



Nom :
Prénom :
Numéro d'étudiant :

CORRECTION

Contrôle continu

Tous documents sur support papier autorisés. Durée : 1h15.
L'ensemble des réponses sera à donner sur les feuilles d'énoncé. Ne pas dégrapher les feuilles.

On s'intéresse à la mise en place d'un logiciel de gestion du personnel d'une petite entreprise.
On prévoit le modèle partiel donné à la figure 1.

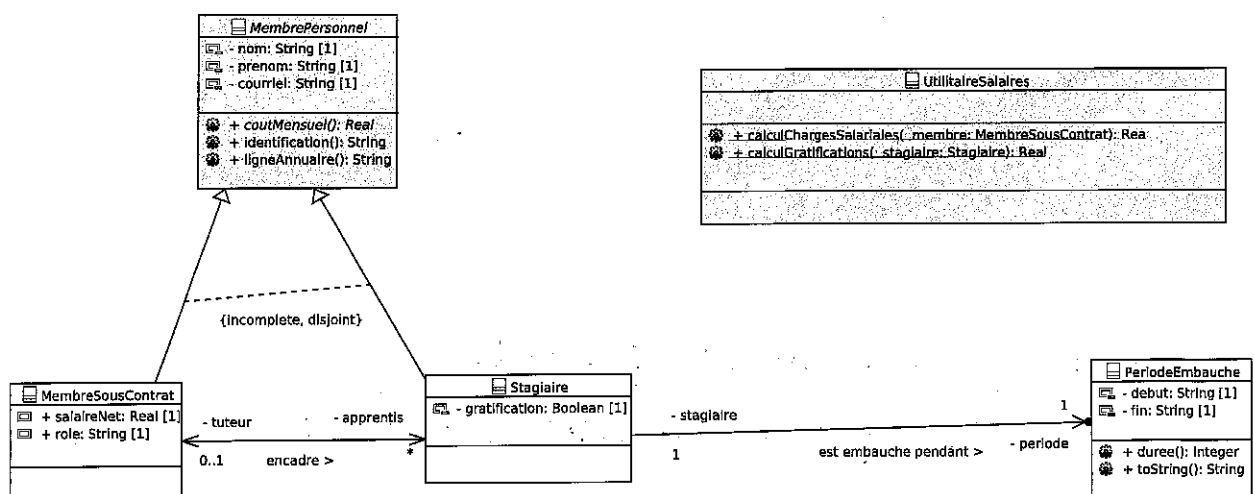


FIGURE 1 – Diagramme de classes partiel

Pour l'instant seuls les membres sous contrat et les stagiaires sont représentés. Chaque membre du personnel a un nom, un prénom, et une adresse électronique (courriel). Pour chaque membre du personnel, on peut calculer son coût mensuel pour l'entreprise, une chaîne correspondant à son identification, et une chaîne correspondant à la ligne lui correspondant dans l'annuaire. Chaque membre sous contrat a de plus un salaire net (ce que le salarié touche effectivement chaque mois), et un rôle (par exemple : responsable des ressources humaines).

Un stagiaire peut recevoir une rémunération ou pas, sous forme de gratifications. Au delà de 8 semaines de stage, c'est une obligation légale de gratifier le stagiaire. Le stagiaire est en poste dans l'entreprise pendant une certaine période, période pour laquelle on peut calculer sa durée (en semaines) et une chaîne caractéristique (sous la forme : "du jj/mm/aaa au jj/mm/aaaa") via la méthode `toString`. Pour simplifier les dates apparaissent sous forme de chaînes de caractères.

Un stagiaire est encadré par un tuteur (qui doit obligatoirement être renseigné au premier jour du stage, mais peut être laissé libre avant cette date). Un membre sous contrat peut encadrer plusieurs stagiaires (apprentis).

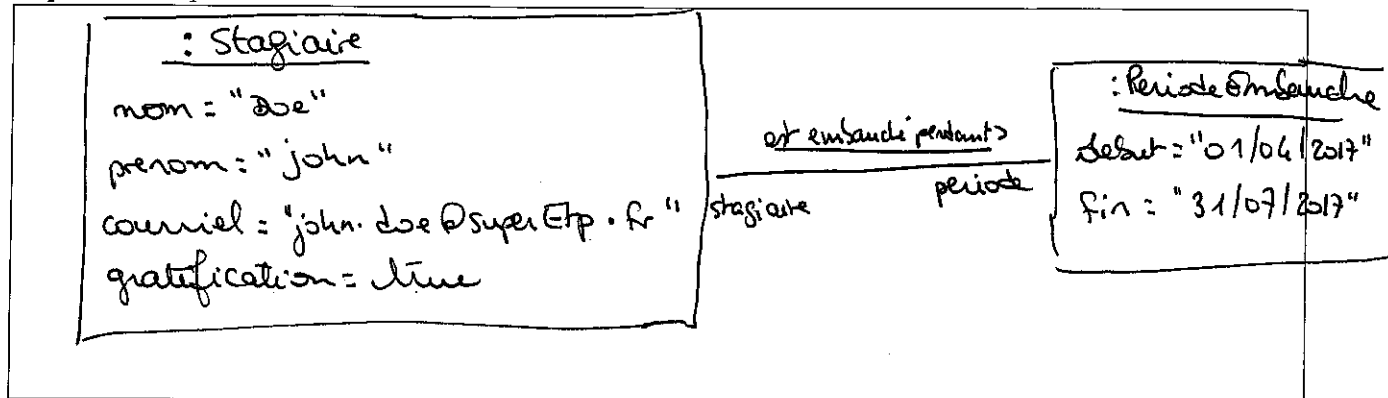
La classe **UtilitaireSalaires** possède deux méthodes de classe, aidant aux calculs des rémunérations : `calculChargesSalariales` retourne les charges salariales mensuelles liées à un membre du personnel sous contrat donné, et `calculGratifications` retourne les gratifications mensuelles pour un stagiaire donné.

