

Nom :
Prénom :
Numéro d'étudiant :
Groupe :

Elements de correction pour le Contrôle continu

Tous documents sur support papier autorisés. Durée : 1h30.

L'ensemble des réponses sera à donner sur les feuilles d'énoncé. Ne pas dégrapher les feuilles.

Les listings sont fournis dans un document séparé.

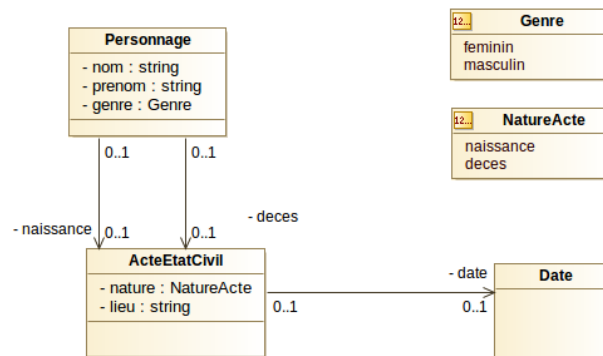
On se place dans le cadre d'une application recensant les personnages historiques français, à destination de l'enseignement de l'Histoire au primaire et au secondaire.

1 Personnages historiques

On propose une première classe pour les personnages historiques : `Personnage`. Un personnage historique a un nom et un prénom, un genre (masculin ou féminin) et deux actes d'état civil : l'un pour la naissance et l'autre pour le décès. Ces actes sont potentiellement non renseignés (non présents). Un acte d'état civil a une nature (pour l'instant on prévoit juste le décès et la naissance), une date, et un lieu (sous forme de chaîne, on y stocke juste la ville où a eu lieu l'acte).

On suppose également disposer d'une classe `Date` simplifiée, dont un squelette vous est donné, et d'une classe `UtilDate` contenant des méthodes utilitaires pour les dates. On dispose des énumérations `Genre` et `NatureActe` et d'une classe `ActeEtatCivil`.

On donne le diagramme de classes partiel suivant, dans lequel les méthodes ne sont pas représentées.



Question 1. Donnez en Java le code pour créer une instance de `Personnage`, de prénom et nom Victor Hugo, né le 26 février 1802 à Besançon (on rappelle que Victor Hugo est un homme). On ne s'intéressera pas ici au décès (juste à titre culturel : Victor Hugo est mort le 22 mai 1885 à Paris).

Réponse à la question 1 :

```
Personnage p=new Personnage("Hugo", "Victor", Genre.masculin);
p.setNaissance(new ActeEtatCivil(NatureActe.naissance, "26/02/1802", "Besancon"));
```

Question 2. Donnez en Java le code de l'accessor en écriture `setDeces` de la classe `Personnage`. Si l'acte d'état civil en paramètre respecte les conditions suivantes, alors on l'affecte au décès. Sinon, on laisse inchangé le décès.

Conditions :

- l'acte est bien un acte de décès
- l'acte de décès a une date postérieure à celle de l'acte de naissance (si celui-ci est connu), autrement dit le décès a lieu après la naissance.

On pourra utiliser les méthodes de la classe `Date` sans les définir.

```
public void setDeces(ActeEtatCivil deces) {
    if (naissance!=null && deces.getDateActe().estApres(naissance.getDateActe()) && deces.
        getNature()==NatureActe.deces){
        this.deces = deces;
    }
}
```

Question 3. Donnez en Java le code d'une méthode `estContemporainDe` (de la classe `Personnage`) qui retourne vrai si et seulement si un personnage est contemporain d'un autre donné. Deux personnages p1 et p2 sont dits contemporains s'ils ont en commun une période de vie au delà de leur 20 ans, qui est supérieure ou égale à 5 ans, c'est-à-dire si le nombre d'années séparant $\max(\text{naissance de p1, naissance de p2})$ de $\min(\text{décès de p1, décès de p2})$ est supérieur ou égal à 25. On pourra utiliser les méthodes de la classe `UtilDate` sans les définir au préalable. On suppose les actes de naissance et de décès saisis.

```
public boolean estContemporainDe(Personnage p){
    return deces!=null && naissance!=null && p.getNaissance()!=null && p.getDeces()!=null
        && (UtilDate.max(naissance.getDateActe(), p.getNaissance().getDateActe()).
            nbAnneesSeparantDe(UtilDate.min(deces.getDateActe(), p.getDeces().getDateActe()))
            -20)>=5;
}
```

