

## Chapitre 4 : Diagrammes de séquences et de collaborations

Les exercices qui suivent ont pour but de permettre aux apprenants la maîtrise des concepts sous-jacents du chapitre intitulé Diagrammes de séquences et de collaborations, tant du point de vue de la technique (quelles notations, comment...) que de celui de l'utilisation (dans quel contexte, pourquoi...). Ils sont (en principe) originaux. Des corrigés sont disponibles.

### Partie I : Modélisation à l'aide de séquences

#### Exercice 4.1

Une association est composée de membres et d'un conseil d'administration de trente membres, dits actifs. Ce nombre peut évoluer en modifiant les statuts. L'administration de l'association est confiée à un bureau composé de membres actifs. Il est formé d'un président, de deux vice-présidents, d'un président d'honneur, d'un secrétaire et de son adjoint, d'un trésorier et de son adjoint. Toutes les fonctions doivent être attribuées pour que la préfecture accepte les statuts de l'association. Aucun membre ne peut cumuler plusieurs fonctions dans le bureau.

Le bureau est élu par le conseil d'administration, lui-même élu par l'assemblée générale, selon les règles fixées par les statuts de l'association. Actuellement, les statuts prévoient un renouvellement annuel du tiers des membres du conseil. Tous les ans, un tiers du conseil d'administration est donc déclaré sortant. Autrement dit, un membre élu l'année N est déclaré sortant l'année N + 3. L'élection d'un nouveau membre à ce conseil est soumis au vote à la majorité absolue lors de l'assemblée générale annuelle présidée par le président de l'association.

Le vote est licite si au moins les deux tiers des membres sont présents. Un membre actif ne se trouve que dans un tiers (le nombre de membres actifs est un multiple de trois). Il y a quatre possibilités de vote pour un scrutin donné : oui, non, blanc, abstention.

Représentez une élection par un diagramme de séquences, en précisant vos hypothèses de travail.

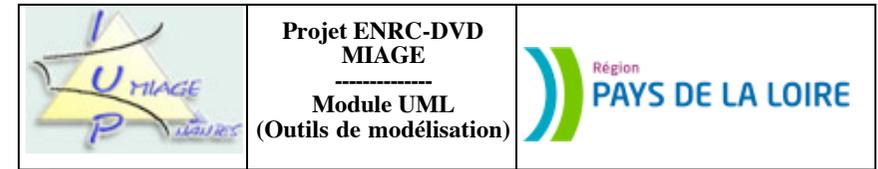
NB : cet énoncé est repris un peu plus loin (cf exercice 4.5), dans un autre contexte. Le diagramme de séquences est remplacé par un diagramme de collaborations. Une comparaison des deux réponses est ainsi possible, ce qui permet de voir les avantages et inconvénients de chacune des notations.

#### Exercice 4.2

Soient les deux scénarios suivants, écrits par un informaticien amateur d'objets :

a) Emprunt effectué par un adhérent : le scénario *Emprunt effectué par un adhérent* s'exécute suivant la séquence ci-après :

1. Un objet métier *unAdhérent* s'adresse à *unBibliothécaire*, objet externe au système, pour emprunter une œuvre : la méthode *demandeEmprunt* est impliquée dans cette action, elle appartient à la classe *Adhérent* et prend en paramètre un objet métier de type *Œuvre*.
2. *unBibliothécaire* vérifie que le nombre d'exemplaires empruntés par *unAdhérent* est inférieur à deux : la méthode d'instance *nombreExemplairesEmpruntés* de la classe *Adhérent* est déclenchée, elle renvoie le nombre d'exemplaires empruntés.
3. *unBibliothécaire* vérifie qu'il existe des exemplaires disponibles pour cette œuvre : pour cela, la méthode d'instance *exemplairesDisponibles* est envoyée à *uneŒuvre*, elle renvoie un exemplaire pouvant être prêté.
4. Le prêt pour l'exemplaire en question (*unPrêt*) peut alors être initialisé : la méthode d'instance *initialiserPrêt* est appelée, elle prend en paramètre l'objet *unExemplaire* associé.
5. *unPrêt* initialise *unTimer*, qui réveillera l'application quinze jours plus tard (voir point 1 ci-dessous).



