

**Contrôle d'informatique no 2**

Durée : 1 heure 45'

Nom : .....

Groupe : 

Prénom : .....

Barème sur 100 points

No	1	2	3
Total points	9 points	43 points (21 + 22)	48 points

**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.**

Indiquer si les affirmations suivantes sont vraies ou fausses (en écrivant, selon le cas, VRAI ou FAUX à l'endroit prévu à cet effet) :

- a) L'adresse d'un objet créé par une méthode (et stockée dans une variable locale à la méthode) ne peut pas être transmise au programme appelant cette méthode (et l'objet devient toujours candidat au Garbage Collector à la fin de l'exécution de la méthode).

.....

- b) On peut affecter à une variable déclarée de type tableau (monodimensionnel) de nombres réels (par exemple *double*) l'adresse d'un tableau (monodimensionnel) de nombres entiers (par exemple *int*).

.....

- c) Dans le corps de la définition d'une méthode (d'instance ou statique), on peut faire appel à la méthode que l'on est en train de définir.

.....

## Sujet no 2.

2.1 Préciser les messages qui seront affichés à l'écran suite à l'exécution du projet contenant les deux classes suivantes :

```
package cms_ctr2;

public class UneDate
{
    private int an;
    private int mois;

    UneDate( )
    {
        this(2007, 2);
        System.out.println("Constructeur sans argument !");
    }

    UneDate(int an, int mois)
    {
        System.out.println("Constructeur avec 2 arguments !");

        if(an >= 1970)                this.an = an;
        else
        {
            System.out.println("An non valide !");
            this.an = 1970;
        }
        if(mois >= 1 && mois <= 12)    this.mois = mois;
        else
        {
            System.out.println("Mois non valide !");
            this.mois = 1;
        }
    }

    public int getAn( )
    {
        return an;
    }

    public int getMois( )
    {
        return mois;
    }

    public static int uneMethode(UneDate premiere,
                                UneDate deuxieme)
    {
        if(premiere.an > deuxieme.an)    return 1;
        if (premiere.an < deuxieme.an)   return -1;
        if(premiere.mois > deuxieme.mois) return 1;
        if(premiere.mois < deuxieme.mois) return -1;
        return 0;
    }

    public void afficher( )
    {
        System.out.println("La date : " + mois + "/" + an + ".");
    }
} //fin de la classe UneDate
```

```

package cms_ctr2;

public class CP_Ctr2_2a
{
    public static void main(String[ ] args)
    {
        UneDate d1, d2, d3, d4, d5;
        d1 = new UneDate(1992, 12);
        d2 = new UneDate( );

        System.out.println("-----");

        d3 = new UneDate(1291, 8);
        d4 = new UneDate(2000, 24);
        d5 = new UneDate(1992, 12);

        System.out.println("-----");

        d2.afficher( );
        d3.afficher( );
        d4.afficher( );

        System.out.println("-----");

        System.out.println(d1 == d2);
        System.out.println(UnDate.uneMethode(d1,d2));
        System.out.println(UnDate.uneMethode(d2,d1));

        System.out.println("-----");

        System.out.println(d1 == d5);
        System.out.println(UnDate.uneMethode(d1,d5));
        System.out.println(UnDate.uneMethode(d5,d1));

        System.out.println("-----");

        d2 = d5;
        System.out.println(d2 == d5);
        System.out.println(UnDate.uneMethode(d2,d5));
        System.out.println(UnDate.uneMethode(d5,d2));

    }    //fin de la méthode main
}      //fin de la classe CP_Ctr2_2a

```

[illegible]

