Série de TD N° 1 Notions de base / Variables / Constantes Fonctions d'entrée-sortie	Licence Professionnelle Développement Informatique POO en Java
--	---

Syntaxe de base de java :

Exercice 1

Dans la fonction main d'une classe **Equation1**, ajouter les instructions qui font la résolution d'une équation du premier degré de la forme $ax + b = 0$.

Solution :

```

import java.util.Scanner;

public class Equation1 {

    public static void main(String[] args) {

        double a,b;
        Scanner sc = new Scanner(System.in);

        System.out.print("Donner a : ");
        a = sc.nextDouble();

        System.out.print("Donner b :");
        b = sc.nextDouble();

        if(a!=0) {
            System.out.println("X=" + -b/a);
        } else if(b==0) {
            System.out.println("Infinité de solutions");
        } else {
            System.out.println("Pas de solution");
        }
        sc.close();
    }
}

```

Exercice 2

Dans la fonction main d'une classe **Equation2**, ajouter les instructions qui font la résolution dans $\mathbb{R}$ d'une équation du second degré de la forme $ax^2+bx+c=0$.

Solution :

```

import java.util.Scanner;

public class Equation2 {

    public static void main(String[] args) {

        double a,b,c,delta;
        Scanner sc = new Scanner(System.in);

        System.out.print("Donner a : ");
        a = sc.nextDouble();

        System.out.print("Donner b :");
        b = sc.nextDouble();

        System.out.print("Donner c :");
        c = sc.nextDouble();

        delta = b*b -4*a*c;

        if(delta > 0) {
            System.out.println("X1 = " + (-b+Math.sqrt(delta))/(2*a));
            System.out.println("X2 = " + (-b-Math.sqrt(delta))/(2*a));
        } else if(delta==0) {
            System.out.println("X2 = " + (-b)/(2*a));
        } else {
            System.out.println("Pas de solutions réelle");
        }
        sc.close();
    }
}

```

Exercice 3

Ecrire un programme qui affiche la table de multiplication d'un entier entré par l'utilisateur.

Solution :

```

package org;

import java.util.Iterator;
import java.util.Scanner;

public class Multiplication {

    public static void main(String[] args) {

```

```

Scanner sc = new Scanner(System.in);
int n;

System.out.println("Donner un nombre :");
n = sc.nextInt();

for (int i = 1; i <=10 ; i++) {
    System.out.println(n+" x "+i+" = "+n*i);
}

sc.close();

}

}

```

Exercice 4

Ecrire un programme qui lit **n** nombres réels et affiche leur moyenne.

Solution :

```

import java.util.Scanner;

public class Moyenne {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int a ,n, s=0;

        System.out.println("Donner le nombre des valeurs : ");
        n = sc.nextInt();

        for(int i=1; i<=n ; i++ ) {
            System.out.print("Donner la valeur num "+i+": ");
            a = sc.nextInt();
            s = s + a;
        }

        System.out.println("La moyenne est "+s/n);

        sc.close();
    }
}

```

```
}  
  
}
```

Exercice 5

Ecrire un programme qui détermine si un nombre entier >2 entré par l'utilisateur est parfait ou non.

Un nombre est parfait s'il est égal à la somme de ses diviseurs strictes.

Exemples des nombres parfaits :

$$6 = 1 + 2 + 3$$

$$28 = 1 + 2 + 4 + 7 + 14$$

Solution :

```
import java.util.Scanner;  
  
public class Parfait {  
  
    public static void main(String[] args) {  
  
        int n, s;  
        Scanner sc = new Scanner(System.in);  
  
        System.out.print("Donner un nombre : ");  
        n = sc.nextInt();  
  
        s = 0;  
        for (int i = 1; i < n; i++) {  
            if(n%i==0){  
                s = s + i;  
            }  
        }  
  
        if(s==n){  
            System.out.println(n+" est parfait");  
        } else {  
            System.out.println(n+" n'est parfait");  
        }  
  
        sc.close();  
    }  
}
```

Une classe avec plusieurs fonctions

Exercice 6

Dans une classe **Mathematique**, déclarer une fonction statique « **public static int factoriel(int n)** » qui calcul et return le factoriel d'un entier n supposé positif.

