Rappels de programmation et modélisation par objets
Les classes imbriquées

Objectifs :

- Reprise rapide des concepts de classe, énumération, attributs d'instances vs de classe, associations UML, utilisation de collections Java.
- Mise en œuvre et utilisation de classes imbriquées statique ou non statique interne.

1 Cercle sportif

Un cercle sportif d'une grande ville atteint la taille critique au delà de laquelle la gestion de ses activités par feuille excel devient impossible. Ce cercle sportif décide donc de (faire) développer un logiciel lui permettant de gérer plus facilement les activités sportives qu'il organise. Ces activités sont principalement des cours de différents sports, et plus marginalement l'organisation d'événements ponctuels. On se limitera ici aux cours. Plusieurs activités sont proposées : natation, course à pieds, gym, pilates, etc. Chaque cours propose l'enseignement d'une de ces activités, à raison d'un et d'un seul créneau par semaine, sur la même période de temps : les semaines d'activité sont fixées annuellement et pour toutes les activités (classiquement, les activités ont lieu de septembre à juin, sauf pendant les semaines de vacances scolaires). On crée donc un planning sur une semaine, qui est appliqué à l'identique pour chacune des semaines d'activité. Chaque année, en juillet, les données de l'année courante sont archivées dans une base de donnée dédiée, et le processus de planification et d'inscription est réinitialisé. Pour simplifier la planification et dans la mesure où une année ressemble à une autre, on garde d'une année à l'autre la même organisation globale (mêmes cours aux mêmes endroits et aux mêmes créneaux, avec les mêmes enseignants), on vide juste les listes d'inscrits. Cette planification de base pourra ensuite être modifiée en fonction de la demande, des disponibilités des enseignants, etc. Les cours ont des durées et des prix différents. Un cours n'ouvre à la rentrée de septembre que s'il y a un nombre minimum d'inscrits. Pour des raisons de sécurité, il y a également pour chaque cours un nombre maximal d'inscrits. Les inscriptions (et réinscriptions) sont ouvertes à la fin de la saison. Chaque nouvel ou ancien adhérent s'inscrit aux cours auxquels il souhaite assister. Il se voit notifier du prix dont il devra s'acquitter au maximum à la deuxième séance. L'inscription n'est validée qu'après le paiement. La validation déclenche la demande de licence auprès de la fédération sportive correspondante. Le premier cours est plein tarif, les cours suivants sont à 10% de réduction pour un même adhérent. Un adhérent a un numéro unique qu'il garde au cours de ses ré-adhésions. L'adhésion en elle même au cercle requiert une cotisation annuelle.

Au cours du processus de planification et d'inscription, de nouveaux cours peuvent être créés, si la demande est forte dans une discipline par exemple. Juste avant la rentrée de septembre, le personnel du cercle consulte la liste des cours dont le seuil minimal n'est pas atteint. Pour chacun, les inscrits sont déplacés vers d'autres cours (à condition que la contrainte de capacité maximale soit respectée) de même discipline et d'horaire similaire. Ces déplacements d'un cours à un autre nécessitent de demander l'accord de l'adhérent. Une procédure doit permettre d'éditer facilement la liste des adhérents pour lesquels on attend soit un paiement, soit un acquittement après déplacement d'un créneau. Un adhérent dont le créneau a été déplacé peut refuser le nouveau créneau. S'il avait déjà payé, il est alors remboursé. Les créneaux ont lieu du lundi au samedi, de 7h à 22h.

Le cercle sportif dispose d'un ensemble de lieux pour ses activités : piscines, salles de sport, stades, etc. Chaque cours a lieu dans un lieu unique, tout au long de l'année (marginalement, un cours peut avoir besoin d'autres locaux, mais alors la demande de locaux se fait en dehors de la gestion informatique du cercle). La plupart des lieux sont loués aux collectivités publiques ou privées qui les possèdent, le cercle n'en dispose donc que pour certains créneaux. Chaque lieu dispose d'un ensemble d'équipements (par exemple : bassin 25m ou 50m, steps, tapis, etc). Lors de la création d'un nouveau cours, on veille à l'affecter à un lieu dont on dispose lors du créneau du cours, et on vérifie que le matériel est adéquat, c'est-à-dire que le lieu dispose du matériel requis par le cours.

