

TP : Père Noël

Compétences visées :

- Mise en pratique de l'ensemble des connaissances acquises en Génie Logiciel : de la conception à l'implantation.
- Utilisation des bonnes pratiques de développement logiciel.



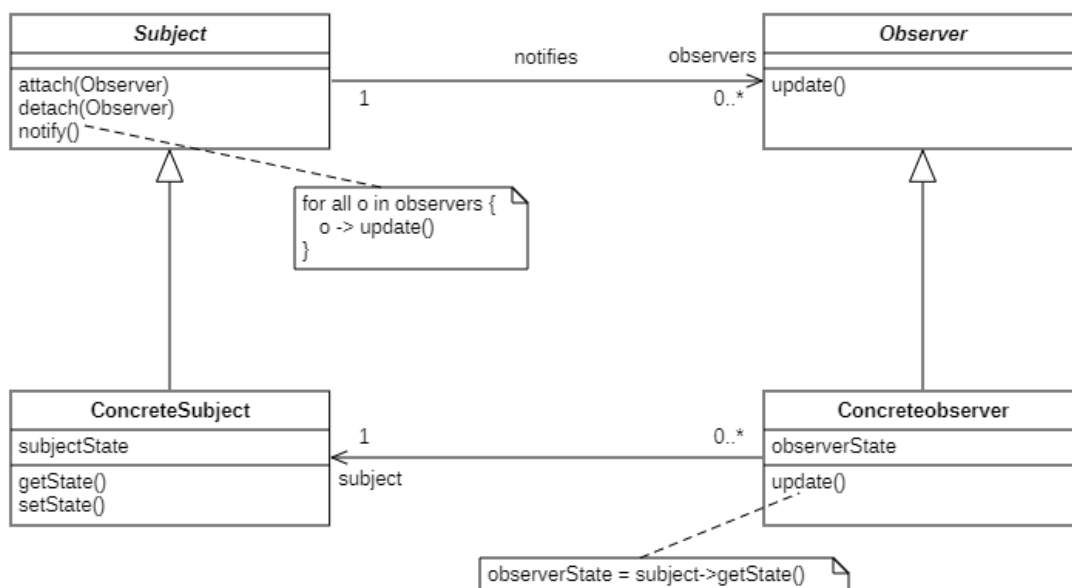
Vous allez devoir aider le Père Noël pour gérer son courrier et organiser la préparation des cadeaux de Noël. Le Père Noël met à disposition des enfants du monde en entier une boîte aux lettres pour qu'ils puissent lui envoyer leurs commandes pleines de cadeaux. Le Père Noël est assisté par un grand nombre de lutins qui préparent tous ces cadeaux avant le jour de la distribution du 25 décembre.

Pour l'ensemble des questions, vous devez réaliser la modélisation de votre solution ainsi que le code associé, en respectant les bonnes pratiques de développement logiciel.

La boîte aux lettres du Père Noël



Le Père Noël est un homme organisé et pressé qui ne peut pas passer tout son temps à côté de sa boîte aux lettres en attendant le courrier de chaque enfant. Il a trouvé dans un de ses vieux livres qu'il existait un patron de conception, nommé en anglais *Observer*, qui permettait d'obtenir des notifications d'un objet tout en ayant un faible couplage entre les objets.



Question 1 : Mettez en place la boîte aux lettres du Père Noël en respectant le patron de conception Observer, en sachant que :