6. *unPrêt* met à jour les données de l'objet *uneŒuvre*, en lui envoyant le message *décrémenterExemplairesDisponibles*.
7. *unPrêt* met à jour les données de l'objet *unAdhérent* en lui envoyant le message *incrémenterNbEmprunts*.
8. Enfin, *unBibliothécaire* attribue *unPrêt* à *unAdhérent* : la méthode *attribuer* de la classe *Adhérent* est déclenchée, elle prend en paramètre *unExemplaire*.

b) Cas de litige avec un adhérent : le scénario *Cas de litige avec un adhérent* s'exécute suivant la séquence ci-après :

1. *unTimer* se déclenche quinze jours après son initialisation (voir point 5 ci-dessus), il notifie le litige à *unBibliothécaire*.
2. En réponse, *unBibliothécaire* met à jour *unPrêt* en lui déclenchant la méthode d'instance *miseEnLitige* de la classe *Prêt*.
3. *unPrêt* déclenche alors la méthode *envoyerAvertissement* de la classe *Prêt*.
4. *unBibliothécaire* envoie à *unAdhérent* le message *imprimerCourierLitige*.
5. *unBibliothécaire* interdit alors l'emprunt de nouveaux exemplaires à *unAdhérent*, en déclenchant la méthode d'instance *interdireEmprunt* de la classe *Adhérent*.

Représentez sous forme graphique ces deux scénarios. Les paramètres seront indiqués. Les synchronisations ne le seront pas. La notation utilisée sera celle des diagrammes de séquences.

NB : cet énoncé est repris un peu plus loin (cf exercice 4.6), dans un autre contexte. Le diagramme de séquences est remplacé par un diagramme de collaborations. Une comparaison des deux réponses est ainsi possible, ce qui permet de voir les avantages et inconvénients de chacune des notations.

#### Exercice 4.3

Mme V. est mandataire occasionnelle dans la société Peter GOLD, spécialisée dans la vente directe de bijoux fantaisie. Cela signifie que Mme V. se déplace à domicile, chez une personne appelée *hôtesse*, et qu'elle y effectue une présentation des bijoux à des personnes appelées *prospects*, réunies chez l'hôtesse. Ces réunions sont sans obligation d'achat. Une prospect peut ou pas du tout intéressée restera prospect. Elle peut aussi devenir *cliente*, en passant commande d'un ou de plusieurs bijoux. Intéressons-nous, dans un premier temps, aux différents types de personnes que Mme V. est amenée à cotoyer. Nous distinguons des prospects, des clientes, des hôtesse et des hôtesse potentielles (encore appelées *candidates* par Mme V.). Donnons-en une définition :

- Prospect : personne assistant à une réunion, sans l'avoir organisée, sans acheter de bijoux, sans se porter volontaire pour l'organisation d'une prochaine réunion.
- Cliente : personne assistant à une réunion et achetant un ou plusieurs bijoux.
- Hôtesse : personne organisant une réunion.
- Candidate : personne se portant volontaire pour l'organisation d'une prochaine réunion.