Ajouter une fonction statique « **public static int combinaison(int n, int p)** » qui calcul C_n^p .

Dans la fonction main calculer : C_8^5

Solution :

```
import java.util.Scanner;

public class Exercice6 {

    public static int factoriel(int n) {
        int f = 1;
        for(int i=1; i<=n; i++) {
            f = f * i;
        }
        return f;
    }

    public static int combinaison(int n, int p) {
        int nf,pf,npf;
        nf = factoriel(n);
        pf = factoriel(p);
        npf = factoriel(n-p);
        return nf/(pf*npf);
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int n,p;

        System.out.println("n=");
        n = sc.nextInt();
        System.out.println("p=");
        p = sc.nextInt();

        System.out.println("comb="+combinaison(n, p));

        sc.close();

    }
}
```

```
}
```

Exercice 7

Un nombre narcissique (ou nombre d'Armstrong de première espèce) est un entier naturel **n** non nul qui est égal à la somme des puissances p-ièmes de ses chiffres en base dix, où p désigne le nombre de chiffres de n :

Exemples de nombres narcissiques :

$$153 = 1^3 + 5^3 + 3^3$$

$$548834 = 5^6 + 4^6 + 8^6 + 8^6 + 3^6 + 4^6$$

Dans une classe Narcissique :

1. Ajouter une fonction statique « **public static int nbreChiffres(int n)** » qui prend en argument un nombre entier naturel n et renvoie **le nombre de chiffre de n**.
Exemple : **nbre_chiffres(153)** renvoie **3**
2. Ecrire en une fonction statique « **public static int estArmstrong(int n)** » qui prend en argument un nombre entier naturel n et renvoie **True** si n est un nombre d'Armstrong ou **False** si non.
3. Dans la fonction **main** vérifier si **9474** est un nombre narcissique.

Solution :

```
package org;

import java.util.Iterator;
import java.util.Scanner;

public class Armstrong {

    public static int nbreChiffres(int n) {
        String s = n + "";
        return s.length();
    }

    public static boolean estAmstrong(int x) {
        int c = nbre_chiffres(x);
        int s = 0, r;
        int n = x;
        while (x!=0) {
            r = x%10;
            x = x/10;
            s = s + (int)Math.pow(r, c);
        }
    }
}
```

```

        return s==n;
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        int a;
        boolean res;

        System.out.println("Donner le nombre à vérifier :");
        a = sc.nextInt();

        res = amstrong(a);
        if(res) {
            System.out.println("Le nombre est d'Armstrong");
        } else {
            System.out.println("Les nombre n'est pas Armstrong");
        }

        sc.close();

    }
}

```

Les tableaux

Exercice 8

Dans une classe **Statistique** :

1. Ajouter une fonction statique « **public static int somme(int T[])** » qui calcule la somme des éléments du tableau T.
2. Ajouter une fonction statique « **public static moyenne(int T[])** » qui calcule et retourne la moyenne des éléments du tableau T passé en paramètre.
3. Ajouter une fonction main à la classe, dans cette fonction déclarer un tableau **A** de 10 entiers et initialiser le par des valeurs de votre choix, puis calculer la somme et la moyenne des éléments de ce tableau.

Correction :

```

public class Statistique {

```

```

public static int somme(int T[]) {
    int somme = 0;
    for (int i = 0; i < T.length; i++) {
        somme += T[i];
    }
    return somme;
}

public static double moyenne(int T[]) {
    int somme = somme(T);
    return (double) somme / T.length;
}

public static void main(String[] args) {
    int[] A = {5, 8, 12, 7, 10, 3, 6, 15, 9, 4};

    int sommeA = somme(A);
    double moyenneA = moyenne(A);

    System.out.println("Somme des éléments de A : " + sommeA);
    System.out.println("Moyenne des éléments de A : " + moyenneA);
}
}

```

Exercice 9

Dans la fonction main d'une classe **Conjugaison**, ajouter les instructions qui permet de conjuguer les verbes de premier groupe au présent de l'indicatif.

