

Nom :  
Prénom :  
Numéro d'étudiant :

## Contrôle continu 3

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.

On s'intéresse à la mise en place d'un logiciel d'assistance à la gestion d'agenda d'un petit entrepreneur en menuiserie.

On prévoit le modèle partiel donné à la figure 1. Les associations sans indication de navigabilité sont bidirectionnelles.

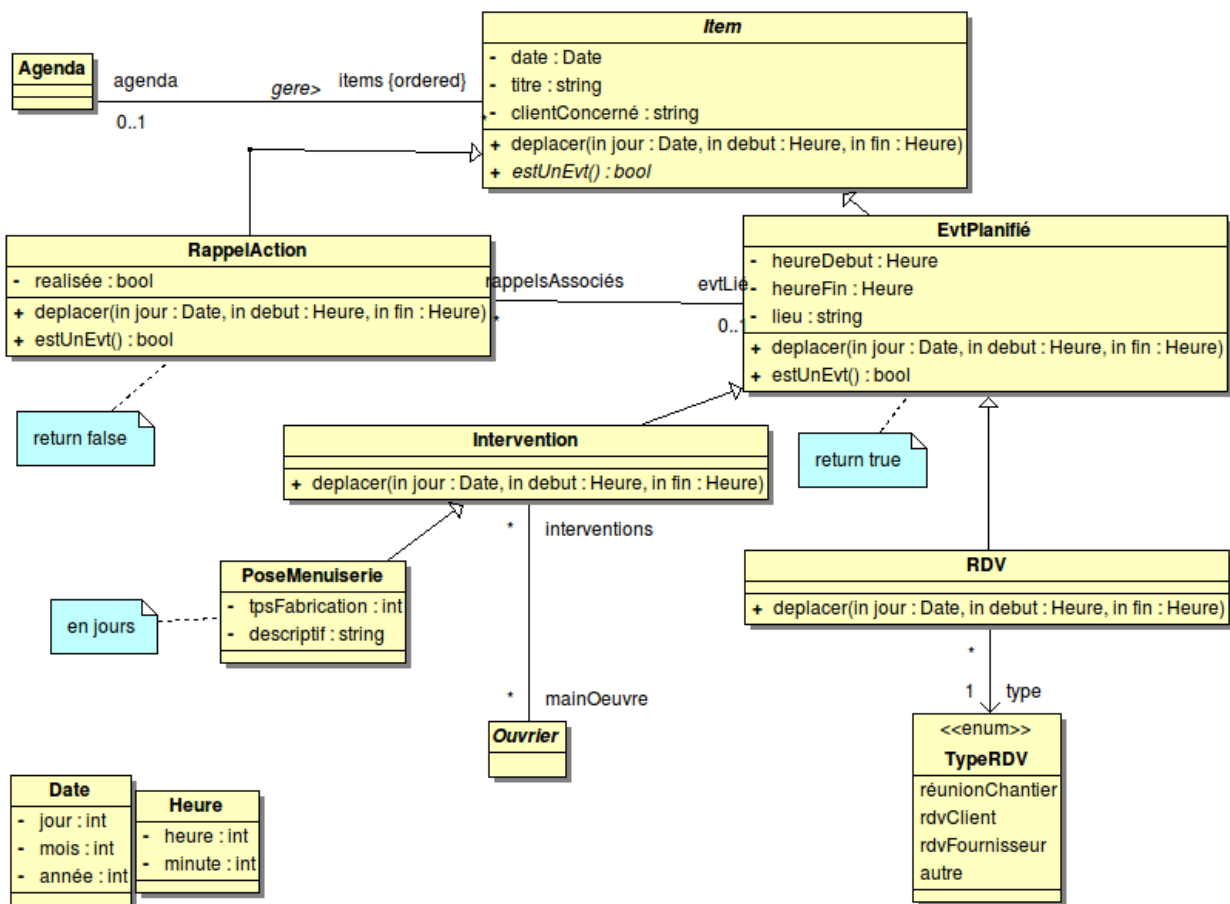


FIGURE 1 – Diagramme de classes partiel

Ce menuisier peut planifier 2 grandes catégories d'évènements : des rendez-vous et des interventions. Les évènements ont un titre, un lieu, une date (jour, mois, année), une heure de début et une heure de fin. Tous les événements peuvent être liés à des rappels. Un rappel a une date (jour, mois, année) et un titre. Il correspond au rappel d'une action à effectuer. A la création, l'action n'est pas réalisée, puis quand l'action est réalisée, le rappel intègre cette information. Un évènement est prêt à être réalisé si toutes les actions lui étant préalables (et matérialisées par des rappels d'action) sont réalisées.

Les interventions sont des événements particuliers. Ils listent la main d'oeuvre affectée à l'intervention (les ouvriers qui doivent intervenir). Les interventions classiques sont modélisées dans le système, on ne représente ici que l'intervention de pose de menuiserie qui a pour spécificité un temps de fabrication de l'huissierie et un descriptif de l'huissierie à poser, et qui dès sa création, possède un rappel pour la commande des huisseries (pour le jour correspondant à la date de la pose moins le temps de fabrication avec une marge d'une semaine). Les interventions non classiques sont représentées par des interventions. Les rendez-vous ont une catégorie : réunion de chantier, RDV client, RDV fournisseur, ou autre. Tous les items d'agenda peuvent stocker le nom du client concerné (qui peut éventuellement être vide) pour les items pour lesquels il n'y aurait pas de client. On sait déterminer si un item d'agenda est un événement planifié ou pas.