Cette classification peut évoluer dans le temps, une même personne pouvant passer du statut de prospect à celui de cliente, puis à celui d'hôtesse, puis, encore plus tard, revenir à un statut de cliente... Ces statuts sont liés les uns aux autres par des relations. Ainsi, une hôtesse peut fort bien acheter des bijoux et donc se comporter aussi comme une cliente. Elle peut uniquement être hôtesse (sans jamais avoir été prospect et/ou cliente) et le rester. Une hôtesse est d'abord (ceci est une règle impérative) candidate. Elle peut, un jour, devenir cliente dans une autre réunion –organisée par une autre hôtesse– ou même prospect dans cette réunion. Mme V. souhaite conserver, pour chaque personne, son nom, son prénom et son adresse complète. Compte-tenu des risques d'homonymie, elle a accepté de numéroter chaque personne mémorisée dans la base de données. Elle peut, en outre, mémoriser, pour les hôtesse seulement (candidates incluses), leur numéro de téléphone. Une personne qui vient à une réunion et qui reste prospect ne sera pas dans la base de données. Une personne déjà dans la base (elle a assisté à une réunion antérieure, au cours de laquelle elle a acheté, ou bien elle a organisé une réunion antérieure) qui vient à une nouvelle réunion et qui se comporte là comme un prospect restera dans la base.

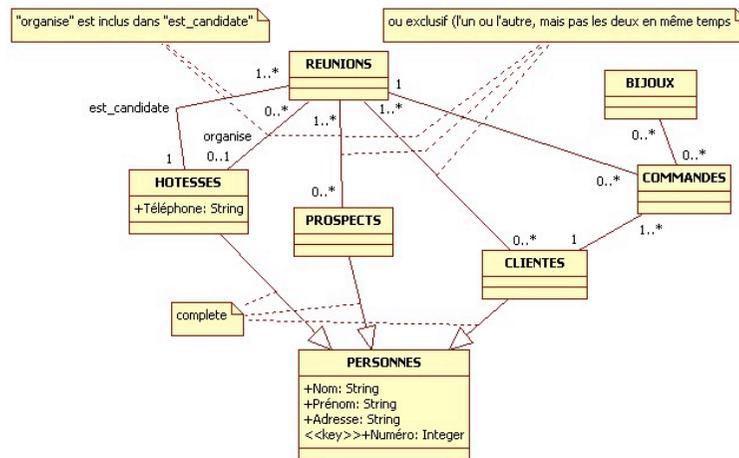
En fin de réunion, Mme V. calcule, avec l'hôtesse, le chiffre d'affaires réalisé. Elle demande également aux personnes présentes (prospects, clientes et hôtesse) s'il y a des candidates pour l'organisation de nouvelles réunions

(RV dans son jargon). Aux futures hôtesse, Mme V. propose une date éventuelle. Ceci n'est pas systématique, environ 50 % des propositions de prochaines réunions n'étant pas datées à cet instant-là mais plus tard (lors d'un appel téléphonique de Mme V.). Les éléments importants sur lesquels travaillent Mme V. et son hôtesse sont les suivants :

- le chiffre d'affaires (CA) réalisé,
- le nombre de personnes présentes,
- le nombre de commandes prises (c'est aussi le nombre de clientes),
- le nombre de RV pris.

Ces quatre informations déterminent la « rémunération » de l'hôtesse. Le nombre total de points obtenus est convertible par l'hôtesse en bijoux-cadeaux. L'hôtesse peut choisir immédiatement son ou ses cadeaux ou conserver ses points pour une prochaine réunion, ce qui lui permettra, par cumul de points, d'avoir de plus beaux cadeaux. Ainsi, une hôtesse ayant obtenu 5 points peut « acheter » un bijou d'une valeur de 4 points et garder un point pour la prochaine réunion qu'elle organisera ou à laquelle elle assistera.

Modélisez le travail que fait Mme V. en fin de réunion par un diagramme de séquences cohérent avec le diagramme de classes suivant, éventuellement complété par les classes nécessaires au travail de fin de réunion.



