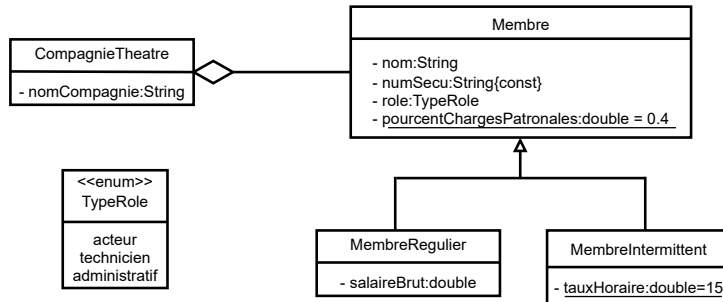


Contrôle continu optionnel HAI7171 – Ecrit 13 janvier 2023  
Programmation par objets (2h)  
Eléments de correction

Université de Montpellier – Faculté Des Sciences  
Master informatique (ICO), géomatique, bioinformatique, Physique numérique

Nous étudions les éléments d'un logiciel de gestion pour les compagnies de théâtre. Une compagnie de théâtre se compose de membres qui peuvent être réguliers ou intermittents et dont on calculera les salaires. Un schéma général en UML de ce que nous allons construire est donné ci-dessous. Il se complètera au fur et à mesure des questions.



**Question 1 (0,5).** Écrivez le code Java pour l'énumération **TypeRole** qui décrit les valeurs possibles des rôles des membres : acteur, technicien ou administratif.

```
public enum TypeRole {acteur,technicien,administratif}
```

**Question 2 (2).** Écrivez pour la classe Membre le code Java de :

- son entête
- ses attributs : chaque membre a un nom, un numéro de sécurité sociale (chaîne de caractères, dont la valeur ne peut changer après attribution, elle est donc définitive) et un rôle ; de plus un attribut permet de stocker le pourcentage (réel entre 0 et 1) qui sera appliqué pour le calcul du salaire chargé, la valeur de cet attribut est globale, c'est-à-dire partagée par tous les membres.
- un constructeur avec les paramètres servant à initialiser les attributs qui vous semblent appropriés.

Les accesseurs existent pour les attributs, seul l'accesseur **setNumSecu** ne peut exister. Expliquer pourquoi.

```
public class Membre {
    private String nom;
    private final String numSecu;
    private TypeRole role;
    private static double pourcentChargesPatronales = 0.4;

    public Membre(String nom, String numSecu, TypeRole role) {
        this.setNom(nom);
        this.numSecu = numSecu;
        this.setRole(role);
    }
}
```

**Question 3 (2).** Écrivez pour les classes représentant les membres réguliers et les membres permanents le code Java de :

- leur entête
- leurs attributs : un membre régulier a un salaire brut. Un attribut permet de stocker le taux horaire des membres intermittents, la valeur de cet attribut est globale, c'est-à-dire partagée par tous les membres intermittents.
- leurs constructeurs avec les paramètres servant à initialiser les attributs qui vous semblent appropriés.

```
public class Regulier extends Membre {
    private double salaireBrut;
    public Regulier(String nom, String numSecu, TypeRole role, double salaireBrut) {
        super(nom, numSecu, role);
        this.setSalaireBrut(salaireBrut);
    }
}
```

```
public class Intermittent extends Membre {
    private static double tauxHoraire = 15;
    public Intermittent(String nom, String numSecu, TypeRole role) {
        super(nom, numSecu, role);
    }
}
```

**Question 4 (2,5).** Écrivez le code Java pour disposer pour les membres (réguliers ou intermittents) d'une méthode calculant et retournant le salaire mensuel :

- pour les membres réguliers, il s'agit simplement de la valeur de l'attribut salaire brut.
- pour les membres intermittents, il s'agit du produit du taux horaire par 140 (correspondant à 140h de travail, qui est le maximum possible d'heures que l'intermittent pourrait avoir sur un mois complet).

Quel schéma de programmation avez-vous appliqué (voir cours sur l'héritage) ?

Devez-vous changer l'entête de la classe **Membre** suite à cette écriture ?

Dans Membre (qui devient abstract)  
abstract public double salaireMensuel();

Dans Regulier  
public double salaireMensuel() {return salaireBrut;}

Dans Intermittent  
public double salaireMensuel() {return 140\*Intermittent.tauxHoraire;}

Il s'agit du schéma par spécialisation avec méthode abstraite.