Etapes :

1. Déclarer un tableau initialisé par les sujets.
2. Déclarer un tableau initialisé par les terminaisons de chaque sujet.
3. Calculer le radical du verbe, exemple : Le radical de **parler** est **parl** (Enlever le dernier er)
4. Afficher chaque sujet suivi du radical concaténé au convenable terminaison.

Correction :

```

import java.util.Scanner;

public class Conjugaison {

    public static void main(String[] args) {

        String[] sujets = {"Je", "Tu", "Il/Elle", "Nous", "Vous", "Ils/Elles"};

        String[] terminaisons = {"e", "es", "e", "ons", "ez", "ent"};
    }
}

```

```

Scanner scanner = new Scanner(System.in);
String verbe = scanner.nextLine();

String radical = verbe.substring(0, verbe.length() - 2);

System.out.println("Conjugaison du verbe au présent de l'indicatif :");
for (int i = 0; i < sujets.length; i++) {
    System.out.println(sujets[i] + " " + radical + terminaisons[i]);
}

scanner.close();
}
}

```

Exercice 10

Ecrire un programme qui calcule la somme de deux polynômes.

Correction :

```

import java.util.Arrays;

public class Polynome {

    public static int[] sommePolynomes(int[] poly1, int[] poly2) {
        int tailleMax = Math.max(poly1.length, poly2.length);

        int[] resultat = new int[tailleMax];

        for (int i = 0; i < tailleMax; i++) {
            int coeff1 = (i < poly1.length) ? poly1[i] : 0;
            int coeff2 = (i < poly2.length) ? poly2[i] : 0;
            resultat[i] = coeff1 + coeff2;
        }

        return resultat;
    }

    public static void main(String[] args) {
        int[] poly1 = {5, 0, 3}; // Représente 5 + 0x + 3x^2
        int[] poly2 = {2, 4, 1, 7}; // Représente 2 + 4x + 1x^2 + 7x^3

        int[] resultat = sommePolynomes(poly1, poly2);

        System.out.println("Somme des polynômes : " +
            Arrays.toString(resultat));
    }
}

```

Exercice 11

Ecrire un programme qui calcule le produit de deux polynômes.

Correction :

```
import java.util.Arrays;

public class Polynome {

    public static int[] produitPolynomes(int[] poly1, int[] poly2) {
        int[] resultat = new int[poly1.length + poly2.length - 1];

        for (int i = 0; i < poly1.length; i++) {
            for (int j = 0; j < poly2.length; j++) {
                resultat[i + j] += poly1[i] * poly2[j];
            }
        }

        return resultat;
    }

    public static void main(String[] args) {
        int[] poly1 = {3, 2, 5}; // Représente 3 + 2x + 5x^2
        int[] poly2 = {5, 1, 2}; // Représente 5 + x + 2x^2

        int[] resultat = produitPolynomes(poly1, poly2);

        System.out.println("Produit des polynômes : " +
            Arrays.toString(resultat));
    }
}
```

Exercice 12

Ecrire un programme dans une classe appelée Tri, qui permet de lire n nombres entiers puis on tri les éléments du tableau dans un ordre croissant avant de l'afficher.

Correction :

```
import java.util.Scanner;
import java.util.Arrays;

public class Tri {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Entrez le nombre d'éléments à trier : ");
        int n = scanner.nextInt();
```

```
int[] nombres = new int[n];

System.out.println("Entrez " + n + " nombres entiers :");
for (int i = 0; i < n; i++) {
    System.out.print("Nombre " + (i + 1) + ": ");
    nombres[i] = scanner.nextInt();
}

Arrays.sort(nombres);

System.out.println("Les nombres triés dans l'ordre croissant : "
+ Arrays.toString(nombres));

    scanner.close();
}
}
```