|  |  |  |
| --- | --- | --- |
| sorbpnbw | Master Langue et Informatique  Université Paris-Sorbonne |  |

**Master Langue et Informatique**

Programmation générique et conception objet

**Atelier n° 1**

**Compilation et exécution en C++**

Les exercices qui vous sont proposés ici ont pour but d’apprendre à créer, à compiler et à exécuter des programmes C++.

## 1. Mon premier programme en C++ sous Eclipse

Le but de cette partie est d’écrire et d’exécuter un programme simple en C++ sous l’environnement Eclipse.

**Exercice 1** : Créer dans le répertoire ***tp1*** le fichier ***MonPremierProgramme.cpp***. Il permettra de générer le fichier exécutable ***MonPremierProgramme.exe***  qui écrira votre nom sur la sortie écran.

L’ensemble des opérations de compilation est décrite dans un fichier du répertoire **tp01** appelée ***makefile***:

**Exercice 2** : Créer un fichier makefile contenant les instructions suivantes

MonPremierProgramme: MonPremierProgramme.o

g++ MonPremierProgramme.o -o MonPremierProgramme.exe

MonPremierProgramme.o: MonPremierProgramme.cpp

g++ -c MonPremierProgramme.cpp

La compilation se fait grâce à l’application **Build Make Target** (bouton droit sur le répertoire **tp1**)

**Exercice 3** : Ajouter la cible (target) ***MonPremierProgramme***. Lancer la commande ***Build*** pour générer l’exécutable.

Il est indispensable, pour exécuter le programme de générer une configuration d’exécution associée grâce à l’application Run (bouton droit sur le fichier MonPremierProgramme.exe)

**Exercice 4** : Créer la configuration d’exécution associée « MonPremierExecutable », de type ***C/C++ local*** pour l’application **tp01/MonPremierProgramme.exe**. Exécuter le programme. En utiliser le Menu Run de la barre de la barre de Menu pour exécuter le programme de nouveau.

## 2. Compilation séparée

Le but de cette partie est d’écrire un programme en C++ lisant un entier sur l’entrée clavier et écrivant sur la sortie écran écrivant sa parité.

**Exercice 1** : Créer quatre fichiers dans le répertoire tp1 : ***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) donnant ***true*** si la parité est paire et ***false*** sinon.

Le fichier  ***Parité.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 Parité.

Le fichier ***Paire.h*** contiendra la déclaration des flots d’entrées/sorties et l’inclusion de ***Parite.h***.

**Exercice 2** : Créer les fichiers et ajouter 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

**Exercice 3** : Générer le fichier exécutable. Exécuter le.

## 3. Création d’une classe date

Vous devez définir une classe simple permettant la gestion d’une date. Le cahier des charges impose que votre classe permette l’affichage et l’incrémentation d’une date. Vous vous inspirerez de la classe Data.java disponible dans le cours de Programmation Objet (cf. Annexe).

**Exercice 1** : Donnez la définition de la classe sous forme d’un fichier d’entête Date.h.

**Exercices 2** : Ecrire le code correspondant au constructeur. Tester le dans un programme de test.

**Exercices 3** : Ecrire le code correspondant à la méthode d’affichage. Tester le.

**Exercice 4** : Ecrire le code correspondant à la méthode d’incrémentation. Tester le.

**Exercice 5** : Ajouter une méthode permettant la saisie d’une date à la définition de la classe et écrire 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 Incrementer () {

// 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 annees 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