

Partiel

Lundi 28 Octobre 2013

Tous documents autorisés (sauf copie du voisin, appareillage électronique, etc). Les téléphones portables, comme tout autre moyen de communication vers l'extérieur, doivent être éteints. Le temps à disposition est d'une heure. Motivez bien vos réponses.

1 Exercice [3 points] (ctors & dtors)

Étant donné le code suivant :

```
#include <iostream>
using namespace std;
class K{
public:
    K() { cout << "K()" << endl; }
    K(int i) { cout << "K(int)" << endl; }
    ~K() { cout << "~K()" << endl; }
};
int main() {
    K k1;
    K k2(3);
    K k3('c');
    K k4(3.4);
    K k5();
    K k6(3,4);
}
```

1. le code contient-il des erreurs (empêchant la compilation)? Si oui lesquelles?
2. combien d'instances de K sont créées à l'exécution de ce programme?
3. combien d'instances de K sont-elles détruites?
4. quels sont les affichages produits à l'exécution du programme?

2 Exercice [6 points] (ctors & dtors – le retour)

Étant donné le code suivant :

```
class K {
private:
    int v;
    static int id;
public:
    K() { id++; v = id; cout << "K() v=" << v << endl; }
    K(const K &k) { id++; v=10+k.id; cout << "K(K&) v=" << v << endl; }
    ~K() { cout << "~K() v=" << v << endl; }
};

K m(K k) {
    K k1(k);
    if (rand()%2==0) { // pile ou face
        return k1;
    } else {
        K k2;
        return k2;
    }
}

int main() {
    srand(time(0)); // init générateur aléatoire
    K k;
    K p = m(k);
    return 0;
}
```

Attention certaines réponses peuvent dépendre du tirage aléatoire...

1. il manque l'initialisation (à la valeur zéro) de la variable statique de classe `id`. Où la placeriez-vous ?
En utilisant quelle syntaxe ?
2. combien d'instances de `K` sont créées à l'exécution de ce programme ?
3. combien d'instances de `K` sont-elles détruites ?
4. quels sont les affichages produits à l'exécution du programme ?

3 Exercice [5 points] (const est mon ami)

Soit l'extrait de programme suivant :

```
class nombre {
private:
    int v;
public:
    nombre() { v = 0; }
    nombre(int i) { v = 2*i; }
    nombre add(nombre n) { return nombre((v+n.v)/2); }
    nombre sub(nombre n) { return nombre((v-n.v)/2); }
    void affiche() { cout << v << endl; }
};

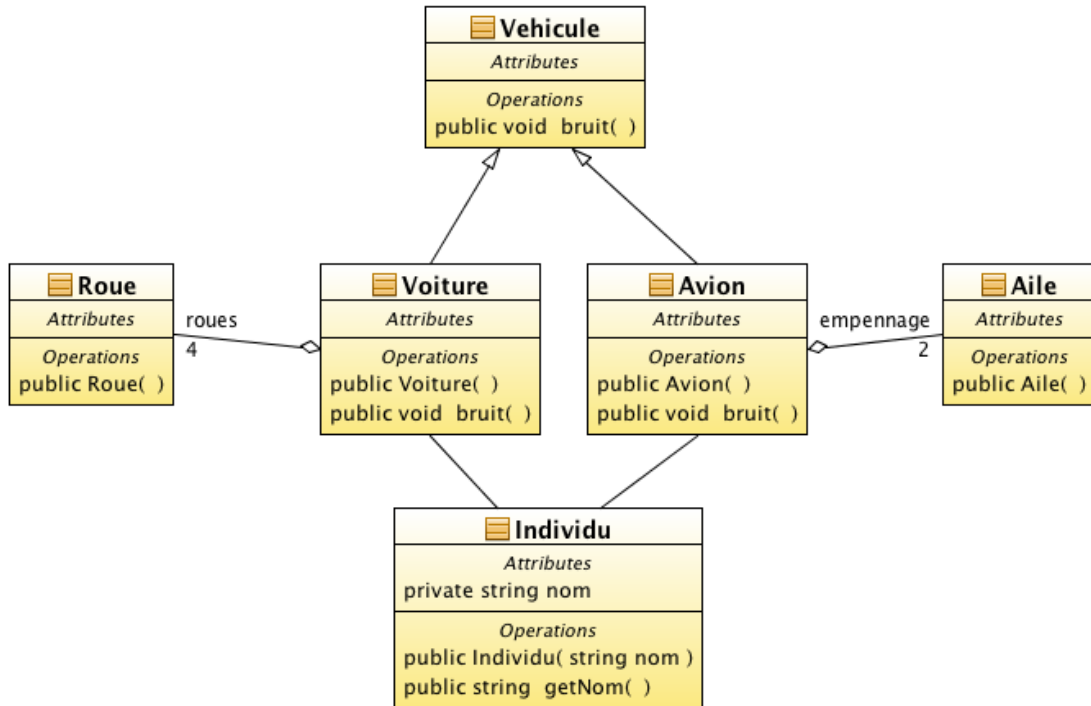
int main() {
    nombre zero;
    zero.affiche();
    nombre un(1);
    un.affiche();
    nombre dix(10);
    dix.affiche();

    nombre n = dix.add(un).sub(zero).add(un);
    n.affiche();
    return 0;
}
```

1. quels sont les affichages produits par son exécution ?
2. on souhaite déclarer le *zéro* de nos nombres comme un objet constant. Quelles modifications du code sont-elles nécessaires ?
3. on souhaite assurer que tous les objets de cette classe soient des objets constants, comment s'y prendre ?

4 Exercice [6 points] (UML)

Soit le diagramme UML suivant :



1. le diagramme est incomplet en ce qui concerne les liens qui unissent **Individu**, **Avion** et **Voiture**. Complétez.
2. pour chaque relation du diagramme, indiquez la nature de cette relation.
3. écrivez un programme C++ qui réalise la conception du diagramme (pas plus) en fournissant un `main` qui crée quelques objets et de sorte qu'une voiture fait « vroom vroom » et un avion fait « brrrrrr » comme bruit lorsqu'il sont en fonctionnement.