

Contrôle d'informatique no 2

Durée : 1 heure 45'

Nom :

Groupe :

Prénom :

No	1	2
Total points	60 points (28 + 32)	60 points (30 + 30)

Remarque générale : toutes les questions qui suivent se réfèrent au langage de programmation Java (à partir du JDK 5.0) et les réponses doivent être rédigées à l'encre et d'une manière propre sur ces feuilles agrafées.

Sujet no 1.

1.1 Préciser les messages qui seront affichés à l'écran suite à l'exécution du projet contenant la classe suivante.

```
package cms_ctr2;

public class CP_Ctr2Exo1_1 {
    public static void main(String[] args) {
        int i;
        for(i=10; i>=0; i=i-2){
            if(i%3 == 1)
            {
                System.out.println("i = " + ++i + ".");
            } else if(i%3 == 2)
            {
                System.out.println("i = " + i++ + ".");
            } else
            {
                System.out.println("i = " + i + ".");
                continue;
            }
            System.out.println("Fin d'un tour de boucle !");
        }
        System.out.println("A la fin, i = " + i + ".");
        System.out.println("Au revoir !");
    } //fin de la méthode main
} //fin de la classe principale
```

This image shows a full page of primary-ruled paper. It features approximately 28 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting practice. The paper is otherwise blank, with no margins, text, or other markings.

1.2 Préciser les messages qui seront affichés à l'écran suite à l'exécution du projet contenant les deux classes suivantes (qui appartiennent à un même package).

```
package cms_ctr2;

public class Auxiliaire {
    private int nb = 10;

    public int getNb() {
        System.out.println("Un getter !");
        return nb;
    }

    public void setNb(int nb) {
        System.out.println("Un setter !");
        this.nb = nb;
    }

    Auxiliaire(int nb) {
        System.out.println("Un Auxiliaire avec un arg !");
        setNb(nb);
    }

    Auxiliaire() {
        this(8);
        System.out.println("Un Auxiliaire sans arg !");
    }

    public void jeTeTiens() {
        --nb;
        switch(nb)
        {
            case 1 : System.out.println("Je dis : nb="+nb+" !");
            case 2 :
            case 3 : System.out.println("Je dis : nb="+nb+" !");
            case 4 :
            case 5 : System.out.println("Je dis : nb="+nb+" !");
            default : tuMeTiens();
        }
        nb--;
    }

    public void tuMeTiens() {
        --nb;
        if(nb > 3)
        {
            System.out.println("Tu dis : nb=" + nb + " !");
            jeTeTiens();
        } else
        {
            System.out.println("Pas de barbichette !");
        }
        nb--;
    }
}

//fin de la classe Auxiliaire
```


[illegible]

Sujet no 2.

Le but de cet exercice est de réaliser une application autonome interactive qui aide l'utilisateur à calculer les solutions réelles d'une équation du deuxième degré à coefficients réels. Cette application doit correspondre à un projet muni d'un package appelé **cms_ctr2** et qui contient deux classes publiques, définies dans deux fichiers à part et appelées **Solution** et, respectivement, **CP**. Par la suite, aux points **2.1** et **2.2**, on vous demande d'écrire le code complet de ces deux classes, en respectant les consignes précisées. Vous pouvez éventuellement répondre au point **2.2** en supposant le point **2.1** résolu correctement.

2.1 Une instance de la classe **Solution** est un objet de type **Solution** qui regroupe (ou encapsule) des informations sur les deux solutions d'une équation du deuxième degré. De plus, la classe **Solution** est munie des méthodes permettant, par exemple, de calculer et d'afficher ces informations.

Ecrire le code complet de la classe publique **Solution** qui appartient au package **cms_ctr2** et qui définit :