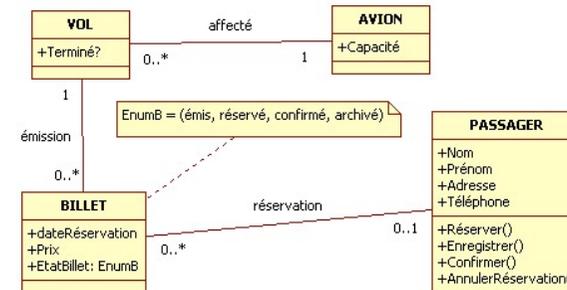
### Exercice 4.4

La compagnie aérienne AirForm nous propose de modéliser en UML une version réduite de son système d'information de réservation de billets de vols. Les vols sont planifiés à l'avance et on leur affecte un avion, un aéroport de départ et d'arrivée, une date de départ et une date d'arrivée. Chaque avion dispose d'une capacité en nombre maximal de passagers. Les billets sont émis pour chaque vole lors de la planification. Il n'y a pas de surbooking (de sur-réservation).

Les usagers achètent des billets. Cet achat entraîne une réservation (via le billet) pour le vol en question. On conserve les nom, prénom, adresse et téléphone des usagers ayant fait une réservation, ainsi que la date de réservation et le prix du billet. Lors de leur enregistrement (au moment du départ), les passager confirment leurs billets pour le vol enregistré. On mémorise cette confirmation.

Lorsque le vol est terminé, on archive les réservations qui le concernaient. Si le vol est annulé, les réservations sont annulées.

Voici le premier diagramme de classes qui a été établi :



Modélisez les traitements suivants par des diagrammes de séquences :

- réserver un billet sur un vol, en faisant l'hypothèse que le client connaît le vol et qu'il n'est pas nécessaire de faire une recherche de vol connaissant les dates de départ et d'arrivée, les lieux de départ et de destination ;
- annuler un vol (il faut donc annuler tous les billets du vol) par l'administrateur des vols.

Le diagramme de classes sera modifié en conséquence.

NB : cet énoncé est repris un peu plus loin (cf exercice 4.7), dans un autre contexte. Le diagramme de séquences est remplacé par un diagramme de collaborations. Une comparaison des deux réponses est ainsi possible, ce qui permet de voir les avantages et inconvénients de chacune des notations.

## Partie II : Modélisation à l'aide de collaborations

### Exercice 4.5

Une association est composée de membres et d'un conseil d'administration de trente membres, dits actifs. Ce nombre peut évoluer en modifiant les statuts. L'administration de l'association est confiée à un bureau composé de membres actifs. Il est formé d'un président, de deux vice-présidents, d'un président d'honneur, d'un secrétaire et de son adjoint, d'un trésorier et de son adjoint. Toutes les fonctions doivent être attribuées pour que la préfecture accepte les statuts de l'association. Aucun membre ne peut cumuler plusieurs fonctions dans le bureau.

Le bureau est élu par le conseil d'administration, lui-même élu par l'assemblée générale, selon les règles fixées par les statuts de l'association. Actuellement, les statuts prévoient un renouvellement annuel du tiers des membres du conseil. Tous les ans, un tiers du conseil d'administration est donc déclaré sortant. Autrement dit, un membre élu l'année N est déclaré sortant l'année N + 3. L'élection d'un nouveau membre à ce conseil est soumis au vote à la majorité absolue lors de l'assemblée générale annuelle présidée par le président de l'association.

Le vote est licite si au moins les deux tiers des membres sont présents. Un membre actif ne se trouve que dans un tiers (le nombre de membres actifs est un multiple de trois). Il y a quatre possibilités de vote pour un scrutin donné : oui, non, blanc, abstention.

Représentez une élection par un diagramme de collaborations, en précisant vos hypothèses de travail.