**Question 5 (1,5).** Écrivez le code Java pour disposer pour les membres d'une méthode calculant et retournant le salaire mensuel chargé. Il s'agit du salaire mensuel auquel on

applique le pourcentage de charges patronales (il faut utiliser l'attribut **pourcentChargesPatronales**. Ici avec sa valeur initialement donnée on augmenterait de 40% le salaire mensuel).

Quel schéma de programmation avez-vous appliqué (voir cours sur l'héritage) ?

Dans Membre :

```
public double salaireAvecCharges() {  
    return this.salaireMensuel()*(1+Membre.pourcentChargesPatronales);  
}
```

Il s'agit du schéma par généralisation.

**Question 6 (2,5).** Écrivez le code Java pour disposer pour les membres d'une méthode de saisie. Quel schéma de programmation avez-vous appliqué (voir cours sur l'héritage) ?

Dans Membre :

```
public void saisie(Scanner sc) {  
    System.out.println("Entrer le nom");  
    this.setNom(sc.next());  
    System.out.println("Entrer le role");  
    this.setRole(TypeRole.valueOf(sc.next()));  
}
```

Dans Regulier :

```
public void saisie(Scanner sc) {  
    super.saisie(sc);  
    System.out.println("Entrer le salaire brut");  
    this.setSalaireBrut(sc.nextDouble());  
}
```

Dans Intermittent, on n'écrit pas de saisie car la version héritée convient.

Il s'agit du schéma par spécialisation.

**Question 7 (2,5).** Écrivez une partie du code Java d'une classe représentant les compagnies de théâtre :

- son entête
- ses attributs. Une compagnie de théâtre a un nom et une liste de membres.
- un constructeur pour initialiser les attributs, choisissez son paramétrage.
- une méthode permettant d'ajouter un membre à la compagnie s'il n'y est pas déjà.

```
public class CompagnieTheatre {  
    private String nomCompagnie="";  
    private ArrayList<Membre> listeMembres =new ArrayList<>();  
    public CompagnieTheatre(String nomCompagnie) {  
        this.setNomCompagnie(nomCompagnie);  
    }  
    public void ajoute(Membre m) {  
        if (! this.listeMembres.contains(m))  
            this.listeMembres.add(m);  
    }  
}
```

**Question 8 (2).** Écrivez pour la classe représentant les compagnies de théâtre une méthode calculant et retournant le coût salarial mensuel chargé total (somme des coûts des salaires chargés des membres).

```
public double coutSalarialMensuelAvecCharges() {
    double somme = 0;
    for (Membre m : this.listeMembres)
        somme += m.salaireAvecCharges();
    return somme;
}
```

**Question 9 (3).** Écrivez pour la classe représentant les compagnies de théâtre une méthode construisant et retournant la sous-liste des membres vérifiant les deux propriétés suivantes :

- ils jouent un certain rôle passé comme premier paramètre
- leur salaire est supérieur à un certain seuil passé comme deuxième paramètre.

```
public ArrayList<Membre> sousListeRoleSalaireSup(TypeRole tr, double seuil){
    ArrayList<Membre> res = new ArrayList<>();
    for (Membre m : this.listeMembres)
        if (m.getRole().equals(tr) && m.salaireMensuel()>=seuil)
            res.add(m);
    return res;
}
```

**Question 10 (1,5).** Écrivez les instructions d'un programme main permettant de :

- créer un membre régulier, l'actrice « Ida », de numéro de sécurité sociale 299 et de salaire mensuel égal à 3000 euros,
- créer un membre intermittent, la technicienne « Anna » de numéro de sécurité sociale 298.
- créer une compagnie de théâtre,
- lui ajouter les deux membres ;
- afficher le coût salarial mensuel chargé total de la compagnie,
- afficher la sous-liste des acteurs ayant un salaire supérieur à 2500 euros.

```
public static void main(String[] args) {
    Regulier r1 = new Regulier("Ida", "299", TypeRole.acteur, 3000);
    Intermittent i2 = new Intermittent("Anna", "298", TypeRole.technicien);
    CompagnieTheatre ct = new CompagnieTheatre("La comédie");
    ct.ajoute(r1); ct.ajoute(i2);
    System.out.println(ct.coutSalarialMensuelAvecCharges());
    System.out.println(ct.sousListeRoleSalaireSup(TypeRole.acteur, 2500));
}
```