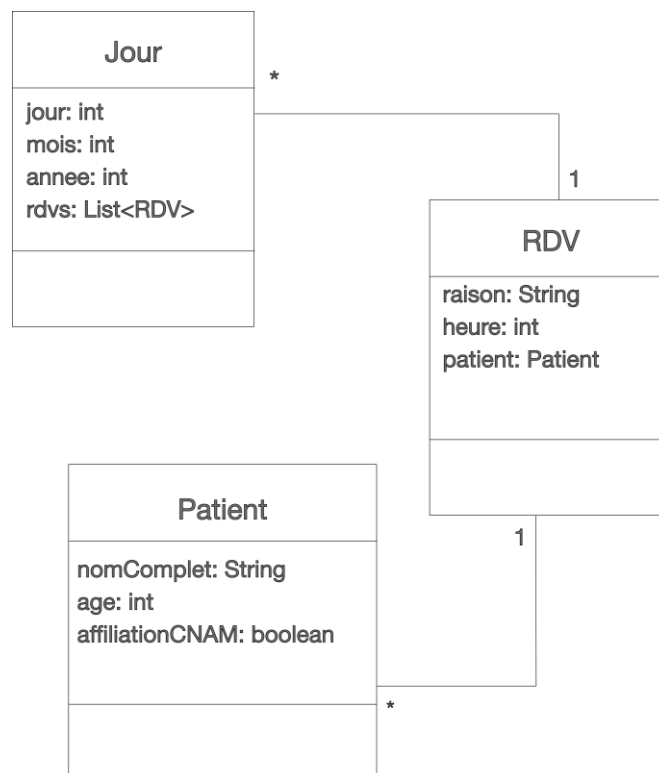


<p>a</p> 	<p align="center">Année universitaire : 2023 - 2024</p> <p align="center">Examen Ecrit</p>	
<p>Module : Conception Orienté Objet et Programmation Java</p> <p>Classes : 1CINFO</p> <p>Durée : 1H30</p> <p>Nombre de pages : 2</p>	<p>Documents autorisés : Oui</p> <p>Internet autorisé : Non</p> <p>Date : 13/05/2024</p> <p>Heure : 18H30</p>	

NB : Après avoir tester tous les TODO sur votre machine dans le projet Template, veuillez répondre sur la feuille d'examen. Vous n'êtes pas obliges de réécrire la signature des méthodes.

Problème

Durant cet examen, nous essayons de développer une application qui vise la prise de rendez-vous pour les patients. Pour ce faire, vous trouverez ci-dessous le diagramme de classe pour modéliser les parties prenantes dans le projet :



NB : Un rendez-vous dure une heure et 2 rendez-vous ne doivent pas avoir le même jour, mois, année et heure. Toutes les modifications doivent se faire sur la classe **AgendaService**.

Pour venir en aide à la secrétaire dans son travail quotidien, nous se trouvons dans le besoin de :

1. Implémenter la méthode ``void ajouterRDV(RDV rdv, Jour j)`` qui permet d'ajouter un rendez-vous dans la liste des rendez-vous d'un jour j (2 pts)
2. Implémenter la méthode ``void annulerRDV(RDV rdv, Jour j)`` qui permet d'annuler un rendez-vous dans la liste des rendez-vous d'un jour j en le supprimant (2 pts)
3. Implémenter la méthode ``Set<RDV> trierRecupererRDV(Jour j)`` qui permet de trier la liste des rendez-vous suivant l'heure du rendez-vous puis les retourner (2 pts)
4. Implémenter la méthode ``void afficherProchainPatient(Jour j, int heure)`` qui permet d'afficher le patient qui passera dans le jour j et a l'heure passée en paramètres. Si personne n'a pris de rendez-vous à cette heure, afficher "Créneau vide" (2 pts)
5. Implémenter la méthode ``int afficherBilanQuotidien(Jour j)`` qui permet de retourner le bilan financier quotidien du docteur sachant qu'une visite coute 70 TND (2 pts)
6. Implémenter la méthode ``boolean trouverPatient(Jour j, String nom)`` qui permet de retourner si oui ou non un patient a pris un rendez-vous le jour j avec **Stream** (2 pts)
7. Implémenter la méthode ``Set<Patient> retournerAffilieCNAM(Jour j)`` qui permet de retourner la liste des patients affiliés CNAM pour le jour j avec **Stream** (2 pts)
8. Implémenter la méthode ``int nombrePatients(List<Jour> jours)`` qui permet de retourner le nombre de patient de la liste passée en paramètre avec **Stream** (2 pts)
9. Implémenter la méthode ``Map<Jour, Integer> nombrePatientparJour(List<Jour> jours)`` qui permet de retourner le nombre de patient par jour de la liste des jours passée en paramètres (2 pts)
10. Implémenter la méthode ``List<RDV> prioriserRDV(Jour j)`` qui permet de trier la liste des rendez-vous suivant les cas prioritaires. Un patient est considéré prioritaire si son âge est supérieur à 65 ans. (2 pts)