



Programmation générique et conception objet

Atelier n°1

Compilation et exécution C++

Exercice 1:

Objectif: apprendre à créer, à compiler et à exécuter des programmes C++.

1. Dans le répertoire TP1, créez le fichier **MonPremierProgramme.cpp**. Il permettra de générer le fichier exécutable **MonPremierProgramme.exe** qui affichera votre nom sur la sortie écran.
L'ensemble des opérations de compilation sont décrites dans le fichier **makefile**.
2. Dans le répertoire TP1, créez le fichier **makefile** contenant les instructions suivantes:

```
MonPremierProgramme: MonPremierProgramme.o
    g++ MonPremierProgramme.o -o MonPremierProgramme.exe

MonPremierProgramme.o: MonPremierProgramme.cpp
    g++ -c MonPremierProgramme.cpp
```

3. La compilation se fait grâce à l'application **Build targets** sur Eclipse ou avec la commande **make** depuis le terminal:

Terminal:

1. Ouvrez le terminal depuis le dossier où se trouve le fichier .cpp et le makefile, puis tapez la commande **make**.
2. Pour exécuter le programme .exe ainsi généré, tapez la ligne suivante:

./MonPremierProgramme.exe

Eclipse:

1. Ajoutez la cible (Target) **MonPremierProgramme** → bouton droit sur le répertoire TP1 → **Build targets** → **Create**.
2. Lancez la commande **Build** pour générer l'exécutable.
3. Pour exécuter le programme, il est indispensable de générer une configuration d'exécution associée grâce à l'application Run: → bouton droit sur le fichier **MonPremierProgramme.exe** → **Run**
4. Créez la configuration d'exécution associée « MonPremierExecutable », de type **C/C++ local** pour l'application TP1/**MonPremierProgramme.exe**. Exécuter le programme en utilisant le menu Run de la barre de Menu.

Exercice 2: Compilation séparée

Objectif: Pouvoir écrire un programme en C++ qui lit un entier sur l'entrée clavier et affiche sa parité.

1. Créez quatre fichiers dans le répertoire TP1/Exe2: **Parite.cpp**, **Parite.h**, **Paire.cpp**, **Paire.h**.
 - Le fichier **Parite.cpp** comportera une fonction **Parite** qui aura comme paramètre d'entrée un entier court (short) et comme paramètre de sortie un booléen (bool) renvoyant **true** si la parité est paire et **false** sinon.
 - Le fichier **Parite.h** contiendra le prototype de la fonction Parite
 - Le fichier **Paire.cpp** contiendra la fonction principale (main) qui sera chargée des entrées-sorties et de l'appel à la fonction Parite.
 - Le fichier **Paire.h** contiendra la déclaration des flots d'entrées/sorties et l'inclusion de **Parite.h**.
2. Créez les fichiers et ajoutez au fichier **makefile** les lignes suivantes :

```
Paire: Paire.o Parite.o
    g++ Paire.o Parite.o -o Paire.exe
Paire.o: Paire.h Parite.h Paire.cpp
    g++ -c Paire.cpp
Parite.o: Parite.h Parite.cpp
    g++ -c Parite.cpp
```

3. Générez le fichier exécutable. Exécutez le.

Exercice 3: Création de la classe Date (TP1/Exe3)

Vous devez définir une classe simple permettant la gestion d'une date.

Le cahier des charges impose que votre classe permettra l'affichage et l'incrémentation d'une date. Vous vous inspirerez de la classe **Date.java** disponible dans le cours de Programmation Objet (cf. Annexe).

1. Donnez la définition de la classe sous forme d'un fichier d'entête Date.h.
2. Implémentez le constructeur dans un fichier Date.cpp après avoir ajouté son prototype dans le fichier d'entête. Testez le dans un programme de test (main.cpp).
3. Écrivez le code correspondant à la méthode d'affichage après avoir ajouté son prototype dans le fichier d'entête. Testez le.
4. Écrivez le code correspondant à la méthode d'incrémentation après avoir ajouté son prototype dans le fichier d'entête. Testez le.
5. Ajoutez une méthode permettant la saisie d'une date à la définition de la classe et écrivez le code.

Annexe Classe.java

```
public class Date {
// attributs de classe
    private int jour, mois, annee;
    public boolean CompareTo (Date d) {
        if (annee != d.annee) return true;
        if (mois != d.mois) return true;
        if (jour != d.jour) return true;
        else return false;
    }
    /** Calcul de la date du lendemain */
    public void Incrémenter () {
// Pas de taille indiquée dans la déclaration des tableaux
        int lmois[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
// prise en compte des années bissextiles
        if (((annee % 4) == 0) && ((annee % 400) != 0)) lmois[1]++;
        jour++;
        if (jour > lmois[mois-1]) {
            jour = 1; mois++;
            if (mois == 13) {annee++; mois = 1;}
        }
    }
    /** Affichage de la date */
    public void Afficher() {
        System.out.println(jour+" "+mois+" "+annee);
    }
    /** Lecture au clavier de la date */
    private void Lire() { // méthode lecture d'une date
        jour = Keyboard.getInt("Entrez le jour");
        mois = Keyboard.getInt("Entrez le mois");
        annee = Keyboard.getInt("Entrez l'année");
    }

    /**
     * Création et initialisation d'une nouvelle instance de Date
     * @param j jour
     * @param m mois
     * @param a année
     */
    public Date(int j, int m, int a) {
        jour = j; mois = m; annee = a;
    }

    /**
     * Création et lecture clavier d'une nouvelle instance de Date
     */
    public Date() {Lire();}
    /**
     * Comparaison entre deux dates
     */
}
```

```
* @param d Date comparée
*/
} // fin de la déclaration de la classe date
```