## 1 Agenda

**Question 1.** Donnez un diagramme d'instances illustrant un agenda avec un seul événement planifié : une pose de menuiserie de temps de fabrication valant 30 jours, de descriptif vide, devant être posée à l'adresse "rue Ada, Montpellier". Le nom de l'événement est "pose fenêtre alu", il a lieu le 31/03/2015 de 14h à 19h, pour la cliente Mme Michu. L'événement associé s'appelle "commande fenêtre alu", et est réalisé.

Réponse à la question 1 :

Dans la suite on suppose disposer sans les écrire de tous les accesseurs. On veillera à bien respecter les noms des éléments du modèle, et à respecter la navigabilité des associations.

**Question 2.** Comment traduisez-vous en Java l'association liant la classe `RappelAction` et la classe `EvtPlanifié` ?

Réponse à la question 2 :

**Question 3.** Donnez en Java le code de l'ensemble des constructeurs avec paramètres de la hiérarchie des items d'agenda.

Réponse à la question 3 :

**Question 4.** Donnez en Java le code de la méthode `getEvtparJour(d:Date):EvtPlanifie[*]` qui retourne l'ensemble des événements planifiés pour une date donnée. On notera que l'extrémité d'association `items` est ordonnée par date croissante.

Réponse à la question 4 :

**Question 5.** Donnez en Java l'ensemble des méthodes `déplacer` de la hiérarchie des items d'agenda, elles ont pour but de déplacer l'item à un nouveau moment passé en paramètre. Il peut y avoir plusieurs rappels d'action qui se chevauchent. On ne peut déplacer un rappel d'action que s'il n'est pas lié à un événement planifié. Les nouvelles heures ne sont pas prises en compte pour les rappels d'action. Un rendez-vous ne peut pas en intersecter temporellement un autre. Une intervention peut en intersecter une autre si les ensembles d'ouvriers associés sont disjoints.

Réponse à la question 5 :

**Question 6.** Afin de faciliter la recherche dans l'agenda, on souhaite disposer dans la classe **Agenda** d'une méthode qui retourne la liste des items d'agenda par ordre alphabétique sur les titres des items. Attention, cette méthode ne doit pas modifier l'ordre de l'extrémité d'association **items**. On implémentera cette méthode et on décrira en Java ce qu'il est nécessaire de rajouter à l'application, sachant qu'on utilisera obligatoirement la méthode **sort** à un paramètre de la classe **Collections**.

