteurs de bibliothèque.

- Seuls les bibliothécaires ont le droit d'ajouter des livres dans les secteurs de type « le livre du mois » car le choix des livres de ces secteurs, correspondant à l'actualité du moment, leur appartient.

- Tous les éditeurs (incluant les éditeurs scientifiques) peuvent ajouter des livres dans les secteurs de bibliothèque (à l'exception des secteurs de type « le livre du mois »).

- Tous les auteurs (incluant les auteurs de fantasy) peuvent ajouter et enlever des livres (généralement leurs propres livres) dans les secteurs de bibliothèque, mais sur les secteurs « le livre du mois » ils ne peuvent qu'enlever des livres.

- Les organismes de censure peuvent seulement enlever des livres, et ceci dans tous les secteurs de bibliothèque, y compris les secteurs « le livre du mois ».

Puis discutez une solution s'en approchant en C++ que vous testerez lors des travaux pratiques. Vous comparerez les accès demandés dans la modélisation et ceux autorisés par le programme que vous avez écrit.