

MODULE POO, EXAMEN ORDINAIRE DU AVRIL 2018
PARTIE 01 : OUTIL POUR LA POO : LE LANGAGE JAVA

// EXERCICE 01 : (4 Points/40)

Dans cet exercice **le travail se fait en mode terminal** (appelé aussi mode console) dans le contexte du système Microsoft Windows 7. Le **Code Source N°1** se trouve dans un seul fichier. Le dossier de travail dans lequel se trouve ce fichier est **d:\travail**. La compilation de ce code source se fait **sans aucune erreur**.

Question 1 : Proposer un nom pour le fichier qui contient tout ce code source (2 pts/40)

Question 2 : Ecrire la commande permettant d'exécuter ce programme (2 pts/40).

```
package examen.poo.session1;
class Main {
    static int a = 5;
    static final int MAX = 54;
    public static String quiSuiJe(){
        return ("Je suis la classe Principale");
    }
}
////////// La Classe Personne//////////
class Personne {
    void identite(String[] args) {
        System.out.println("Je suis une Personne");
    }
}
////////// La Classe Etudiant//////////
class Etudiant {
    void identite(String[] args) {
        System.out.println("Je suis un Etudiant");
    }
    public static void main(String[] args) {
        System.out.println("Salam, je suis la classe Etudiant");
    }
}
```

Code Source N° 1

Réponse 1 : lib.java (le nom doit commencer par une minuscule)

/// 2 points/40

Réponse 2 : java examen.poo.session1.Etudiant

/// 2 points/40

// EXERCICE 02 : (4 Points/40)

Le **Code Source N°2** montre une partie de la classe **CodeurDecodeur**. Supposons que toutes les méthodes de cette classe (non reportées dans le **Code Source N°2**) ne contiennent pas d'erreur. Seule, la compilation de la méthode **codage()** signale une erreur. Sans modifier la méthode **codage** corriger le code source.

ATTENTION: LORS DE L'ECRITURE DE LA CORRECTION, INDIQUER LE NUMERO DE LIGNE OU DOIT ETRE PLACEE LA CORRECTION. EXEMPLE: 12: x=8;

```
1 :package examen.poo.session1;
2 :public class CodeurDecodeur {
3 :     double codeDeBase;
3 :     public static CodeurDecodeur codage(){
4 :         CodeurDecodeur cd = new CodeurDecodeur();
5 :         cd.setCodeDeBase(167.0);
6 :         return cd;
7 :}
// Ci dessous les autres méthodes de cette classe
// .....
}
```

Code Source N° 2

// 4 Pts/40

// EXERCICE 03 : (6 Points / 40)

La compilation du programme du **Code Source N° 3** signale une erreur. Après correction de cette erreur, l'exécution de ce programme affiche sur écran le message ci-dessous

La valeur maximale du tableau tabInt est: 40

Trouver et corriger l'erreur **EN RESPECTANT OBLIGATOIREMENT LES 2 REGLES SUIVANTES:**

- Les prototypes et type de retour des 2 méthodes ne doivent pas être modifiés
- Aucune écriture n'est permise l'extérieur des 2 méthodes

```
package examen.poo.session1;
public class Question03 {
    int max(int[] ti){
        if (ti == null) return -1;
        if (ti.length == 0) return -1;
        int iMax = 0;
        for(int i = 1; i < ti.length; i=i+1 )
            if (ti[i] > ti[iMax]) iMax = i;
        return iMax;
    }
    public static void main(String[] args) {
        int[] tabInt = { 10,20,30,40,25,35};
        System.out.println("La valeur maximale du tableau tabInt est: " +
            tabInt[new Question03().max(tabInt)]);
    }
}
```

Code Source N° 3

// 6 points /40

// EXERCICE 04 : (6 Points / 40)

Nous désirons que le programme décrit dans le **Code Source N° 4** imprime sur écran les 2 lignes suivantes :

```
500
49
```

Cependant l'exécution de ce programme tel qu'écrit dans le **Code Source N° 4** donne une autre valeur.

Question 01 : Quelle est cette valeur ? **REPONSE : 10, 10** **2 pts / 40**

Question 02 : Ajuster le code du programme pour qu'il affiche effectivement les 2 lignes désirées (500 puis 49) (3pts)

ATTENTION!

