

Examen «C++» - M1

Lorsque des exemples sont demandés, il est interdit de répondre par un exemple issu du cours ou du sujet; une réflexion est donc demandée.

Exercice 1

1. Si l'on dit que toutes les carottes sont des plantes, qu'exprime t-on ?
 - a. Quel est le nom de cette relation ?
 - b. Comment se représente t-elle en UML ?
 - c. Comment peut-on l'exprimer en C++ ?
 - d. Comment se nomme la technique C++ ?
2. Quelle(s) différence(s) faites-vous entre la relation de généralisation et l'héritage ?
3. Comment se représente la généralisation multiple en UML ? (un exemple qui ne soit pas dans le cours)
4. Quel(s) problème(s) posent l'héritage multiple ?
5. Composition
 - a. Représentez cette relation en UML (un exemple qui ne soit pas dans le cours)
 - b. Comment s'exprime t-elle en C++ ?

Exercice 2

1. Que produit l'exécution du programme contenu dans les trois cadres à droite ? Indiquez quels affichages sont réalisés.
2. Que se passe t-il si on qualifie la méthode `f()` de `virtual` ? Affichage ?
3. Quel est le nom de ce mécanisme ?

```
class A {
public:
    void f() {
        cout << "A::f()" << endl;
    }
}
```

```
class B : public A {
public:
    void f() {
        cout << "B::f()" << endl;
    }
};
```

```
int main() {
    A a, *pa; B b, *pb;
    pa = &a; pb = &b;
    a.f(); pa->f();
    b.f(); pb->f();

    a = b;
    a.f();
    pa = pb;
    pa->f();

    exit(0);
}
```

Exercice 3

1. Si l'on souhaite créer une classe abstraite en C++, c'est-à-dire qui n'est pas instanciable ni statiquement, ni dynamiquement, comment s'y prend-on ? Donnez un exemple C++.
2. Si l'on souhaite que cette classe reste abstraite mais puisse être dérivée en une sous-classe qui elle sera instanciable que doit-on faire ? Donnez un exemple C++.
3. Qu'est qu'une fonction amie ? Donnez un exemple C++.

Exercice 4

1. Créez un type `Tableau` qui permettra de contenir un nombre fixé d'entiers.
2. Ajoutez un opérateur `[]` d'accès aux éléments et qui contrôle la validité des indices.
3. Ajoutez un opérateur `+` permettant de créer un `Tableau` dont chaque élément sera la somme des éléments de même indice dans les deux arguments.
4. Ajoutez un opérateur `+=` permettant d'ajouter à chaque élément du `Tableau` cible l'entier de même indice du `Tableau` source.
5. Ajoutez un opérateur `<<` permettant d'écrire un `Tableau` dans un flux.
6. Rendez votre type `Tableau` générique de sorte que l'on puisse créer un `Tableau` d'entiers, aussi bien qu'un `Tableau` de chaînes de caractères, ou encore un `Tableau` de `Tableau` d'entiers.

Problème

On vous propose d'écrire un logiciel permettant de générer du code C++ à partir d'un diagramme UML. Pour la version de démonstration, on se limite à des boîtes (contenant un titre et pouvant contenir des attributs et des actions, *i.e.* de simples chaînes de caractères) et à des liens entre ses boîtes (deux types de liens, les flèches et les liens simples). L'illustration suivante représente un de ces possibles diagrammes.

1. Faites une conception du problème et représentez-la en UML, en oubliant les aspects graphiques de la question!
2. Implémentez les concepts de votre diagramme UML en C++
 - a. ne pas oublier d'ajouter l'opérateur de sortie `<<` qui permettra de générer le code C++ correspondant à un diagramme et de l'afficher sur le flux
3. On désire rajouter la possibilité d'associer des noms et des nombres aux liens simples. Les noms serviront dans la génération de code afin de nommer plus explicitement les variables représentant les relations et les nombres représenteront l'arité de l'association.
 - a. que cela change-t-il dans le diagramme précédent ?
 - b. dans le code ?