Le cercle sportif emploie des moniteurs pour chacune des disciplines enseignées. Chaque moniteur est qualifié pour enseigner une ou plusieurs disciplines. Un cours est dispensé par un et un seul moniteur.

1.1 Questions pour se remémorer les bases

Question 1. On modélise les lieux par une énumération. Un lieu est disponible à certains créneaux pour le cercle sportif. Chaque lieu a un nom (qui n'a pas forcément le format d'un littéral, par exemple il peut contenir des espaces comme dans "stade de foot"). Vous remplirez l'énumération Lieu à votre convenance. On veut pouvoir savoir pour chaque lieu s'il est disponible pour le cercle sportif pour un créneau donné.

Un créneau est représenté par : un jour de la semaine d'une part (une énumération), et une heure de début et de fin d'autre part. Les créneaux sont munis de méthodes permettant de déterminer si un créneau en chevauche un autre, ou est inclus dans un autre.

On développera une classe Heure simple, où les heures et les minutes sont des entiers contraints. La granularité la plus fine est de 5 minutes.

Vous réfléchirez bien aux énumérations : pouvez-vous ajouter un constructeur à l'énumération permettant d'initialiser le nom des littéraux ? Cela signifie t-il que l'on peut ajouter dynamiquement des littéraux par appel au constructeur ?

On mettra dans chaque classe et énumération une méthode toString de telle manière à ce que pour les lieux la chaîne retournée soit le nom du lieu, et pour les créneaux une chaîne de format `JS HH:MM - HH:MM` avec JS le jour de la semaine (lundi, mardi, etc), HH une heure sur deux digits, MM une heure sur deux digits. Pour le cas où les heures sont incorrectes, `HH:MM` sera remplacé par `ERREUR`.

Proposez une modélisation et une implémentation de ces classes et énumérations. Testez avec soin.

Question 2. On s'intéresse à la classe Adherent. On limitera les informations sur l'état civil de l'adhérent à son nom pour simplifier. Un adhérent peut avoir acquitté sa cotisation pour l'année en cours ou pas. Un adhérent se voit attribuer un numéro unique à sa création dans le système. On décide d'attribuer comme numéro à un nouvel adhérent le nombre d'adhérents ayant déjà été créés majoré de 1. On maintient la dernière année pour laquelle une adhésion a été payée de manière à pouvoir supprimer l'adhérent au bout de 5 ans sans adhésion. Modélisez et implémentez une première version de la classe Adherent avec uniquement : ses attributs, son constructeur (appelé lors d'une nouvelle inscription), et une méthode reAdhesion qui est appelée quand un adherent existant paye sa cotisation pour l'année en cours.

1.2 Une première classe interne

Question 3. Dans l'hypothèse où on n'aurait pas besoin de la classe Heure en dehors de la classe créneau, on envisage d'utiliser une classe imbriquée. On ne souhaite pas particulièrement partager les heures d'un créneau à l'autre. Mettez en place votre solution (on pourra la mettre en place dans une duplication de la classe Creneau, de manière à garder les deux solutions dans le projet).

2 Classes internes

Question 4. Créez une classe liste chaînée. Cette classe possèdera un attribut privé référençant la racine de la liste, qui sera de type Node. La classe Node sera une classe interne à la classe liste de manière à faciliter l'accès des noeuds à la liste et de la liste aux noeuds. Un noeud a un nom, et connait son successeur. Ecrire de quoi créer un noeud sans successeur, ajouter un entier en tête de liste, connaître la taille de la liste, et afficher la liste.

Question 5. Ajoutez à la méthode de la méthode d'ajout de noeud une assertion vérifiant que la taille de la liste après l'ajout est égale à la taille avant l'ajout, augmentée de 1.

Question 6. Ajoutez une méthode renverser qui retourne une nouvelle liste qui est la liste receveur retournée.

3 Pour les plus rapides

Question 7. Terminez une version raisonnablement utilisable du cercle sportif : proposez un diagramme UML où vous ferez apparaître les classes, les attributs et les associations qui vous semblent judicieux, modélisez ensuite les méthodes qui vous paraissent importantes, et proposez-en une implémentation en Java.