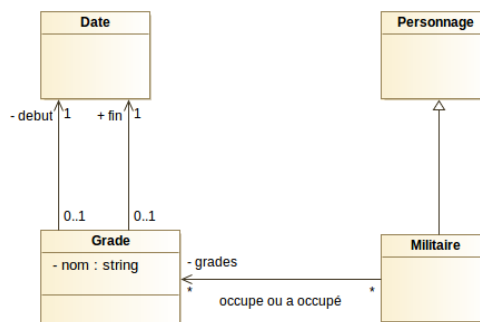
Question 4. Si on voulait ajouter à la classe `Personnage` tous les autres actes d'état civil le concernant (les mariages notamment), comment proposeriez-vous de procéder ? Quels attributs et méthodes Java faudrait-il rajouter ?

On rajouterait une association de `Personnage` vers `ActeEtatCivil`, de cardinalité `*` côté `ActesEtatCivil`. On pourrait enlever les associations pour la naissance et le décès, mais cela dépend de l'usage. Il faudrait ajouter en Java une collection pour ces actes (`ArrayList<ActesEtatCivil> autresActes` par exemple) et une méthode permettant d'ajouter ces actes, prenant en paramètre un acte et l'ajoutant à la collection, probablement en maintenant ordonnée par date cette collection. On rajoute autant de littéraux d'énumération que nécessaire dans `NatureActe`.

2 Personnages historiques et militaires

On introduit des personnages historiques particuliers : les militaires. D'autres personnages particuliers seront introduits plus tard.

Les militaires occupent ou ont occupé des grades. Les grades sont triés par ordre décroissant de date : le dernier grade occupé est en tête des grades. Un grade a une date de début et une date de fin, ainsi qu'un nom (par exemple lieutenant, ou général). On notera qu'il s'agit ici d'une chaîne de caractères mais qu'il aurait été plus judicieux de créer une énumération. Un diagramme de classes UML est proposé ci-dessous, ainsi qu'une portion de code (voir document avec les listings).



Question 5. On souhaite que les personnages historiques disposent d'une méthode `civilite` qui retourne sous forme de chaîne la civilité. Dans le cas général, la civilité est "M." (pour monsieur) s'il s'agit d'un personnage masculin, ou "Mme" (pour madame) s'il s'agit d'un personnage féminin. Pour les militaires, en revanche la civilité est le nom du dernier grade occupé. Si aucun grade n'est renseigné pour un militaire, alors la civilité sera celle d'un personnage normal ("M." ou "Mme" selon le genre). Écrivez le code Java nécessaire pour l'implémentation de la civilité, en précisant bien dans quelle classe vous mettez quel code.

```

// Dans Personnage
public String civilite () {
    if (genre==Genre.masculin) return "M. ";
    else return "Mme ";
}

// Dans Militaire
public String civilite () {
    if (!grades.isEmpty()) {
        return grades.get(0).getNom()+" ";
    } else {
        return super.civilite();
    }
}
  
```

Question 6. Tous les personnages historiques disposent d'une méthode `etatcivil`, qui retourne une chaîne de caractère composée de la civilité, du nom et du prénom (séparés par des espaces). Par exemple pour Victor Hugo l'état civil est "M. Hugo Victor" et pour le général De Gaulle l'état civil est "général De Gaulle Charles".