**2.2** Préciser les messages qui seront affichés à l'écran suite à l'exécution du projet contenant les deux classes suivantes :

```
package cms_ctr2;

public class UneClasse
{
    boolean statel;
    boolean state2;
    private static int nb = 10;

    UneClasse( )
    {
        nb++;
    }

    UneClasse(boolean st1, boolean st2)
    {
        statel = st1;
        state2 = st2;
        nb++;
    }

    public static int getNb( )
    {
        return nb;
    }

    public UneClasse methode_A( )
    {
        UneClasse aux = new UneClasse( );
        aux.statel = statel;
        aux.state2 = state2;
        System.out.println("Appel un !");
        return aux;
    }

    public UneClasse methode_B( )
    {
        UneClasse aux = new UneClasse();
        //! est l'opérateur logique "négation"
        aux.statel = !statel;
        aux.state2 = !state2;
        System.out.println("Appel deux !");
        return aux;
    }

    public static UneClasse methode_C(boolean flag, UneClasse arg)
    {
        UneClasse aux;
        if(flag)
            aux = arg.methode_A( );
        else
            aux = arg.methode_B( );
        System.out.println("Appel trois !");
        return aux;
    }

    public static void afficher(UneClasse arg)
    {
        System.out.println("Premier = " + arg.statel +
                           " ; deuxième = " + arg.state2+".");
    }
}

//fin de la classe UneClasse
```

```

package cms_ctr2;

public class CP_Ctr2_2b
{
    public static void main(String args[ ])
    {
        UneClasse tab[ ] = new UneClasse[5];

        for(int i = 0; i < tab.length; i++)
        {
            //% est l'opérateur arithmétique "modulo"
            if(i % 3 == 0)
            {
                tab[i] = new UneClasse();
                System.out.println("-----");
            }else if(i % 2 == 0)
            {
                tab[i] = new UneClasse(true, false);
            }else
            {
                tab[i] = new UneClasse(false, true);
            }
            UneClasse.afficher(tab[i]);
        }

        System.out.println("-----");

        UneClasse ref1 = new UneClasse(true, false);
        UneClasse ref2 = new UneClasse( );
        UneClasse ref3 = new UneClasse(false, true);

        ref1 = ref2.methode_B( );
        ref3 = UneClasse.methode_C(false, ref2);
        ref2 = ref3.methode_A( );

        System.out.println("-----");

        UneClasse.afficher(ref1);
        UneClasse.afficher(ref2);
        UneClasse.afficher(ref3);

        System.out.println("-----");

        System.out.println("Des objets créés : " +
                           UneClasse.getNb( ) + " !");

    }    //fin de la méthode main
}      //fin de la classe CP_Ctr2_2b

```

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

### Sujet no 3.

Le but de cet exercice est de réaliser un jeu interactif du type "**Pierre-Feuille-Ciseaux**". Le programme et l'utilisateur choisissent un des trois objets suivants : "**Pierre**" indiqué par la lettre majuscule **P**, "**Feuille**" indiqué par la lettre majuscule **F** et "**Ciseaux**" indiqué par la lettre majuscule **C**. De façon générale, la "**Pierre**" bat les "**Ciseaux**" (en les émoussant), les "**Ciseaux**" battent la "**Feuille**" (en la coupant), la "**Feuille**" bat la "**Pierre**" (en l'enveloppant). Ainsi chaque objet bat un autre objet, fait match nul contre son homologue et est battu par le troisième.

On suppose que l'utilisateur fait confiance au programme qui choisit d'abord son objet sans le communiquer. Ensuite, l'utilisateur introduit au clavier la lettre correspondant à son objet choisi. Finalement, le programme compare les deux choix et annonce le gagnant jeu par jeu. De plus, une partie se termine quand le premier joueur réalise trois jeux gagnants et l'utilisateur peut jouer plusieurs parties à la suite.

**La stratégie de jeu** du programme est la suivante :

- au début, le programme choisit la "**Pierre**", c'est-à-dire la lettre majuscule **P** ;
- après chaque jeu gagné, le programme maintient le même choix qu'auparavant ;
- après chaque jeu perdu, le programme change d'objet en choisissant celui avec lequel l'utilisateur vient de gagner.

Il s'agit d'écrire le contenu d'un fichier source **ClassePrincipale.java** qui englobe deux classes appelées **Auxiliaire** et **ClassePrincipale**, la dernière classe étant déclarée publique.

#### 3.1

La classe **Auxiliaire** doit contenir une méthode publique et statique appelée **calculer** qui sera utilisée dans la classe **ClassePrincipale**. Cette méthode a deux arguments de type caractère et renvoie une valeur de retour de type entier, à savoir :

- **1** si le premier argument est la lettre majuscule **P** et le deuxième argument la lettre majuscule **C** ou si le premier argument est **C** et le deuxième **F** ou si le premier argument est **F** et le deuxième **P** ;
- **-1** si le premier argument est la lettre majuscule **C** et le deuxième argument la lettre majuscule **P** ou si le premier argument est **F** et le deuxième **C** ou si le premier argument est **P** et le deuxième **F** ;
- **0** dans tous les autres cas.



