

**DEVOIR SURVEILLÉ**Matière: **POO**Calculatrice Scientifique: **Non autorisée**Documents: **Non autorisés**

Enseignants Chelbi Samia, Chouchan Haykel, Mhafchi Houda, Hedhli Jihen

Durée: **1h00'**

Classes : DSI21, DSI23, DSI22, RSI21.

Date: 11/11/2015

*Il vous est demandé d'apporter un soin particulier à la présentation de votre copie.***EXERCICE 1 :****5 PTS**

- Pour la classe D définie comme suit:

```
classD
{ public int x;
  public D() {x=3;}
  public D( int a) { this(); x=x+a; }
  public D( int a, int b){ this(b); x= x-a; } }
```

Qu'affichera le code suivant?

```
D a=new D(5,6);
System.out.println(a.x);
```

- (a) 1
- (b) 2
- (c) 3
- (d) 4

Déterminer s'il y a des erreurs, corrigez les et affichez l'exécution du programme:

- ```
class test1 {
    public static void main () {
        System.out.println("bonjour") ;
    }
    public void affiche () {
        System.out.println("monsieur") ;
    } }

class test2 {
    public static void main (String[] args) {
        test1.main();
        test1.affiche();
    } }
```

**EXERCICE 2 :****15PTS**

On désire programmer une mini-application permettant de gérer un ensemble d'étudiants. Chaque étudiant est caractérisé par les informations suivantes:

- Nom( attribut privé de type chaîne de caractères)
- 3 notes (que vous pouvez mettre sous forme d'un tableau `Note[]` de réel)

La classe *Etudiant* dispose d'un certain nombre de méthodes :

- Un constructeur qui prend comme paramètre le nom et les 3 notes pour créer un objet étudiant.
- Un getter (`getNom()`) pour récupérer la valeur du nom.
  - Un setter (`setNom()`) pour modifier la valeur du nom.
- Une méthode *calcul\_Moyenne()* pour calculer la moyenne des 3 notes et la retourner.
  - Une méthode *String resultatAnnée()* qui retourne une chaîne vide.
- Une méthode *affiche()* pour afficher les données suivantes : le nom, la liste des notes, et la moyenne des notes. Sous la forme suivante :

*nom étudiant : Mohamed*

*notes universitaires : 11.0 13.0 18.0 7.0*

*moyenne universitaire : 12.25*

### 1) Ecrire la classe *Etudiant*

Il existe en fait deux types d'étudiants, ceux en **formation initiale** et **les apprentis** :

En plus de leurs notes universitaires (les 3 notes de la question 1),

- Les étudiants en formation initiale sont caractérisés par un nombre d'absence. Ils sont déclarés admis si leur moyenne universitaire est  $\geq 10$  et si leur nombre d'absence est  $< 10$ .
- Les apprentis sont caractérisés par le nom de l'entreprise dans laquelle ils travaillent, et ont une note supplémentaire «note pratique ». Leur moyenne de l'année est la moyenne de la moyenne universitaire et de cette note pratique.

Un apprenti est déclaré admis si sa moyenne de l'année est  $\geq 10$ .

### 2) Créer deux classes *EtudiantInitial* et *Apprenti* héritant de la classe *Etudiant*.

Pour chacune de ces deux classes on dispose de:

- Un constructeur avec paramètres
- Une méthode *String resultatAnnée()* qui retourne une chaîne « admis » ou « non admis » en fonction des résultats de l'étudiant.
- Une méthode *void affiche()* qui fera appel à la méthode *affiche()* de la classe mère *Etudiant* en la complétant de façon à obtenir les éditions suivantes :

#### Exemple 1

*Etudiant en formation initiale*

*nom étudiant : Mohamed*

*notes universitaires : 11.0 13.0 18.0 7.0*

*moyenne universitaire : 12.25*

*nombre d'absence : 8*  
*résultat final: admis*

*Etudiant en formation initiale*  
*nom étudiant : ALi*  
*notes universitaires : 10.0 9.0 8.0 14.0*  
*moyenne universitaire : 10.25*  
*nombre d'absence : 12*  
*résultat final: non admis*

*Apprenti de l'entreprise : EDF*  
*nom étudiant : Moez*  
*notes universitaires : 9.0 9.0 10.0 9.0*  
*moyenne universitaire : 9.25*  
*note pratique : 11.0*  
*moyenne Année : 10.125 à supprimer , la methode resultat\_annee calcule le moyen de l'annee et ne le retourne pas*  
*résultat final: admis*

- 3) Une classe **Institut** regroupe un ensemble d'étudiant et elle est dotée des attributs suivants :
- TabE :tableau d'étudiants
  - nbE : nombre Etudiant( entier)

Cette classe dispose des méthodes suivantes : constructeur, **void ajout(Etudiant E)** permettant d'ajouter un étudiant à la fin du tableau et une méthode affiche() qui affiche l'ensemble des étudiants du tableau.

Ecrire la classe **Institut**

- 4) Ecrire une classe Test qui :
- crée 3 étudiants décrits dans l'exemple1 ci-dessus,
  - crée un Institut et lui ajoute les 3 étudiants déjà créés,
  - Affiche le contenu de l'institut.

je propose une seule classe dans laquelle on demande de créer un tableau de 3 étudiants, créer 3étudiants diversifiés (initiale et apprentis), ajouter les au tableau puis les afficher