Fiche TD 1 : Introduction à Java

Question de cours (facultative - pour révision) :

- Quelles sont les principales différences entre les modèles objet (comme le java) et procédurale (comme le langage C) ?
- Quels sont les avantages de la programmation OO ?

Exercice 1 :

Ecrire le programme Java permettant de calculer la somme des 100 premiers entiers et le produit des 10 premiers entiers avec une seule classe (un seul fichier Java).

Exercice 2 :

Reprendre l'exercice 1, mais cette fois-ci en séparant les calculs (somme et produit) dans deux méthodes différentes.

Exercice 3 :

Reprendre l'exercice 1, mais cette fois-ci avec deux classes définies dans deux fichiers Java indépendants. Les méthodes dans une classe et l'affichage dans la deuxième.

Exercice 4 :

Le programme Java suivant permet de lire un entier au clavier :

```
import java.io.*;
public class LectureEntier {
        public static void main (String [] args) {
        // On commence par déclarer un objet lecteur sur le clavier
        // Le clavier est associé à la variable System.in
        // Un InputStreamReader (lecteur sur un flux) est créé dessus
        // Puis on ajoute une couche BufferedReader nécessaire pour lire des informations par ligne
        BufferedReader lecteurClavier=new BufferedReader(new InputStreamReader(System.in));
        int valeur = 0;
        String chaineLue;
                try { // Lecture d'une ligne au clavier
                System.out.print("Entrer un entier : ");
                chaineLue = lecteurClavier.readLine();
                // Conversion de la chaîne en entier
                valeur = Integer.parseInt(chaineLue);
                }
                catch (Exception e) {
                System.out.println("Erreur d'E/S " + e.getMessage());
                }
        System.out.println("La valeur est : " + valeur);
        }
}
```

Essayez d'implémenter la classe Readln décrite dans le chapitre des opérations d'entrée sortie.
Dans le reste des exercices vous pouvez utiliser les fonctions de la classe Readln pour la lecture d'un élément de type entier, caractère, etc., sans avoir besoin d'implémenter la partie de lecture de cet élément.

Exercice 5 :
Ecrire un programme Java permettant de calculer la valeur absolue d'un nombre réel x. La valeur absolue du nombre réel x est le nombre réel |x| : |x| = x , si x >= 0 et |x| = -x si x < 0.

Exercice 6 :
On souhaite écrire un programme Java de calcul des n premiers nombres parfaits. Un nombre est dit parfait s'il est égal à la somme de ses diviseurs, 1 compris.
Par exemple : 6 = 1+2+3, est un nombre parfait.

Exercice 7 :
On souhaite écrire un programme Java de calcul et d'affichage des n premiers nombres premiers. Un nombre entier est premier s'il n'est divisible que par 1 et par lui-même. On doit utiliser des boucles while et do...while imbriquées. Par exemple : 37 est un nombre premier.

Exercice 8 :
Reprendre l'exercice précédent, mais cette fois-ci en utilisant des boucles for imbriquées.

Exercice 9 :
Ecrire un programme Java qui détermine et affiche la plus grande valeur des éléments d'un tableau. On complétera le programme suivant :

```
public class Valeurmax {
        public static void main(String[] args){
        // Définition d'un tableau de 10 entiers
        int [] Tableau = {5, 3, 12, 4, 7, 9, 1, 8, 19, 2};
        }
}
```

Exercice 10 :
Ecrire un programme Java permettant de trier, ensuite afficher un tableau d'entiers contenant les éléments : {8, 5, 7, 9, 2, 1, 12, 6}. Utiliser l'algorithme de tri par sélection.

Exercice 11 :
Ecrire un programme en java qui permet de saisir l'heure courante sous la forme (h :m :s) et affiche cette heure incrémentée d'une seconde

Exercice 12 :
Ecrire une classe « Factorielle » qui permet de calculer la factorielle d'un entier passé en paramètre

Exercice 13 :

Ecrire une classe « EquationSecDegre » qui permet de résoudre une équation de second degré de ma forme $A x^2 + B x + C = 0$ avec A,B,C

Exercice 14 :

Reprendre l'exercice 9, mais cette fois-ci il faut définir une classe TriEntiers permettant de trier des entiers stockés dans un tableau. L'en-tête de la méthode de tri est le suivant :

```
public void TriTableauEntiers(int [] Tableau)
```

Tester votre programme à l'aide du programme TestTriEntiers suivant :

```
public class TestTriEntier {
        public static void main(String[] args){
        int [] Tableau = {12, 3, 1, 4, 9, 5, 2, 8, 3, 6};
        TriEntiers Tri = new TriEntiers();
        Tri.TriTableauEntiers(Tableau);
                for (int i = 0; i < Tableau.length; i++)
                System.out.println(Tableau[i] + " ");
        }
}
```

Exercice 15 :

Définir en Java une classe TriStrings contenant une fonction permettant de trier des chaînes de caractères. On utilisera la méthode compareTo de la classe String pour comparer deux chaînes de caractères. int compareTo(String s) renvoie un résultat négatif si la chaîne appelante est inférieure à s (le paramètre), 0 si elles sont égales, et un résultat positif sinon.

L'en-tête de la fonction de tri est le suivant :

```
public void TriTableauChaines(String [] Tableau)
```

Tester votre programme à l'aide du programme TestTriChaines suivant :

```
public class TestTriChaines {
        public static void main(String[] args){
                String [] Tableau = {"un", "deux", "trois", "quatre", "cinq"};
                TriChaines Tri = new TriChaines();
                Tri.TriTableauChaines(Tableau);
                        for (int i = 0; i < Tableau.length; i++)
                        System.out.println(Tableau[i] + " ");
        }
}
```

Exercice 16 :

Implémenter une classe « Tableau » qui permet de :

- Déclarer deux tableau t1 et t2 d'entiers et créer pour t1 et t2 des tableaux de 5 entiers.
- Initialiser t2 avec les valeurs : 1, 567, -23, 78 et 4.
- Comment faire la déclaration, la création et l'initialisation d'un tableau t3 en une seule instruction ?
- Ecrire une méthode « minTab » pour trouver le plus petit élément d'un tableau.
- Ecrire une méthode « maxTab » pour trouver le plus grand élément d'un tableau.
- Ecrire une méthode « sommeTab» pour effectuer le somme d'un tableau et afficher le résultat.
- Ecrire une méthode «moyennePositive» et «moyenneNégative» qui permettent de calculer respectivement la moyenne des entiers positifs et négatifs.

Exercice 17 :

Ecrire un programme (classe UDC) permettant de saisir un entier positif et inférieur à 999, ensuite il affiche les chiffres des unités, dizaines et des centaines.

Exercice 18 :

Ecrire un programme qui permet d'inverser la chaine de caractères passée en paramètre

Exercice 19 :

On appelle palindrome un mot pouvant se lire indifféremment dans les deux sens : par exemple LAVAL

- Ecrire une fonction « estPalindrome » vérifiant si un mot est un palindrome
- Ecrire une fonction « estPalindrome » prenant en argument un tableau d'entier et renvoyant un booléen indiquant si le tableau est un palindrome. Ecrire également la méthode main permettant de tester cette fonction
- En utilisant la méthode charAt, écrire une autre fonction prenant cette fois ci une String en argument et vérifiant si c'est un palindrome. La chaîne vérifiée sera récupérée par main sur la ligne de commande

Exercice 20 :

Ecrire un programme qui prend en paramètre deux arguments, le premier est un entier strictement positif (n>0), le deuxième est une chaîne de caractère (maChaine). Ce programme permet d'afficher n fois la chaine de caractère (maChaine)