Réponse à la question 6 :

**Question 7.** Afin de faciliter la recherche dans l'agenda, on souhaite disposer dans la classe **Agenda** d'une méthode qui retourne la liste des items d'agenda par ordre alphabétique sur les noms des clients des items. Attention, cette méthode ne doit pas modifier l'ordre de l'extrémité d'association **items**. On implémentera cette méthode et on décrira en Java ce qu'il est nécessaire de rajouter, sachant qu'on suppose la question précédente déjà réalisée et que l'on s'interdira d'écrire un algorithme de tri.

Réponse à la question 7 :

## 2 Modélisation des ouvriers

Fort de son expérience avec l'informatisation de son agenda, notre entrepreneur décide d'intégrer au système une modélisation plus fine des ouvriers, afin de calculer plus facilement leur salaire. Les ouvriers employés par l'entrepreneur sont soit sous contrat permanent, soit des ouvriers temporaires. Il y a deux sortes d'ouvriers temporaires : les intérimaires et les stagiaires en alternance.

Dans la version actuelle de la modélisation, on ne s'intéressait pas à la disponibilité des ouvriers. On va maintenant s'y intéresser. On positionnera notamment de quoi statuer sur la disponibilité d'un ouvrier pour une date donnée et une plage horaire donnée. Un ouvrier sous contrat permanent est supposé disponible s'il n'y a pas d'événement dans l'agenda le concernant à ce moment, et s'il n'a pas déclaré de congé à ce moment (l'ouvrier sous contrat pose des périodes de congés, une période ayant

un jour de début et de fin). L'entrepreneur ne disposant pas de l'agenda des ouvriers qu'il embauche en intérim, on suppose qu'un ouvrier en intérim est disponible s'il n'y a pas d'événement dans l'agenda le concernant à ce moment. Un stagiaire en alternance est disponible s'il n'y a pas d'événement dans l'agenda le concernant à ce moment et s'il est bien en période de présence en entreprise (un stagiaire en alternance n'est en entreprise que pendant certaines périodes, matérialisées par une date de début et une date de fin).

On s'intéresse également à la modélisation des salaires. Pour chaque ouvrier, on veut calculer son salaire pour un mois donné (couple mois, année). Le salaire d'un ouvrier sous contrat est égal au salaire brut mensuel diminué des charges (pour simplifier, les charges sont calculées en appliquant au brut un taux de charge fixe). Le salaire d'un ouvrier en intérim est égal au salaire brut horaire défini pour cet ouvrier, multiplié par le nombre d'heures présentes dans des événements de l'agenda pour le mois donné et pour cet ouvrier. On y applique également une moins value qui correspond aux charges, dont le taux est également fixe. Le salaire d'un stagiaire en alternance est un pourcentage du SMIC mensuel, le pourcentage est propre à chaque stagiaire.

On suppose disposer d'une classe nommée Util et qui contient sous forme d'attributs statiques les constantes qui vous sont nécessaires : le SMIC mensuel, le taux de charge pour les ouvriers sous contrat, le taux de charges pour les ouvriers intérimaires).

**Question 8.** Donnez la modélisation des ouvriers, en vous basant uniquement sur ce qui est donné dans l'énoncé (on ne s'intéresse pas à l'identification de l'ouvrier par exemple). Cette modélisation devra impérativement faire apparaître :

- Une hiérarchie de classes où les relations d'héritage sont correctement modélisées.
- Des associations entre classes correctement spécifiées (avec des cardinalités, des noms de rôle, etc).
- L'ensemble des éléments nécessaires pour le calcul de la disponibilité des ouvriers. Chaque méthode introduite sera commentée pour spécifier son comportement.
- L'ensemble des éléments nécessaires pour le calcul du salaire mensuel devant être reçu par les ouvriers. Chaque méthode introduite sera commentée pour spécifier son comportement.

On ne fera pas apparaître les accesseurs, ni les constructeurs. On précisera bien ce qui est abstrait et ce qui est concret.

Réponse à la question 8 :