### Exercice 4.6

Soient les deux scénarios suivants, écrits par un informaticien amateur d'objets :

a) Emprunt effectué par un adhérent : le scénario *Emprunt effectué par un adhérent* s'exécute suivant la séquence ci-après :

1. Un objet métier *unAdhérent* s'adresse à *unBibliothécaire*, objet externe au système, pour emprunter une œuvre : la méthode *demandeEmprunt* est impliquée dans cette action, elle appartient à la classe *Adhérent* et prend en paramètre un objet métier de type *Œuvre*.
2. *unBibliothécaire* vérifie que le nombre d'exemplaires empruntés par *unAdhérent* est inférieur à deux : la méthode d'instance *nombreExemplairesEmpruntés* de la classe *Adhérent* est déclenchée, elle renvoie le nombre d'exemplaires empruntés.
3. *unBibliothécaire* vérifie qu'il existe des exemplaires disponibles pour cette œuvre : pour cela, la méthode d'instance *exemplairesDisponibles* est envoyée à *uneŒuvre*, elle renvoie un exemplaire pouvant être prêté.
4. Le prêt pour l'exemplaire en question (*unPrêt*) peut alors être initialisé : la méthode d'instance *initialiserPrêt* est appelée, elle prend en paramètre l'objet *unExemplaire* associé.
5. *unPrêt* initialise *unTimer*, qui réveillera l'application quinze jours plus tard (voir point 1 ci-dessous).
6. *unPrêt* met à jour les données de l'objet *uneŒuvre*, en lui envoyant le message *décrémenterExemplairesDisponibles*.
7. *unPrêt* met à jour les données de l'objet *unAdhérent* en lui envoyant le message *incrémenterNbEmprunts*.
8. Enfin, *unBibliothécaire* attribue *unPrêt* à *unAdhérent* : la méthode *attribuer* de la classe *Adhérent* est déclenchée, elle prend en paramètre *unExemplaire*.

b) Cas de litige avec un adhérent : le scénario *Cas de litige avec un adhérent* s'exécute suivant la séquence ci-après :

1. *unTimer* se déclenche quinze jours après son initialisation (voir point 5 ci-dessus), il notifie le litige à *unBibliothécaire*.
2. En réponse, *unBibliothécaire* met à jour *unPrêt* en lui déclenchant la méthode d'instance *miseEnLitige* de la classe *Prêt*.
3. *unPrêt* déclenche alors la méthode *envoyerAvertissement* de la classe *Prêt*.
4. *unBibliothécaire* envoie à *unAdhérent* le message *imprimerCourierLitige*.
5. *unBibliothécaire* interdit alors l'emprunt de nouveaux exemplaires à *unAdhérent*, en déclenchant la méthode d'instance *interdireEmprunt* de la classe *Adhérent*.

Représentez sous forme graphique ces deux scénarios. Les paramètres seront indiqués. Les synchronisations ne le seront pas. La notation utilisée sera celle des diagrammes de collaborations.

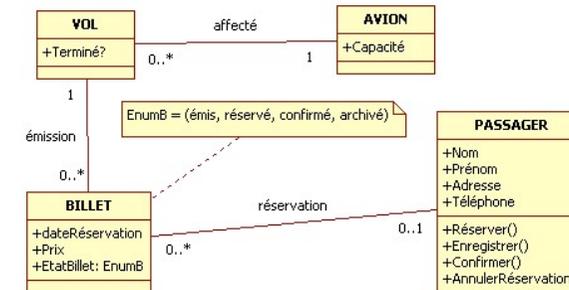
### Exercice 4.7

La compagnie aérienne AirForm nous propose de modéliser en UML une version réduite de son système d'information de réservation de billets de vols. Les vols sont planifiés à l'avance et on leur affecte un avion, un aéroport de départ et d'arrivée, une date de départ et une date d'arrivée. Chaque avion dispose d'une capacité en nombre maximal de passagers. Les billets sont émis pour chaque vole lors de la planification. Il n'y a pas de surbooking (de sur-réservation).

Les usagers achètent des billets. Cet achat entraîne une réservation (via le billet) pour le vol en question. On conserve les nom, prénom, adresse et téléphone des usagers ayant fait une réservation, ainsi que la date de réservation et le prix du billet. Lors de leur enregistrement (au moment du départ), les passager confirment leurs billets pour le vol enregistré. On mémorise cette confirmation.