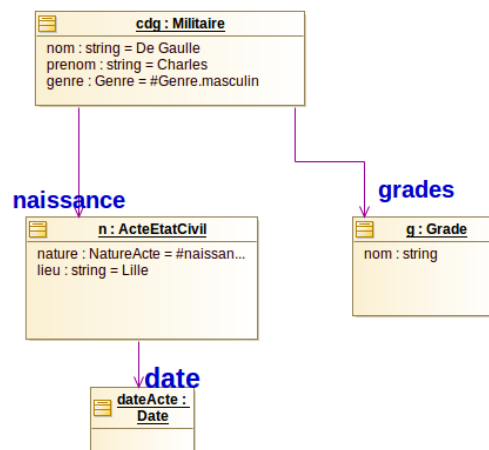
Écrivez le code Java nécessaire pour l'implémentation de l'état civil, en précisant bien dans quelle classe vous mettez quel code.

```
// Dans Personnage
public String etatCivil() {
    return civilite()+nom+" "+prenom;
}
```

Question 7. Tous les personnages historiques ont une sphère d'influence, on veut mettre en place une méthode la calculant. La sphère d'influence d'un personnage est l'ensemble des personnages qu'il a influencés. Dans le cas général, on ne peut pas déterminer quelle est la sphère d'influence d'un personnage historique. Pour un militaire, la sphère d'influence est l'union de tous les subordonnés directs d'un militaire pendant sa carrière (on suppose que l'information sur les subordonnés directs est disponible dans la classe `Grade` même si ce n'est pas le cas actuellement). Expliquez comment procéder pour la mise en place du calcul de la sphère d'influence. On ne donnera pas de code, juste une explication à grands traits.

On ajoute une méthode abstraite `sphereinfluence() :Personnage[*]` dans la classe `Personnage`. On rend donc `Personnage` abstrait. On définit `sphereinfluence` dans la classe `Militaire` comme l'union des subordonnés directs (en parcourant tous les grades occupés par le militaire).

Question 8. Donnez un diagramme d'objets représentant un militaire de genre masculin, nommé Charles de Gaulle, né le 22 novembre 1890 à Lille, dont le seul grade renseigné est celui de général (sans dates de début ni de fin renseignées).

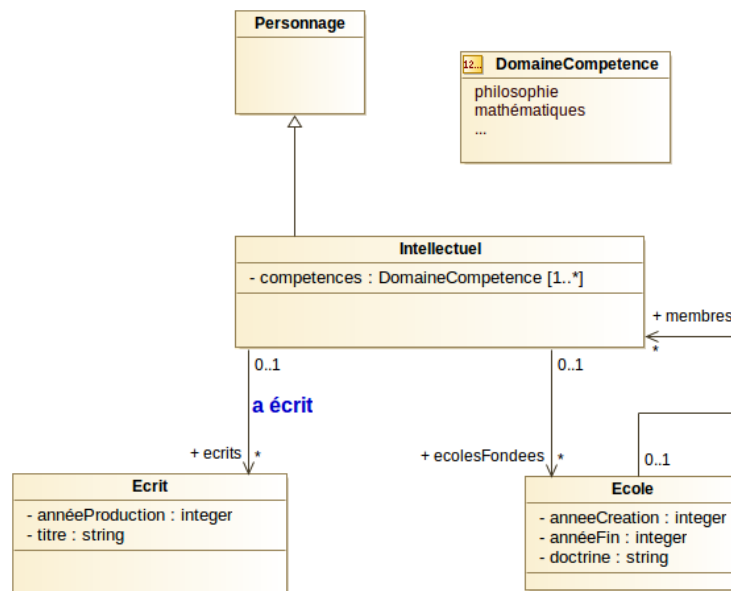


On ne peut pas vraiment représenter les dates car on ne sait pas comment elles sont modélisées.

3 Personnages historiques et intellectuels

On souhaite maintenant modéliser les intellectuels qui sont des personnages historiques particuliers. Les intellectuels ont un ou plusieurs domaine(s) de compétence (philosophie, mathématiques, etc). Ils peuvent produire des écrits. Un écrit a une année de production, et un titre. Les intellectuels peuvent fonder des écoles. Par exemple, Pythagore a fondé l'école pythagoricienne. Une école a une année de création et une année de fin. Une école a des membres, qui sont des intellectuels. Une école peut avoir une doctrine (sous forme de chaîne de caractères).

Question 9. Proposez un modèle de classes UML pour les intellectuels, sans y intégrer les méthodes.



Il y avait beaucoup d'autres façons de modéliser !