### 3.2

La classe publique *ClassePrincipale* doit contenir la méthode *main* et utiliser la méthode statique définie dans la classe *Auxiliaire*. Le but de la classe *ClassePrincipale* est de réaliser le jeu interactif et elle doit respecter les consignes suivantes :

- a) on affiche une seule fois, tout au début, le message "**Le but du jeu ... !**" (qui normalement devrait donner à l'utilisateur toutes les informations concernant le but et le déroulement du jeu comme on l'a fait au début du **Sujet no 3.**) ;
- b) on affiche au début de chaque jeu (y compris le tout premier) le score de la partie courante ;
- c) on demande à l'utilisateur de faire son choix précisé par une des lettres majuscules suivantes : **P** (pour la "*Pierre*"), **F** (pour la "*Feuille*") et **C** (pour les "*Ciseaux*") ;
- d) si l'utilisateur introduit au clavier un caractère qui n'est pas une des lettres majuscules indiquées ci-dessus, le programme revient (autant de fois que nécessaire) au point c), après avoir affiché le message "**Choix non valide !**" ;
- e) autrement, c'est-à-dire si l'utilisateur a introduit au clavier une des lettres majuscule indiquées au point c), le programme calcule le gagnant du jeu (en fonction de son propre choix, effectué conformément à la stratégie de jeu déjà expliquée, et du choix de l'utilisateur) et on affiche un message qui annonce, selon le cas, à qui revient le point gagnant ou s'il s'agit d'une égalité ;
- f) si aucun de deux joueurs n'a encore obtenu 3 points gagnants, le programme revient au point b) ;
- g) par contre, dès que le premier joueur réalise 3 points, on annonce la fin de la partie, le score final et le gagnant ;
- h) ensuite, on demande à l'utilisateur s'il veut jouer une nouvelle partie (la lettre majuscule **O** ou la lettre minuscule **o** signifiant une réponse affirmative de sa part) ;
- i) si l'utilisateur introduit au clavier une des lettres **O** ou **o**, le programme revient au point b) ;
- j) par contre, si l'utilisateur introduit au clavier une lettre différente de **O** et **o**, le programme se termine après avoir affiché le message "**Au revoir !**".

**Remarque :** à l'exécution, la sortie du programme doit respecter la mise en page donnée comme exemple ci-après.

### Indications :

- au début de chaque partie, le score initial est de 0 points pour l'utilisateur et 0 points pour le programme (JVM) ;
- tout au début de la première partie, le programme choisit la "*Pierre*", c'est-à-dire la lettre majuscule '*P*' ;
- le calcul du gagnant d'un jeu est effectué à l'aide de la méthode ad-hoc de la classe *Auxiliaire* ;
- après chaque jeu, le choix suivant du programme est fixé en fonction de la stratégie de jeu discutée (i.e. le même choix qu'auparavant si le programme a gagné ou le dernier choix de l'utilisateur si celui-ci a gagné).

<pre>Le but du jeu ... ! Le score : Vous = 0 et JVM = 0. Faites votre choix (P ou F ou C) : <b>F</b> Un point pour vous ! Le score : Vous = 1 et JVM = 0. Faites votre choix (P ou F ou C) : <b>F</b> Egalité ! Le score : Vous = 1 et JVM = 0. Faites votre choix (P ou F ou C) : <b>Z</b> Choix non valide ! Faites votre choix (P ou F ou C) : <b>P</b> Un point pour JVM ! Le score : Vous = 1 et JVM = 1. Faites votre choix (P ou F ou C) : <b>P</b> Un point pour JVM ! Le score : Vous = 1 et JVM = 2. Faites votre choix (P ou F ou C) : <b>F</b> Egalité ! Le score : Vous = 1 et JVM = 2. Faites votre choix (P ou F ou C) : <b>C</b> Un point pour vous ! Le score : Vous = 2 et JVM = 2. Faites votre choix (P ou F ou C) : <b>F</b> Un point pour JVM ! Fin de partie ! Le score final : Vous = 2 et JVM = 3. JVM a gagné ! Voulez vous jouer à nouveau ? (O/N) <b>O</b></pre>	<pre>Le score : Vous = 0 et JVM = 0. Faites votre choix (P ou F ou C) : <b>C</b> Egalité ! Le score : Vous = 0 et JVM = 0. Faites votre choix (P ou F ou C) : <b>F</b> Un point pour JVM ! Le score : Vous = 0 et JVM = 1. Faites votre choix (P ou F ou C) : <b>F</b> Un point pour JVM ! Le score : Vous = 0 et JVM = 2. Faites votre choix (P ou F ou C) : <b>P</b> Un point pour vous ! Le score : Vous = 1 et JVM = 2. Faites votre choix (P ou F ou C) : <b>C</b> Un point pour JVM ! Fin de partie ! Le score final : Vous = 1 et JVM = 3. JVM a gagné ! Voulez vous jouer à nouveau ? (O/N) <b>N</b> Au revoir !</pre>
--	--

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

[illegible]

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.

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.