Lorsque le vol est terminé, on archive les réservations qui le concernaient. Si le vol est annulé, les réservations sont annulées.

Voici le premier diagramme de classes qui a été établi :



Modélisez les traitements suivants par des diagrammes de collaborations :

- réserver un billet sur un vol, en faisant l'hypothèse que le client connaît le vol et qu'il n'est pas nécessaire de faire une recherche de vol connaissant les dates de départ et d'arrivée, les lieux de départ et de destination ;
- annuler un vol (il faut donc annuler tous les billets du vol) par l'administrateur des vols.

## Partie III : Cohérence entre séquences, collaborations et classes

### Exercice 4.8

Soient les diagrammes suivants, extraits de la spécification d'un cas de facturation.

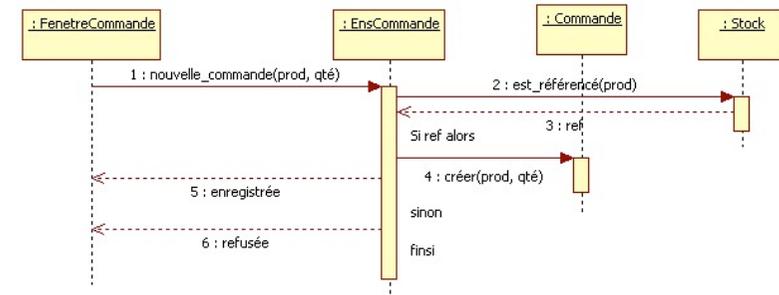


Figure n° 1 : diagramme de séquences, nouvelle commande

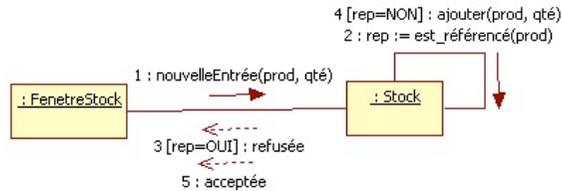


Figure n° 2 : diagramme de collaborations, nouvelle entrée

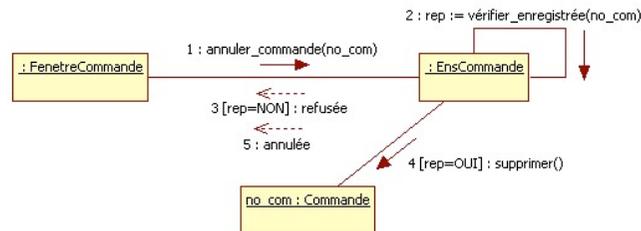


Figure n° 3 : diagramme de collaborations, annuler une commande

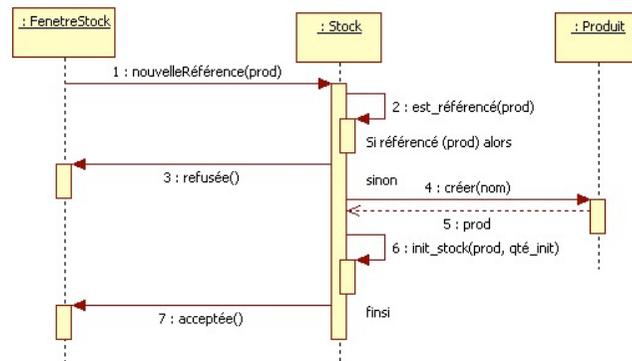
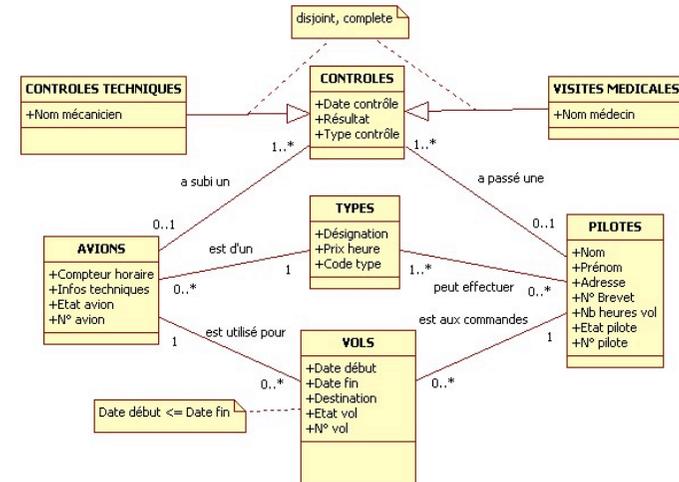