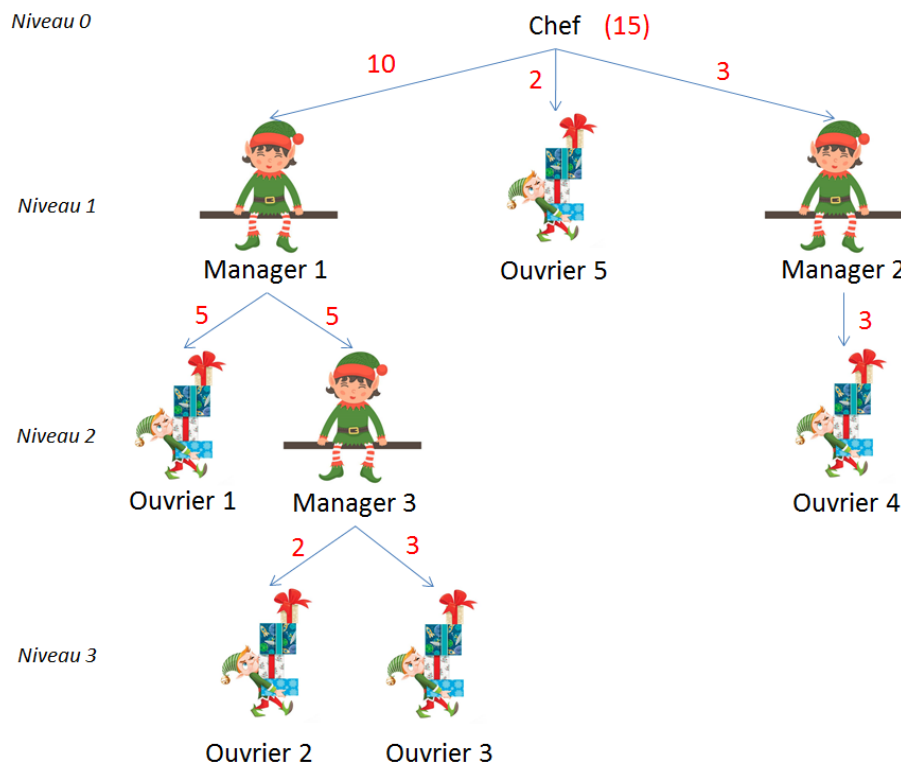
- Le courrier (*Mail*) contiendra toujours un sujet, un contenu et un nombre de cadeaux souhaités, qui est initialisé par défaut à 0, ainsi qu'une date de réception de type `LocalDateTime`. Le sujet et le contenu du courrier, une fois initialisés, ne peuvent plus être modifiés. Il en est de même pour le nombre de cadeaux souhaités.
- La boîte aux lettres (*MailBox*) permettra d'enregistrer la date de réception du courrier reçu, de compter le nombre de courriers reçus au total dans la boîte aux lettres et de notifier le Père Noël qu'un nouveau courrier est arrivé.

Les lutins du Père Noël

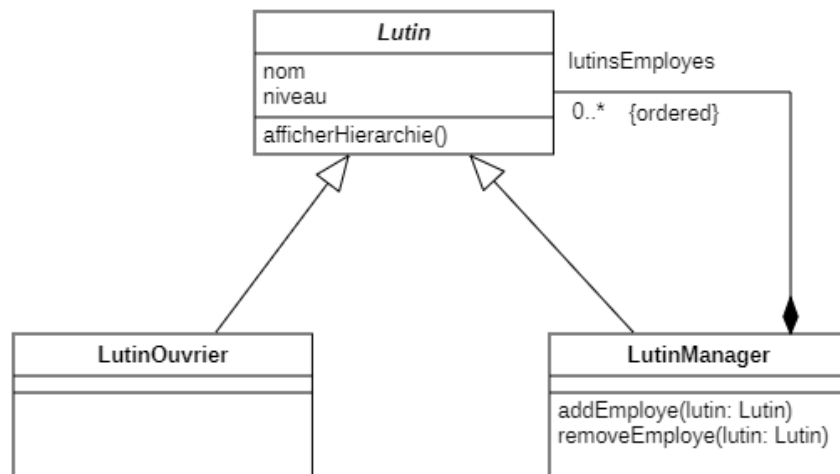


Pour que les cadeaux soient tous prêts dans les temps, le Père Noël a mis en place un système de management hiérarchique de ses lutins. Il a désigné, parmi ses meilleurs lutins managers, un unique chef qui récupère le courrier de sa boîte aux lettres pour connaître le nombre de cadeaux à préparer.