- a) un champ nommé **sol_1** de type réel, déclaré sans modificateur d'accès et sans valeur initiale explicite et qui sert à stocker, soit la valeur de la première solution réelle de l'équation, soit la valeur spéciale "Not A Number" si l'équation n'a pas de solution réelle ;
- b) un champ nommé **sol_2** de type réel, déclaré privé et sans valeur initiale explicite et qui sert à stocker, soit la valeur de la deuxième solution réelle de l'équation, soit la valeur spéciale "Not A Number" si l'équation a deux solutions réelles confondues ou si l'équation n'a pas de solution réelle ;
- c) une méthode publique "getter" nommée **getSol_2** qui retourne (sans aucune vérification) la valeur du champ privé **sol_2** ;
- d) une méthode publique "setter" nommée **setSol_2** qui permet la modification de la valeur du champ privé **sol_2** ; cette méthode a un seul argument de type réel et doit respecter les consignes suivantes :
 - i. si la valeur du champ **sol_1** n'est pas la valeur spéciale "Not A Number", on stocke dans le champ **sol_2** la valeur de l'argument de la méthode ;
 - ii. autrement, on stocke dans le champ **sol_2** la valeur spéciale "Not A Number" ;
- e) un constructeur public (surchargé) avec deux arguments de type réel et qui doit respecter les consignes suivantes :

- i. la valeur du premier argument est stockée dans le champ **sol_1** ;
 - ii. la valeur du deuxième argument "est stockée" dans le champ **sol_2** par un appel à la méthode **setSol_2** ;
- f) un constructeur public (surchargé) avec un seul argument et qui doit respecter les consignes suivantes :
 - i. la valeur du premier argument est stockée dans le champ **sol_1** ;
 - ii. la valeur spéciale "Not A Number" est stockée dans le champ **sol_2** ;
- g) un constructeur public (surchargé) sans argument et qui doit respecter les consignes suivantes :
 - i. la valeur spéciale "Not A Number" est stockée dans le champ **sol_1** ;
 - ii. la valeur spéciale "Not A Number" est stockée dans le champ **sol_2** ;
- h) une méthode publique et statique nommée **calculerSolutions** dont le fonctionnement sera détaillé ci-dessous ;
- i) une méthode publique (d'instance) nommée **afficher**, qui ne retourne pas de résultat et qui n'a pas d'argument ; elle doit respecter les consignes suivantes (voir aussi l'exemple d'affichage à l'exécution présenté plus loin) :
 - i. si le champ **sol_1** et le champ **sol_2** ont tous les deux la valeur spéciale "Not A Number", on affiche à l'écran le message "*Pas de solution réelle !*" ;
 - ii. autrement si (seulement) le champ **sol_2** a la valeur spéciale "Not A Number", on affiche à l'écran le message "*Deux solutions réelles confondues :* " suivi, sur la même ligne, par la valeur de la solution (qui est stockée dans le champ **sol_1**) ;
 - iii. autrement, on affiche à l'écran le message "*Deux solutions réelles distinctes :* " suivi, sur la même ligne, par les valeurs des deux solutions (qui sont stockées, respectivement, dans les champs **sol_1** et **sol_2**).

La méthode publique et statique **calculerSolutions** :

- ne retourne pas de résultat ;
- a 4 arguments (formels ou muets) :
 - i. les premiers 3 arguments sont de type réel et correspondent aux 3 coefficients d'une équation du deuxième degré écrite sous la forme $ax^2 + bx + c = 0$ (c'est-à-dire, en ordre, le coefficient a du x au carré, le coefficient b du x et le terme libre c) ;

- ii. le quatrième (le dernier) argument est de type **Solution** et représente la référence vers un objet **Solution** qui existe déjà et qui va être modifié pour stocker les informations concernant les solutions de l'équation mentionnée au point i. ci-dessus ;
- si la valeur du premier argument (correspondant au coefficient a) est zéro, la méthode affiche le message "*Ce n'est pas une équation du deuxième degré !*" (et prend fin) ;
- autrement, on calcule la valeur du discriminant (correspondant à l'expression $b^2 - 4ac$) ;
- si la valeur du discriminant est négative, on stocke dans les champs **sol_1** et **sol_2** de l'objet référencé par le quatrième argument de la méthode la valeur spéciale "Not A Number" (et la méthode prend fin) ;
- si la valeur du discriminant est nulle, on stocke dans le champ **sol_1** de l'objet référencé par le quatrième argument de la méthode la valeur de la solution (c'est-à-dire $-b/2a$) et dans le champ **sol_2** du même objet la valeur spéciale "Not A Number" (et la méthode prend fin) ;
- si la valeur du discriminant est strictement positive, on stocke dans les champs **sol_1** et **sol_2** de l'objet référencé par le quatrième argument de la méthode les deux solutions, à savoir les valeurs $(-b + \sqrt{b^2 - 4ac})/2a$ et $(-b - \sqrt{b^2 - 4ac})/2a$ (et la méthode prend fin).

