

INFO-H-200 : Programmation orientée objet

TP 1 - Python à Java

Professeur : Hugues Bersini

<http://cs.ulb.ac.be/public/teaching/infoh200>

Année académique 2017-2018

De Python à Java, la syntaxe

Ecrivez d'abord un programme Python comportant une fonction réalisant l'opération demandée pour chaque exercice. Poursuivez ensuite en réalisant le code Java équivalent !

Exercice 1.0

Ecrire un programme qui affiche tous les nombres pairs entre deux bornes reçues en arguments du programme.

Exercice 1.1

Ecrire un programme qui convertisse en radians une valeur d'angle exprimée en degré, entrée à la main par l'utilisateur, et qui l'affiche.

Exercice 1.2

Ecrire un programme qui teste si le `String` reçu en argument est un palindrome (radar, sos).

Exercice 1.3

Ecrire un programme qui calcule et affiche la moyenne d'un tableau d'entiers déterminé (c'est à dire construit explicitement dans le `main()`).

Exercice 1.4

Ecrire un programme qui trouve le minimum d'un tableau d'entiers passés en paramètre. Réaliser deux fonctions : une première appelée "minFor" qui effectuera cette recherche de minimum en employant une boucle 'for' et une seconde appelée "minWhile" qui effectuera la recherche à l'aide d'une boucle 'while'.

Exercice 1.5

Ecrire un programme qui affiche le minimum d'un tableau d'entiers déterminé. Utiliser l'instruction *for-each*.

INFO-H-200 - Programmation orientée objet

TP 1

Corrections

Exercice 1.0

Solution en Python :

```
def showEven(min,max):
    for i in range(min,max):
        if i%2 == 0 and i != 0:
            print(str(i) + " ",end="\n")

if __name__ == "__main__":
    min = 0
    max = 20
    showEven(min,max)
```

Solution en Java :

```
public class Main {

    public static void showEven(int min, int max){
        for (int i = min; i <= max; i+=1){
            if(i%2 == 0 && i != 0){
                System.out.print(i+" ");
            }
        }
        System.out.println();
    }

    public static void main (String args[]) {
        int min = 0;
        int max = 20;
        showEven(min,max);
    }
}
```

Exercice 1.1

Solution en Python :

```
import math

def degToGrad(val):
    return val * math.pi / 180

if __name__ == "__main__":
    deg = float(input("Entrez l'angle en degres : "))
    grad = degToGrad(deg)
    print("L'angle en gradiants est : " + str(grad))
```

Solution en Java :

```
import java.lang.Math;
import java.util.Scanner;

public class Main {

    public static double degToGrad(double val){
        return val * Math.PI / 180;
    }

    public static void main (String args[]) {
        Scanner myScan = new Scanner(System.in);
        System.out.println("Entrez l'angle en degres : ");
        double deg = myScan.nextDouble();
        myScan.close();
        double grad = degToGrad(deg);
        System.out.println("L'angle en gradiants est : " +
            String.valueOf(grad));
    }
}
```

Exercice 1.2

Solution en Python :

```
def isPalindrome(word):
    length = len(word)
    isPal = True
    for i in range(int(len(word)/2)):
        if word[i] != word[-i-1]:
            isPal = False
    return isPal

if __name__ == "__main__":
    pal = "radar"
    isPal = isPalindrome(pal)
    if isPal:
        print(str(pal) + " est un palindrome.")
    else:
        print(str(pal) + " n'est pas un palindrome.")
```

Solution en Java :

```
public class Main {

    public static boolean isPalindrome(String word){
        int length = word.length();
        boolean isPal = true;
        for (int i=0; isPal && i<length/2; i++){
            if(word.charAt(i) != word.charAt(length-i-1)){
                isPal = false;
            }
        }
        return isPal;
    }

    public static void main (String args[]) {
        String pal = "radar";
        boolean isPal = isPalindrome(pal);

        if(isPal){
            System.out.println(pal + " est un palindrome.");
        }else{
            System.out.println(pal + " n'est pas un palindrome.");
        }
    }
}
```

Exercice 1.3

Solution en Python :

```
def average(tab):
    sum = 0
    for i in range(len(tab)):
        sum += tab[i]
    return sum/len(tab)

if __name__ == "__main__":
    tab = [10,1,5,2]
    moyenne = average(tab)
    print("La moyenne est : " + str(moyenne))
```

```
import java.util.ArrayList;
import java.util.Arrays;

public class Main {
    public static double averageTab(int[] tab){
        double sum = 0.0;
        for(int i = 0; i < tab.length; i++){
            sum += tab[i];
        }
        return sum/tab.length;
    }

    public static double averageArrayList(ArrayList<Integer> list){
        double sum = 0.0;
        for(int i = 0; i < list.size(); i++){
            sum += list.get(i);
        }
        return sum/list.size();
    }

    public static void main (String args[]) {
        int[] tab = {10, 1, 5, 2};
        double moyenne = averageTab(tab);
        System.out.println("La moyenne est : " + String.valueOf(moyenne));

        // ArrayList ne peut contenir que des objets
        ArrayList<Integer> list = new ArrayList<Integer>(Arrays.asList(10, 1,
            5, 2));
        double moyen = averageArrayList(list);
        System.out.println("La moyenne est : " + String.valueOf(moyen));
    }
}
```

Solution en Java :

Exercice 1.4

Solution en Python :

```
def minFor(tab):
    minimum = tab[0]
    for i in range(1,len(tab)):
        if tab[i] < minimum:
            minimum = tab[i]
    return minimum

def minWhile(tab):
    minimum = tab[0]
    i = 0
    while i < len(tab):
        if tab[i] < minimum:
            minimum = tab[i]
        i += 1
    return minimum

if __name__ == "__main__":
    tab = [2,5,100,4,1]
    minimum = minFor(tab)
    print("Le minimum est : " + str(minimum))
    minimum = minWhile(tab)
    print("Le minimum est : " + str(minimum))
```

Solution en Java :

```
public class Main {
    public static int minFor(Integer tab[]){
        int minimum = tab[0];
        for (int i = 0; i < tab.length; i++){
            if (tab[i] < minimum){
                minimum = tab[i];
            }
        }
        return minimum;
    }

    public static int minWhile(Integer tab[]){
        int minimum = tab[0];
        int i = 0;
        while (i < tab.length) {
            if (tab[i] < minimum){
                minimum = tab[i];
            }
            i++;
        }
        return minimum;
    }

    public static void main (String args[]) {
        int[] tab = {2,5,100,4,1};
        int minimum = minFor(tab);
        System.out.println("Le minimum est : "+ String.valueOf(minimum));
        minimum = minWhile(tab);
        System.out.println("Le minimum est : "+ String.valueOf(minimum));
    }
}
```

Exercice 1.5

Solution en Python :

```
def minForEach(tab):
    minimum = tab[0]
    for elem in tab:
        if elem < minimum:
            minimum = elem
    return minimum

if __name__ == "__main__":
    tab = [2,5,100,4,1]
    minimum = minForEach(tab)
    print("Le minimum est : " + str(minimum))
```

Solution en Java :

```
public class Main {
    public static int minForEach(int tab[]){
        int minimum = tab[0];
        for(int item : tab){
            if(item < minimum){
                minimum = item;
            }
        }
        return minimum;
    }

    public static void main (String args[]) {
        int[] tab = {10, 1, 5, 2};
        int minimum = minForEach(tab);
        System.out.println("Le minimum est : " + minimum);
    }
}
```