Figure n° 4 : diagramme de séquences, nouvelle référence

Utilisez ces schémas pour proposer un premier diagramme de classes, en expliquant les choix faits.

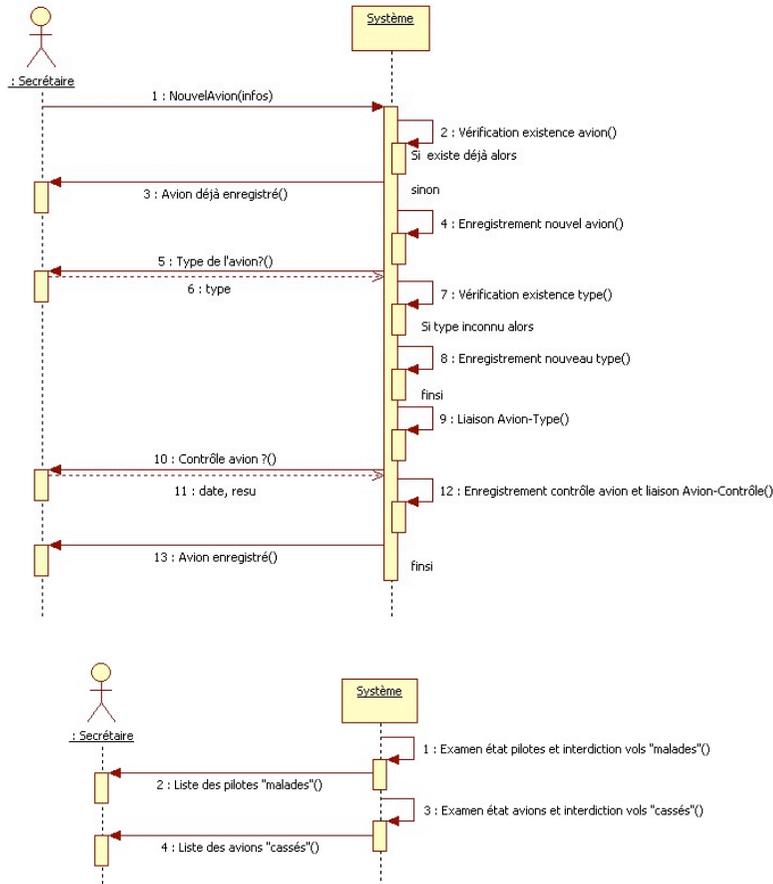
### Exercice 4.9

Soit le diagramme de classes suivant :



Soient les deux scénarios suivants, correspondant, pour le premier, au traitement permettant d'enregistrer un nouvel avion, avec son type (celui-ci est ajouté si nécessaire) et le contrôle qu'il a subi avant l'achat, et, pour le second, au travail effectué tous les matins par l'ordinateur pour mettre en évidence les pilotes devant passer une visite médicale et les avions à envoyer en révision.

- reprenez le premier (l'enregistrement d'un nouvel avion) et transformez-le en un diagramme de séquences.
- rendez le diagramme de classes cohérent avec ce nouveau diagramme en y incorporant les éléments nouveaux mis en évidence dans le diagramme de séquences.
- reprenez le second scénario (la mise en évidence des pilotes et avions à contrôler) et transformez-le en un diagramme de collaboration.
- rendez le diagramme de classes initial (celui fourni en début d'exercice) cohérent avec ce nouveau diagramme, en y incorporant les nouveaux éléments mis en évidence dans le diagramme de collaborations.



