**TD1. Diagrammes de classes**

**Question 12** *Définissez la classe UML représentant un étudiant, caractérisé, entre autres, par un identifiant, un nom, un prénom et une date de naissance.*

**Question 13** *Définissez la classe UML représentant un enseignant, caractérisé, entre autres, par un identifiant, un nom, un prénom et une date de naissance.*

**Question 14** *Définissez la classe UML représentant un cours, caractérisé par un identifiant, un nom, le nombre d’heures de cours magistral, le nombre d’heures*

*de travaux dirigés et un nombre d’heures de travaux pratiques que doit suivre un étudiant.*

**Question 15** *Définissez les associations « dispenser » et « responsabilité » qui peuvent exister entre un enseignant et un cours.*

**Question 16** *Définissez la classe UML représentant un groupe d’étudiants en utilisant les associations.*

**Question 17** *Définissez l’association possible entre un groupe d’étudiants et un cours.*

**Question 18** *Pensez-vous qu’il soit possible de définir un lien d’héritage entre les classes UML représentant respectivement les étudiants et les enseignants ?*

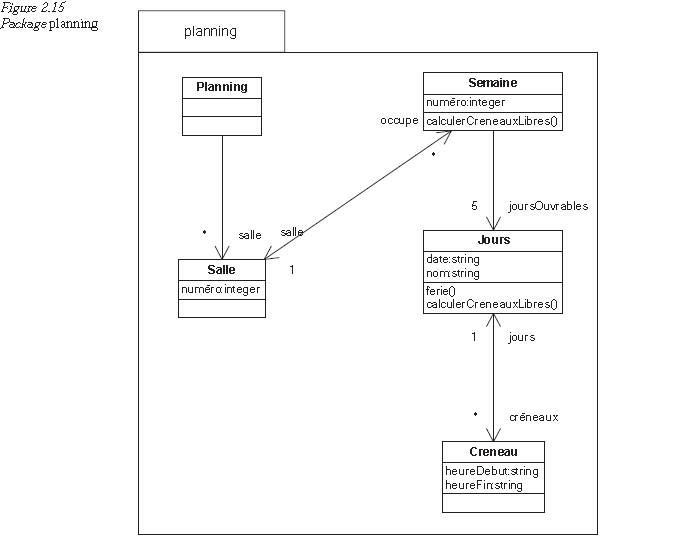
**Question 19** *Pensez-vous qu’il soit possible de définir un lien d’héritage entre les classes UML représentant respectivement les étudiants et les groupes d’étudiants ?*

**Question 20** *On nomme coursDeLEtudiant() l’opération permettant d’obtenir l’ensemble des cours suivis par un étudiant. Positionnez cette opération dans une classe, puis précisez les paramètres de cette opération, ainsi que les modifications à apporter aux associations préalablement identifiées pour*

*que votre solution soit réalisable.*

**Question 21** *On nomme coursDeLEnseignant() l’opération permettant d’obtenir l’ensemble des cours dans lesquels intervient un enseignant. Positionnez cette opération dans une classe, puis précisez les paramètres de cette opération, ainsi que les modifications à apporter aux associations préalablement identifiées pour que votre solution soit réalisable.*

**Question 22** *Expliquez le diagramme de classes représenté à la* ***figure2.15.***



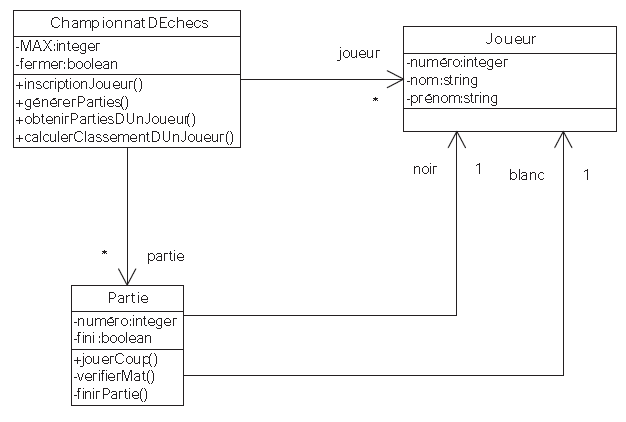
**Question 23** *Positionnez toutes vos classes (Etudiant, Enseignant, Cours, GroupeEtudiant) dans un package nommé Personnel.*

**Question 24** *Liez vos classes pour faire en sorte qu’un créneau soit lié à un cours !*

**Question 25** On suppose que le système est en milieu distribué. Les packages planning et personnel sont dans 2 sous-systèmes différents. Représentez le déploiement du système.

**TD3. Diagrammes de séquence UML**

L’application ChampionnatEchecs, qui devra permettre de gérer le déroulement d’un championnat d’échecs est actuellement en cours de développement. L’équipe de développement n’a pour l’instant réalisé qu’un diagramme de classes de cette application *(****voir* *figure****).*



***figure 6.8***

La classe ChampionnatDEchecs représente un championnat d’échecs. Un championnat se déroule entre plusieurs joueurs (voir classe Joueur) et se joue en plusieurs parties (voir classe Partie). La propriété MAX de la classe ChampionnatDEchecs correspond au nombre maximal de joueurs que le championnat peut comporter. La propriété fermer permet de savoir si le championnat est fermé ou si de nouveaux joueurs peuvent s’inscrire.