Tous les prix sont donnés en euros.

On suppose disposer, sans les avoir écrits au préalable, de tous les accesseurs en lecture. Concernant le code Java, on suppose disposer de tous les en-têtes de classes, ainsi que de la déclaration de tous les attributs (provenant des attributs et des associations du modèle UML, en gardant les mêmes noms que dans le modèle).

Question 1. Proposez un diagramme d'objets représentant un stagiaire nommé John Doe, d'adresse électronique john.doe@superEtp.fr touchant des gratifications, sans tuteur affecté, effectuant son stage du 01/04/2017 au 31/07/2017.

Réponse à la question 1 :



Question 2. Définissez en Java une méthode `verificationGratifs` dans la classe `Stagiaire` permettant de vérifier que si la durée du stage excède 8 semaines, alors le stagiaire reçoit des gratifications (la méthode retournera vrai si la propriété est vérifiée, et faux sinon). On utilisera la méthode `duree` de la classe `PeriodeEmbauche` sans la définir.

Réponse à la question 2 :

```

public boolean verificationGratifs() {
    return periode.duree() >= 8 // gratification ;
}

// on suppose periode non null
    
```

Question 3. Proposez en Java l'ensemble du code nécessaire pour mettre en place les opérations suivantes pour tous les membres de l'entreprise :

- `coutMensuel`. Cette méthode retourne le coût mensuel pour l'entreprise du membre du personnel. Pour un stagiaire, ce coût est 0 si le stagiaire ne reçoit pas de gratifications, et le montant des gratifications sinon. Pour un membre sous contrat, le coût est le salaire net du membre, plus les charges salariales lui correspondant. On utilisera les méthodes `calculGratifications` et `calculChargesSalariales` de la classe `UtilitaireSalaires` sans les définir.
- `identification`. Cette méthode retourne pour les membres sous contrat comme pour les stagiaires une chaîne sous la forme : prénom nom.
- `ligneAnnuaire`. Cette méthode retourne pour les membres sous contrat une chaîne de la forme : prénom nom (adresse électronique) : rôle, et pour les stagiaires une chaîne de la forme : prénom nom (adresse électronique) : en stage du jj/mm/aaa au jj/mm/aaaa , tuteur : prénom, nom. On utilisera la méthode `toString` de la classe `PeriodeEmbauche` sans la définir. On supposera ici que le tuteur a bien été défini.

Réponse à la question 3, `coutMensuel` :

Dans `MembrePersonnel`

```
public abstract float coutMensuel();
```

Dans `MembreSousContrat`

```
public float coutMensuel() {
    return salaireNet + UtilitaireSalaires.calculChargesSalariales(this);
}
```

Dans `Stagiaire`

```
public float coutMensuel() {
    if (!gratifications) return 0.0f;
    else return UtilitaireSalaires.calculGratifications(this);
}
```

Réponse à la question 3, identification :

Dans Membre Personnel

```
public String identification() {
    return prenom + " " + nom;
}
```

Réponse à la question 3, ligneAnnuaire :

Dans Membre Personnel

```
public String ligneAnnuaire() {
    return identification() + "(" + email + "):";
}
```

Dans Membre Sous Contrat

```
public String ligneAnnuaire() {
    return super.ligneAnnuaire() + role;
}
```

Dans Stagiaire

```
public String ligneAnnuaire() {
    return super.ligneAnnuaire() + "en stage " + periode.toString() +
        ", tuteur = " + tuteur.identification();
}
```

Question 4. Proposez en Java des constructeurs paramétrés pour les membres du personnel. On initialisera l'ensemble des attributs présents dans le modèle, la période pour les stagiaires, mais pas le tuteur pour les stagiaires ni les apprentis pour les membres sous contrat.

Réponse à la question 4 :

Dans MembrePersonnel

```
public MembrePersonnel (String nom, String prenom, String courriel) {
    this.nom = nom;
    this.prenom = prenom;
    this.courriel = courriel;
}
```

Dans Stagiaire MembreSousContrat

```
public MembreSousContrat (String n, String p, String c, float s, float String r) {
    super(n, p, c);
    salaireNet = s;
    role = r;
}
```

Dans Stagiaire

```
public Stagiaire (String n, String p, String c, boolean g, PeriodeEmbauche pe) {
    super(n, p, c);
    graphique = g;
    periode = pe;
}
```

Question 5. Proposez les instructions Java permettant de déclarer et créer le membre sous contrat Jane Doe, d'adresse électronique jane.doe@superEtp.fr, chef de projet, de salaire net 2500 euros. Puis affichez sa ligne d'annuaire.

Réponse à la question 5 :

```
} MembrePersonnel {
a { MembreSousContrat } jane = new MembreSousContrat ("Doe", "Jane",
    "jane.doe@superEtp.fr", 2500.0f,
    "chef de projet");
    System.out.println (jane.ligneAnnuaire());
}
```

Question 6. Un stagiaire prépare un diplôme au sein d'un établissement de formation. Complétez le schéma ci-dessous pour modéliser le diplôme préparé par un étudiant. Un diplôme sera décrit par un nom et un niveau (en termes de nombre d'années après le bac, par exemple pour la licence, ce nombre est 3). Il est préparé au sein d'un établissement qui sera ici décrit par un nom, et une adresse. On veillera à bien isoler le concept d'établissement, et à utiliser des associations à bon escient.

On ne fera pas apparaître les accesseurs, ni les opérations (donc pas non plus les constructeurs). Réponse à la question 6 :