### Exercice 4.10

Les cinémas associatifs de plusieurs petits villages souhaitent se regrouper pour mutualiser leur gestion et coordonner leurs diffusions, de façon à offrir un meilleur service à leurs clients. Ils souhaitent en particulier automatiser la gestion de la programmation et de la vente des tickets, jusque là effectuée « à la main ».

Chaque cinéma contient un nombre fixe de salles, qui contiennent un nombre fixe de places, toutes numérotées. Dans chaque salle, des séances sont programmées à des horaires précis, uniquement l'après-midi et en soirée. Une salle peut toutefois être en travaux.

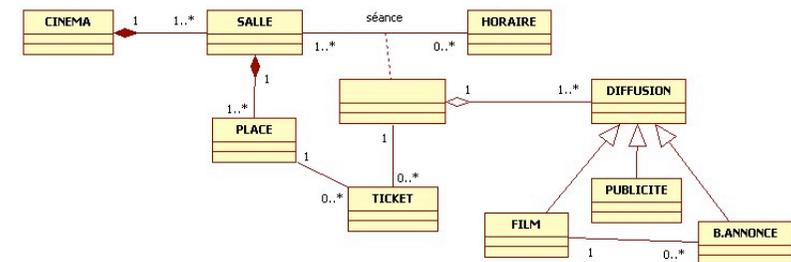
Au cours d'une séance, plusieurs séquences doivent être diffusées dans un ordre précis. Il y a au moins le film principal (dont le titre est l'affiche de la séance) et éventuellement des bandes annonces et des films publicitaires, mais ce n'est pas systématique. Un film peut être diffusé plusieurs fois s'il a du succès.

La programmation est hebdomadaire et mise à jour le mercredi matin. Elle est figée par le responsable de la programmation, 15 jours à l'avance (pour pouvoir faire de la publicité). Le système est censé conserver uniquement les informations de la semaine en cours. Le mercredi matin, un technicien doit donc tout d'abord éditer les statistiques de la semaine passée, puis réinitialiser le système avec la programmation de la semaine suivante. Les statistiques sont transmises au gestionnaire du cinéma.

Pour réguler les ventes et les flux de clients, les gestionnaires ont choisi de vendre des places numérotées. Chaque ticket vendu fait donc référence à une salle, une séance et un numéro de place précis. Ils ne sont pas nominatifs. Il n'y a pas de sur-réservation. Les tickets peuvent être vendus à l'avance, mais uniquement pour la semaine en cours (ie. jusqu'au mardi soir suivant). Un ticket peut être échangé contre un autre (pour une autre séance ou pour une autre place de la même séance) au plus tard une heure avant la séance correspondante.

Les tickets étant numérotés, l'hôtesse de caisse doit proposer aux clients des positions dans la salle (devant, au milieu, au fond, sur les côtés...) et s'efforcer de proposer des places voisines lorsqu'un client demande plusieurs places. Si une proposition convient au client, il achète les places. Sinon, il peut choisir d'aller à une autre séance ou tout simplement repartir.

Un premier diagramme de classes a été établi :



a) Donnez un diagramme de séquences correspondant à l'achat d'une ou de plusieurs places pour une séance pour un client. Ce diagramme ne doit traiter que la réservation, pas le paiement. La réservation peut ne pas aboutir (eg. s'il n'y a plus assez de places ou si les différentes solutions de placement proposées par l'hôtesse ne conviennent pas au client).

b) Donnez un diagramme de collaborations correspondant à l'échange d'un ticket (un seul). Le diagramme ne doit traiter qu'une seule demande d'échange (pour une séance et un placement demandés par le client), échange qui peut ne pas aboutir. En cas d'échec (ou de succès), le client peut faire d'autres demandes d'échange.

-----fin du texte-----