ChampionnatDEchecs possède les opérations suivantes :

• inscriptionJoueur(in nom:string, in prénom:string) : integer permettant d’inscrire un nouveau joueur dans le championnat si le nombre de joueurs inscrits n’est pas déjà égal à MAX et si le championnat n’est pas déjà fermé. Si l’inscription est autorisée, cette opération crée le joueur et retourne son numéro dans le championnat.

• générerPartie() : permet de fermer le championnat et de générer toutes les parties nécessaires.

• obtenirPartieDUnJoueur(in numéro :integer) : Partie[\*] : permet d’obtenir la liste de toutes les parties d’un joueur (dont le numéro est passé en paramètre).

• calculerClassementDUnJoueur(in numéro :interger) : integer permettant de calculer le classement d’un joueur (dont le numéro est passé en paramètre) pendant le championnat.

La classe Partie représente une des parties du championnat. La classe Partie est d’ailleurs associée avec la classe ChampionnatDEchecs, et l’association précise qu’un championnat peut contenir plusieurs parties. Une partie se joue entre deux joueurs. Un joueur possède les pièces blanches et commence la partie alors que l’autre joueur possède les pièces noires. Les associations entre les classes Partie et Joueurs précisent cela. La propriété numéro correspond au numéro de la partie (celui-ci doit être unique). La propriété fini permet de savoir si la partie a déjà été jouée ou pas.

La classe Partie possède les opérations suivantes :

• jouerCoup( coup:string) : permet de jouer un coup tant que la partie n’est pas finie. Le traitement associé à cette opération fait appel à l’opération vérifierMat afin de savoir si le coup joué ne met pas fin à la partie. Si tel est le cas, l’opération

finirPartie est appelée.

• vérifierMat() : boolean permettant de vérifier si la position n’est pas mat.

• finirPartie : permet de préciser que la partie est finie. Il n’est donc plus possible

de jouer de nouveaux coups.

La classe Joueur représente les joueurs du championnat. La classe Joueur est d’ailleurs associée avec la classe ChampionnatDEchecs, et l’association précise qu’un championnat peut contenir plusieurs joueurs. La propriété numéro correspond au numéro du joueur (celui-ci doit être unique). Les propriétés nom et prénom permettent de préciser le nom et le prénom du joueur.

Un championnat d’échecs se déroule comme suit :

• Un administrateur de l’application crée un championnat avec une valeur MAX.

• Les participants peuvent s’inscrire comme joueurs dans le championnat.

• L’administrateur crée l’ensemble des parties.

• Les participants, une fois inscrits, peuvent consulter leur liste de parties.

• Les participants, une fois inscrits, peuvent jouer leurs parties. Nous ne nous intéressons qu'aux coups joués par chacun des deux joueurs. Nous ignorons l'initialisation de la partie (identification du joueur qui a les pions blancs et donc qui commence la partie).

• Les participants peuvent consulter leur classement.

Dans les questions suivantes, nous allons spécifier des exemples d’exécution de ChampionnatDEchecs avec des diagrammes de séquence.

**Question 52** *Comment modéliser les administrateurs et les participants ?*

**Question 53** *Représentez par un diagramme de séquence le scénario d’exécution correspondant à la création d’un championnat et à l’inscription de deux joueurs. Vous assurerez la cohérence de votre diagramme avec le diagramme de classes fourni à la figure 6.8.*

**Question 54** *Représentez par un diagramme de séquence le scénario d’exécution correspondant à la création de l’ensemble des parties pour le championnat créé à la question 53. Vous assurerez la cohérence de votre diagramme avec le diagramme de classes fourni à la figure 6.8.*

**Question 55** *Représentez par un diagramme de séquence le scénario d’exécution correspondant au déroulement de la partie d’échecs entre deux joueurs.*

*Vous pouvez considérer une partie qui se termine en quatre coups. Vous assurerez la cohérence de votre diagramme avec le diagramme de classes fourni à la figure 6.8.*

**Question 56** *Est-il possible de générer automatiquement le code d’une opération de cette application à partir de plusieurs diagrammes de séquence ?*

**Question 57** *Est-il possible de construire des diagrammes de séquence à partir du code d’une application ?*

*Une équipe de développement souhaite réaliser une application Calculus qui permet à des utilisateurs d’effectuer des opérations arithmétiques simples sur des entiers : addition, soustraction, produit, division. Cette application a aussi une fonction mémoire qui permet à l'utilisateur de stocker un nombre entier qu'il pourra ensuite utiliser pour n'importe quelle opération. Les opérations peuvent directement s'effectuer sur la mémoire.*

*L’utilisateur se connecte et ouvre ainsi une nouvelle session. Puis, dans le cadre d’une session, l’utilisateur peut demander au système d’effectuer une suite d’opérations.*

**Question 58** *Utilisez des diagrammes de séquences pour représenter les différents scénarios d’exécution du service Calculus.*

**Question 59** *Pour chacune des instances apparaissant dans votre diagramme de classes, créez la classe correspondante.*

*Nous souhaitons réaliser le diagramme de cas d’utilisation du championnat d’échecs présenté dans ce TD2.*

**Question 79.** *Donnez la liste des acteurs du système.*

**Question 80.** *Donnez la liste des cas d’utilisation du système en les liant aux acteurs.*

**Question 81.** *Donnez le diagramme de cas d’utilisation du système.*

**Question 82.** *Reprenez les diagrammes de séquence réalisés au TD6 pour l’application de championnat d’échecs, et expliquez comment les relier au diagramme de cas d’utilisation obtenu à la question précédente.*

**Question 83.** *En le diagramme de séquence système obtenu à partir de la question 81.*