Examen L3 -P00 18/12/2013

Lisez bien l'énonce jusqu'à la fin. Vous pouvez répondre aux questions dans n'importe quel ordre en précisant bien leur numéro. Les barèmes sont indicatifs

Question 1. (5.5pts) Nous définissons une classe Notes pour représenter les notes des étudiants d'un groupe de TD. Cette classe a un attribut (une variable) privé *notesMat* qui est une matrice dont les éléments sont des réels (double[ ] [ ]).

```
class  Notes{
private double[ ] [ ] notesMat;
public Notes(int nbetud, int nbexam) { notesMat= new double[nbetud][nbexam];}
```

1. Ecrire une méthode qui rend pour résultat un vecteur qui représente la note moyenne de chaque étudiant *(double[] moyen_etud( ))*.
2. Ecrire une méthode qui rend pour résultat un vecteur qui représente la note moyenne pour chaque examen *(double[] moyen_exam( )*).
3. Ecrire une méthode pour tester si les notes sont supérieures ou égales à 0, sinon elle lance une exception.
4. Soit n un objet de classe *Notes*. Ecrire une méthode *main*, dans laquelle en utilisant des méthodes définies - tester si toutes les notes sont valides (>= 0), 2- afficher le nombre d'étudiants qui ont une moyenne < 10.

(Si $n.notesMat = \begin{bmatrix} 10 & 6 \\ 5 & 7 \\ 9 & 5 \end{bmatrix}$, le vecteur pour les notes moyennes des 3 étudiants serait [8 6 7], et celui pour les notes moyennes pour les 2 examens serait [8 6].)

Question 2. (3pts) On considère les définitions de classes suivantes:

```
class A{
void f( A o) {System.out.print("1 ");}
}
class B extends A{
void f(A o) {System.out.print("2 ");
// void f(B o) {System.out.print("3 ");}
}
```

a-Indiquer ce qu'affiche le fragment du programme suivant:

```
A a= new A();
A ab= new B();
B b=new B();
a.f(a);    a.f(ab);      a.f(b);     System.out.println();
ab.f(a);  ab.f(ab);     ab.f(b);     System.out.println();
b.f(a);    b.f(ab);      b.f(b);     System.out.println();
```

b-Si on supprime le commentaire dans la classe B (on ajoute une nouvelle méthode dans B), qu'affiche maintenant le fragment du programme au-dessous?

Question 3. (8.5pts) Nous considérons les classes suivantes.

```
import java.util.*;
class Date {
private int mois,annee;
public Date ( int m, int a)   {mois=m; annee=a;               }
abstract class Employe{
protected String nom;
protected  Date periode;
```

1. La classe abstraite *Employe* a 2 attributs: le nom, la date de la période considérée.Compléter le deuxième constructeur.

```
public Employe(String n, Date d)
{nom=n;  periode=d; }
public Employe(String n, int m, int a){.....................}
abstract public  double calculSalaire();                                    }
```

2. Les interfaces *ServiceAdmin* et *ServiceTechnique* sont définies comme ci-dessous. Les classes concrètes *EmployeAdmin*, *EmployeTechnique* sont dérivées de la classe *Employe* et elles ont un attribut privé *nbh* pour indiquer le nombre d'heures effectuées. La classe *EmployeAdmin* implémente Interface *ServiceAdmin*, la classe *EmployeTechnique* implémente Interface *ServiceTechnique*. Donner le code pour la classe *EmployeAdmin*.

```
interface ServiceAdmin {
public static final double tauxPrime=0.25;
public static final double prixHoraire=15;}
interface ServiceTechnique {
public static final double tauxPrime=0.15;
public static final double prixHoraire=11;}
```

3. La classe *Entreprise* est définie comme ci-dessous avec un attribut *String* pour le nommer et une liste d'employés *employes*. On constitue la liste en fonction des heures effectuées dans les services, un employé peut apparaître deux fois dans la liste, si il travaille dans les deux services durant la période considérée.

```
class Entreprise{
private String nom;
private List<Employe> employes ;
```

3.1.Donner une méthode pour calculer le nombre total d'heures effectuées pour l'employé passé en paramètre.

```
public static int {Employe e){.................}
```

3.2. Donner des méthodes qui rend 1-un ensemble de noms des employés (pas de répétition) 2- un ensemble de noms des employés qui travaillent dans les deux services.

```
public Set<String> ensembleEmploye(){.............}
public Set<String>doubleService(){.......................}
```

3.3. Donne une méthode qui retourne un tableau associatif avec clé: le nom de l'employé et valeur: le salaire totale de l'employé (la somme des salaires dans les deux services).

```
public  Map<String,  Double >  salairesTotal(){.................}
```

Question 4. (3 pts)

```
public class Box<T> {
private T t;
public void add(T t) {this.t = t;}
public T get() {    return t;}
public static void main(String[] args) {.....................
}}
```

1- Dans la méthode main, créer un objet *intBox* où T est un entier; créer un objet *stBox* où T est un String.

2- L'objet *intBox* appelle la méthode *add* avec paramètre 10 et L'objet *stBox* appelle la méthode *add* avec paramètre "Hello".

3- Est-il possible de faire *intBox= stBox*

4-Quel est le résultat de la comparaison:

if (intBox.getClass()==stBox.getClass())

MAP< K,V >: int size(); boolean isEmpty(); put(K key, V value) ; V get (K key); boolean containsKey(K key); boolean containsValue(V value)

Collection< E > : int size(); boolean isEmpty(); boolean contains(Object element); boolean add(E element); boolean remove(Object element);