Indications :

- la valeur spéciale "Not A Number" est disponible dans la classe enveloppe (wrapper class) **Double** sous la forme d'un champ statique (et final) appelé **NaN** ;
- afin de tester si une variable réelle a la valeur spéciale "Not A Number", on utilise la méthode statique **isNaN** de la classe enveloppe **Double**, qui retourne un résultat de type booléen et qui a un seul argument de type réel ; plus précisément :
 - i. si la variable réelle passée comme argument à la méthode **isNaN** a bien la valeur spéciale "Not A Number", le résultat retourné est **true** ;
 - ii. autrement (c'est-à-dire si la variable réelle passée comme argument à la méthode **isNaN** a une valeur réelle "normale"), le résultat retourné est **false** ;
- afin de calculer une racine carrée, on peut appeler la méthode statique **sqrt** de la classe (prédéfinie) **Math**.

[illegible]

[illegible]

2.2 Ecrire le code complet de la classe publique **CP** qui appartient au package **cms_ctr2** et qui contient la méthode **main**. Cette classe réalise la partie interactive (voir aussi l'exemple d'affichage à l'exécution présenté plus loin) et utilise la classe **Solution**.

Plus précisément, dans la méthode **main** :

- a)** on déclare (et, le cas échéant, on initialise) les variables dont on a besoin pour réaliser le dialogue avec l'utilisateur ;
- b)** on déclare une variable nommée **maSol** de type **Solution** et on y stocke l'adresse d'un objet qui est créé à l'aide du constructeur sans argument de la classe **Solution** ;
- c)** on affiche (une seule fois) le message de début "*Le but du programme est ...* " (qui remplace un message plus complet du genre "Je suis votre assistant pour la résolution des équations du deuxième degré à coefficients réels ! ") ;
- d)** on demande à l'utilisateur et on stocke le coefficient du x au carré ;
- e)** si la valeur introduite par l'utilisateur est zéro, on affiche le message "*Valeur non valide!*" et on revient (autant de fois que nécessaire) au point **d** ;
- f)** on demande à l'utilisateur et on stocke le coefficient du x ;
- g)** on demande à l'utilisateur et on stocke le terme libre ;
- h)** on appelle la méthode statique **calculerSolutions** de la classe **Solution**, en lui passant comme arguments effectifs les trois coefficients indiqués par l'utilisateur et la référence **maSol** de l'objet **Solution** créé au point **b** ;
- i)** on appelle la méthode (d'instance) **afficher** de la classe **Solution** pour l'objet créé au point **b** ;
- j)** on affiche le message "*Voulez-vous continuer ? (O/N)*" (afin de savoir si l'utilisateur veut résoudre une autre équation du deuxième degré) ;
- k)** on stocke la réponse de l'utilisateur ;
- l)** si l'utilisateur répond par la lettre **O** ou **o**, l'exécution revient au point **d** ;
- m)** autrement, le programme affiche une seule fois le message "*Au revoir !*" et s'arrête.

A l'exécution, la sortie du programme doit respecter la mise en page donnée dans l'exemple ci-dessous.

Le but du programme est ...
 Introduisez le coefficient du x au carré :
1
 Introduisez le coefficient du x :
2
 Introduisez le terme libre :
1
 Deux solutions réelles confondues : $x = -1.0$.
 Voulez-vous continuer ?(O/N)
O
 Introduisez le coefficient du x au carré :
0
 Valeur non valide !
 Introduisez le coefficient du x au carré :
0
 Valeur non valide !
 Introduisez le coefficient du x au carré :
1
 Introduisez le coefficient du x :
0
 Introduisez le terme libre :
9
 Pas de solution réelle !
 Voulez-vous continuer ?(O/N)
O
 Introduisez le coefficient du x au carré :
1
 Introduisez le coefficient du x :
1
 Introduisez le terme libre :
-2
 Deux solutions réelles distinctes : $x_1 = 1.0$ et $x_2 = -2.0$.
 Voulez-vous continuer ?(O/N)
H
 Au revoir !

This image shows a full page of primary-ruled paper. It features approximately 28 horizontal dotted lines spaced evenly down the page, providing a guide for handwriting practice. The paper is otherwise blank, with no margins, text, or other markings.

[illegible]