- L'ajustement ne doit pas modifier les prototypes de méthodes.
- Le nombre d'instructions et la structure du code doivent rester les mêmes.
- Pour la réponse : pour toute correction, indiquer le numéro de l'instruction à corriger et sa correction

```
1 :package examen.poo.session1;
2 : public class Question02 {
3 :     static int code = 10;
4 :     static void positionnerLeCode(int code){
5 :         if (code > 10)
6 :             Question02.code = code*25;
7 :         else
8 :             Question02.code = code -5;
9 :     }
10 :    public static void main(String[] a){
11 :        positionnerLeCode(20);
12 :        System.out.println(code);
13 :        positionnerLeCode(5);
14 :        System.out.println(code);
15 :    }
16 :}
```

Code Source N° 4

//2pts / 40

//2pts / 40

// EXERCICE 05 : (4 Points/40)

Compléter le **Code source N° 5** afin que son exécution affiche sur console la ligne suivante

```
10,122
```

```

package examen.poo.session1;
public class Question04 {
    static int[] a = new int[10]; // 2pts/40
    public static void manipTable(){
        a[5] = 122;
        System.out.println(a.length + ","+a[5]);
    }
    public static void main(String[] args) {
        manipTable(); // 2pts/40
    }
}
    
```

Code Source N° 5

PARTIE 02 : TRANSFORMATION NOTATION GRAPHIQUE VERS LE LANGAGE JAVA

// EXERCICE 06 : (8 Points/40)

Le Code Source N° 6 représente le diagramme de classe N°1. Compléter ce code

ATTENTION :

--RESPECTER OBLIGATOIREMENT LA REGLE SUIVANTE : tout écriture faite (variable, constantes, méthodes), si elle peut être réutilisée, il faut obligatoirement la réutiliser, autrement la réponse est considérée comme fausse même si en programmation elle est correcte.

--DANS CET EXERCICE TOUTES LES CLASSES SONT DANS LE FICHIER Question06.java

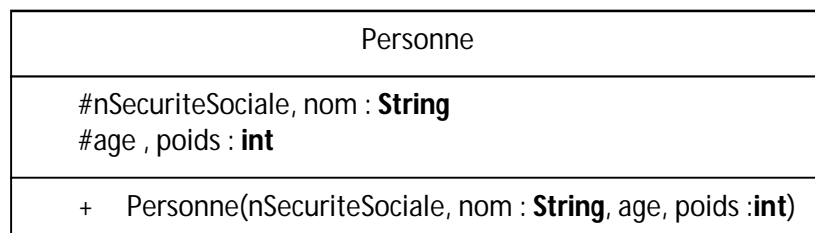
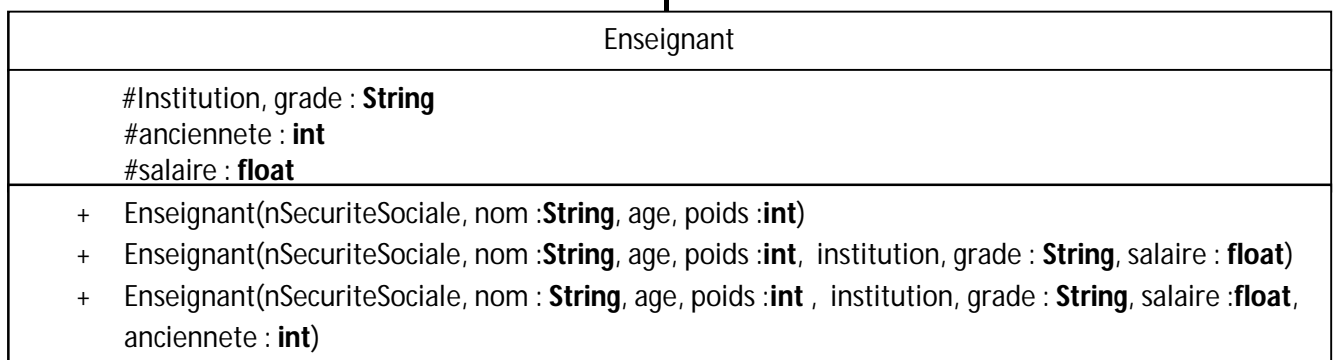