L'usine du Père Noël est organisée ainsi : le lutin chef répartit les cadeaux à des lutins ouvriers ou à des lutins managers du niveau inférieur, qui, à leur tour, répartissent les cadeaux à des lutins ouvriers ou managers. Seuls les lutins ouvriers préparent des cadeaux. Les lutins managers ne font que répartir les cadeaux à préparer aux lutins sous leur ordre dans le niveau inférieur.



Le Père Noël a commencé à faire la modélisation de son usine en suivant les principes donnés dans le patron de conception *Composite*. Voici ce qu'il a obtenu :



Il est évident que, d'après cette modélisation, le Père Noël n'a pas fait de Génie Logiciel depuis un bon moment !

Question 2 : Complétez le modèle du Père Noël et implémentez-le en respectant les contraintes suivantes :

- Un lutin a un nom qui est défini, une fois pour toute, lors de son initialisation (constructeur), son niveau lui est affecté lorsqu'il est ajouté dans la hiérarchie des lutins.
- L'affichage de la hiérarchie doit permettre d'afficher les lutins comme dans l'exemple qui suit :

```

++ Lutin [nom=Chef, niveau=0]
  ++ Lutin [nom=Manager 1, niveau=1]
    -- Lutin [nom=Ouvrier 1, niveau=2]
  ++ Lutin [nom=Manager 3, niveau=2]
    -- Lutin [nom=Ouvrier 2, niveau=3]
    -- Lutin [nom=Ouvrier 3, niveau=3]
  -- Lutin [nom=Ouvrier 5, niveau=1]
++ Lutin [nom=Manager 2, niveau=1]
  -- Lutin [nom=Ouvrier 4, niveau=2]
  
```

Les lignes commençant par ++ indiquent que c'est un lutin manager alors que celles qui commencent par -- indiquent que c'est un lutin ouvrier. En fonction du niveau dans la hiérarchie, les lignes sont décalées de 3 espaces. Ici, les lutins « Manager 1 », « Ouvrier 5 » et « Manager 2 » sont tous directement sous les ordres du lutin Chef. La méthode `toString()` renvoie « `Lutin [nom=XXXX, niveau=Y]` », elle doit être générée automatiquement et pas modifiée.

- Parmi les lutins managers, il y a un lutin chef qui est abonné à la boîte aux lettres du Père Noël. Ce chef est capable de lire les courriers et de les dépiler ; c'est-à-dire que la boîte aux lettres du Père Noël doit contenir une structure de données permettant de réceptionner les courriers dans l'ordre d'arrivée de ceux-ci afin que le chef lutin puisse les lire dans l'ordre et connaître le nombre de cadeaux à préparer.

- Le lutin chef répartit le nombre de cadeaux à préparer aux lutins du niveau inférieur. Les lutins managers ne doivent que répartir les cadeaux entre les lutins ouvriers et les lutins managers qui sont directement sous ses ordres. Les lutins ouvriers sont les seuls à préparer des cadeaux. Mettez en place un compteur pour savoir le 26 décembre combien de cadeaux ont été préparés par chaque lutin. Vous pouvez choisir l'algorithme de répartition des cadeaux que vous voulez qu'il soit équitable ou non.

Bonus possibles et indépendants

- Ajoutez des temps de traitement (aléatoire ou non) pour les différentes actions des lutins.
- Rendez le Père Noël unique (patron de conception *Singleton* version Thread-safe).
- Actuellement, votre application est séquentielle. Rendez le traitement de l'écriture et de la lecture de la boîte aux lettres parallélisable.
- Ajoutez d'autres usines dirigées de préparation de cadeaux. Chaque usine est dirigée par un chef Lutin.
- Ajoutez une boîte aux lettres dédiées aux échanges entre la Mère Noël et le Père Noël.

Bonnes fêtes de fin d'année !