**SERIE D’EXERCICES PROGRAMMATION POO JAVA 2017/2018**

### Exercice 1 : Conversion de températures

En utilisant la formule C = (5/9)(F-32) écrire en langage JAVA un programme Degres.java qui lit une température exprimée en degrés Fahrenheit et affiche sa valeur en degrés centigrades ou degrés Celsius.

***Solution***

/\*\*

 \* Degres.java

 \*

 \* lit une tempÃ©rature exprimÃ©e en degrÃ©s Fahrenheit et affiche sa valeur en

 \* degrÃ©s Celsius.

public class Degres {

 public static void main(String[] args) {

 Scanner sc = new Scanner(System.in);

 System.out.print("entrez une tempÃ©rature (en Â°F) : ");

 double tempF = sc.nextDouble();

 double tempC = (5.0 / 9.0) \* (tempF - 32); // attention ! : il faut

 // bien Ã©crire 5.0/9.0

 // et non pas 5/9 car sinon on effectue une

 // division entiÃ¨re et le rÃ©sultat de l'expression

 // est alors toujours Ã©gal Ã  0

 System.out.println("cette tempÃ©rature Ã©quivaut Ã  Â°C : " + tempC);

 }

} // Degres

### Exercice 2 : Conversion de durées

Ecrire un programme Hjms.java en langage JAVA qui pour un nombre de secondes donné calcule et affiche son équivalent en nombre de jours, d'heures, de minutes et de secondes.

***Solution***

import java.util.Scanner;

public class Hjms {

 public static void main(String[] args) {

 int duree;

 Scanner sc = new Scanner(System.in);

 System.out.print("entrez une durÃ©e (en sec.) : ");

 duree = sc.nextInt();

 int nbJours = duree / (24 \* 60 \* 60); //l'opÃ©rateur '/' est l'opÃ©rateur de division entiÃ¨re

 duree = duree % (24 \* 60 \* 60); //l'opÃ©rateur '%' est l'opÃ©rateur modulo, reste de la

 //division entiÃ¨re

 int nbHeures = duree / (3600);

 duree = duree % 3600;

 int nbMin = duree / 60;

 int nbSec = duree % 60;

 System.out.println("Cette duree equivaut Ã  ");

 System.out.println(nbJours + " jours " + nbHeures + " heures "

 + nbMin + " minutes " + nbSec + "secondes");

 }

} // Hjms

### Exercice 3 : Tester si un nombre est premier

Un nombre est n premier s’il a seulement deux diviseurs : 1 et n. Ecrire un programme Premier.java qui permet de tester si un nombre introduit par l'utilisateur est premier ou non.

***Solution***

import java.util.Scanner;

public class Premier {

 public static void main(String[] args) {

 Scanner sc = new Scanner(System.in);

 int nb; // le nombre à tester

 int div; // un diviseur potentiel de nb

 boolean aUnDiviseur; // true si nb % div == 0 (nb divisible par div)

 boolean encore; // pour pouvoir tester un autre nombre

 do {

 System.out.print("entrez un entier : ");

 nb = sc.nextInt();

 div = 2;

 aUnDiviseur = false;

 while (!aUnDiviseur && div \* div <= nb) {

 if (nb % div == 0 && div != nb)

 aUnDiviseur = true;

 else

 div++;

 }

 if (aUnDiviseur)

 System.out.printf("%d n'est pas premier. Il est divisible par %d \n",nb,div);

 else

 System.out.printf("%d est premier\n", nb);

 System.out.println("Continuer O/N : ");

 encore = "O".equals(sc.next().toUpperCase());

 } while (encore);

 }}

### Exercice 4 : Moyenne olympique

Ecrire un programme qui lit au clavier une suite de nombres réels positifs ou nuls (correspondant à des notes), terminée par la valeur -1, et calcule la moyenne olympique de ces valeurs, c'est à dire la moyenne des notes sans prendre en compte la note la plus élevée ni la note la moins élevée.

***Solution***

import java.util.Scanner;

public class MoyenneOlympique {

 private static Scanner sc = new Scanner(System.in);

 /\*\*

 \* methode utilitaire pour la lecture d'une note.

 \* On vérifie que la note est bien dans l'intervalle 0..20 ou que

 \* c'est la valeur -1 pour terminer la saisie. Dans ce cas on vérifie

 \* que trois notes au moins ont déjà été saisies.

 \* Si la note est incorrecte, ou si -1 est rentré et qu'il n'y a pas

 \* encore au moins trois notes de saisies, on redemande la saisie de la

 \* note.

 \* @param nbNotesLues. Le nombre de notes déjà lues.

 \* @return la note lue.

 \*/

 public static double lireUneNote(int nbNotesLues) {

 double noteLue;

 boolean noteOK= false;

 do { noteOK = false;

 System.out.print("Donner un note entre 0 et 20 (-1 pour arrêter à partir de la 3ème entrée) : "); noteLue = sc.nextDouble();

 if (noteLue < -1 || noteLue > 20)

 System.out.println("Valeur incorrecte ! Donnez une note entre 0 et 20 (-1 pour arrêter)");

 else if (noteLue == -1 && nbNotesLues < 3)

 System.out.println("vous devez rentrez au moins trois notes"); else

 noteOK = true;

 } while (!noteOK);

 return noteLue;

 }

 public static void main(String[] args) {

 double total = 0.0; // le cumul des notes saisies

 int nbNotes = 0; // le nombre de notes saisies

 double max = Double.MIN\_VALUE; // la note la plus forte

 // initialisée avec la plus petite valeur codable pour un double

 double min = Double.MAX\_VALUE; // la note la plus faible

 // initialisée avec la plus grande valeur codable pour un double

 double note = lireUneNote(0);

 while (note !=-1 ){

 total += note;

 if (max <= note)

 max = note;

 if (min >= note) // on ne met pas un else if

 min = note; // car la premiere note sera

 // à la fois min et max

 nbNotes++;

 note = lireUneNote(nbNotes);

 }

 System.out.println("Nombre de notes saisies " + nbNotes);

 System.out.println("Note la plus basse: " + min);

 System.out.println("Note la plus haute: " + max);

 System.out.println("Moyenne olympique: " + (total - (min + max)) / (nbNotes - 2));

 } }