Diagramme de classe N° 01



```

package examen.poo.sessi on1;
public class Question06 {
    // Cette classe est utilisée juste pour indiquer le numéro de la question
}
class Personne {    /// 1 points/40 // faux -1 points // vide : 0 point
    protected String nSecuri teSoci ale;
    protected String nom;
    protected int age;
    protected int poi ds;
    // 1er constructeur: Constructeur par défaut
    public Personne(){
    // 2ème constructeur:
    public Personne(String nSecuri teSoci ale, String nom, int age, int poi ds){
        this.nSecuri teSoci ale = nSecuri teSoci ale;
        this.nom = nom;
        this.age = age;
        this.poi ds = poi ds;
    }
}
class Enseignant extends Personne {    // 1 points, faux - 1 points , vide 0
    protected String insti tution;
    protected String grade;
    protected int anci ennete;
    protected float sal ai re;

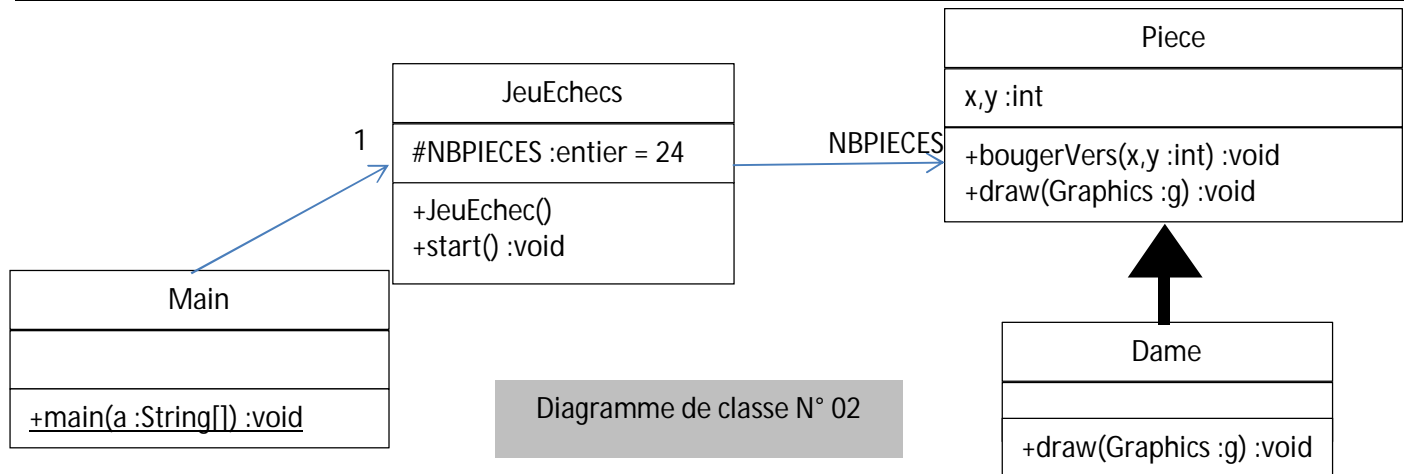
    public Enseignant(){
    public Enseignant(String nSecuri teSoci ale, String nom, int age, int poi ds){
        super(nSecuri teSoci ale, nom, age, poi ds); //2 points, faux 0 points
    }
    public Enseignant(String nSecuri teSoci ale, String nom, int age, int poi ds,
        String insti tution, String grade, float sal ai re){
        // 1- thi s(nSecuri teSoci ale, nom, age, poi ds);
        // 2-super (nSecuri teSoci ale, nom, age, poi ds);
        super (nSecuri teSoci ale, nom, age, poi ds);    // 1,5 Pts /40,
        this.insti tution = insti tution;    // Les 3 instructions
        this.grade = grade;    // sur
        this.sal ai re = sal ai re;    // 0.5
    }
    public Enseignant(String nSecuri teSoci ale, String nom, int age, int poi ds,
        String insti tution, String grade, float sal ai re, int anci ennete){
        this(nSecuri teSoci ale, nom, age, poi ds, insti tuti on, grade, sal ai re); //1,5 pts
        this.anci ennete = anci ennete; // 0.5 points,
    }
}
}

```

// EXERCICE 07 : (8 Points/40)

Le **diagramme de classe** N° 2 représente partiellement un jeu de dame. Le **code source** N° 7 est une transformation partielle du **diagramme de classes** N° 2 en langage Java. Compléter le **code source** N°7 pour qu'il puisse représenter fidèlement le **diagramme de classe** N°2.

ATTENTION : DANS CET EXERCICE CHAQUE CLASSE A SON PROPRE FICHER.



Code Source N° 7

```

public class Main01{
    public static void main(String[] a){
        new JeuEchecs().start();
    }
}

public class JeuEchecs{
    public final int NBPIECES = 24;           /// 0.5 Pts / 40
    Piece[] tabPiece = new Piece[NBPIECES];  /// 1.5 pts /40
    public void start(){
        /// Constructeur Par défaut
    }
    public JeuEchecs(){                       ///// 0.5
        for (int i = 0; i < NBPIECES; i++) tabPiece[i] = new Piece(this); //2 pts
    }
}

class Piece{
    protected JeuEchecs jeuEchecs;           // 1 pts
    int x , y;
    /// Les Constructeurs
    /// 1. 1er constructeur: Le Constructeur Par défaut
    public Piece(){
        //..... Code cachée. Non concerné par le test
    }
    /// 2. 2ème constructeur:
    public Piece(JeuEchecs jeuEchecs){        //2 Pts/40
        this.jeuEchecs = jeuEchecs;           // 0.5/40
    }
    /// Les méthodes
    public void bougerVers(int x, int y){
        this.x = x; this.y = y;
    }
    public void draw(Graphics g){}
}

public class Dame extends Piece{
    public void draw(Graphics g){}
}
    
```