



TD 5 : surdéfinition, surcharge

Algorithmie

Master EEA 1^{ère} année 2006

1. Introduction

1.1. Edition du fichier

Afin d'écrire votre script C++, vous pouvez utiliser n'importe quel éditeur de texte, cependant xemacs ou nedit paraissent être un bon choix. On lance l'éditeur à partir d'une console (« nedit & »).

Le fichier aura une extension .cpp.

1.2. Compilation

Ouvrez un terminal et placez-vous dans le répertoire dans lequel vous avez enregistré votre script C. Afin de compiler votre script, utiliser la commande :

```
g++ nom_du_fichier.cpp -o nom_executable
```

Ce qui donne, par exemple, si votre script du exo1 se nomme exo1.cpp et que vous voulez que votre exécutable se nomme exo1:

```
g++ exo1.cpp -o exo1
```

Le compilateur génère alors un fichier exo1 que vous pouvez exécuter en utilisant la commande ./exo1

1.3. En tête de fichier

L'en-tête du fichier d'extension « .cpp » commencera par :

```
#include <iostream>  
using namespace std;
```

La première ligne fait appel à la librairie relative aux fonctions d'entrées sorties.

La seconde ligne définit au compilateur que l'on va utiliser les librairies standard du langage C++.

2. Exercice 1

```
#include <iostream>  
using namespace std;
```

```
const double MEASURE_MAX = 10;  
const double MEASURE_MIN = 0;
```

```
class CDevice {  
private:  
    int Id;  
    double meas;  
public:
```

```

    CDevice ();
    CDevice (int id);
    bool setValue(double val);
    CDevice operator+ (const CDevice &a);
};

CDevice::CDevice (){
    Id = 1;
    meas = 0;
}

CDevice::CDevice (int Id) {
    this->Id = Id;
    this->meas = 0;
}

bool CDevice::setValue(int val) {
    if ((val < MEASURE_MIN) || (val > MEASURE_MAX))
        return false;
    this->meas = (double) val;
    return true;
}

int main(int argc, char* argv[])
{
    CDevice dev1;
    CDevice dev2 (2);
    ...
}

```

1. Comment appelle-t-on la méthode CDevice (int Id) ?
2. Est-il possible de créer une autre méthode « setValue » qui prendrait comme paramètre un double ? Comment ?
3. On imagine pouvoir additionner deux instruments, cette opération n'affectera que le membre meas. Comment surcharger l'opérateur + pour effectuer cette action ?
4. Pourrait-on implémenter le même principe avec la multiplication en augmentant la rapidité de traitement du processeur ? Comment ?
5. Quel est le problème de cette implémentation peut adapter à la classe CDevice ?

3. Exercice 2

Cet exercice est la suite du TD précédent.

Le but de cet exercice est de manipuler une classe CComplexe ($z = a+ib$).

1. Surchargez l'opérateur + afin d'additionner deux complexes.
2. Surchargez l'opérateur * de manière à ce que cette opération soit la plus rapide possible.
3. Créez une méthode statique (ayant comme paramètre un objet CComplexe) qui affiche à l'écran un nombre complexe sous la forme partie imaginaire et partie réelle.