IUT DE PARIS UNIVERSITÉ PARIS DESCARTES	DUT Informatique – Semestre 3 **Design pattern** TP n°5 – Patterns de création — K. Foughali, M. Ouziri, M. Ziane

NB : exercices indépendants

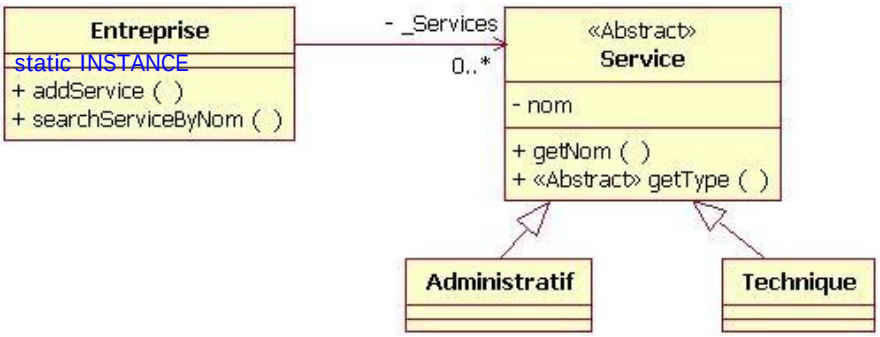
Exercice 1 – Gestion des services d'une entreprise : les patterns « *Factory* » et « *Singleton* »

L'application demandée dans cet exercice permet seulement de gérer les services d'une entreprise.

L'entreprise est structurée en deux types de service : service administratif et service technique.

L'application doit permettre l'ajout et la suppression de services pendant l'exécution. Par ailleurs, d'autres types de service peuvent être considérés dans de futures évolutions.

L'application décrite par le diagramme de classes ci-dessous a été proposée. Son code Java est fourni (à récupérer du *Commun*).



Travail demandé

1. L'application ne gère qu'une seule entreprise.
   a. Quel pattern permet de mettre en œuvre cette précision ? Le pattern singleton
   b. Représenter ce pattern sur le diagramme de classes ci-dessus.
   c. Coder le pattern dans l'application fournie (corriger au passage les erreurs générées dans les tests).
   d. Valider ce pattern avec un test JUnit.
2. Faire une analyse de maintenabilité de l'application proposée (automatiser cette tâche à l'aide de l'outil Puck).
3. Reconcevoir la fonction d'ajout de nouveaux services en utilisant le pattern « *Factory* » :
   a. S'assurer (et compléter si nécessaire) les tests de validation d'ajout de services.
   b. Modéliser ce pattern sur le diagramme de classes de l'application.
   c. Remanier le code de l'application à l'aide du pattern « *Factory* ». La factory doit être injectée via le constructeur de *Entreprise*.
   d. Quel est le coût d'ajout d'un nouveau type de service ?
4. Travail facultatif : à l'aide de l'outil Puck, démontrer que l'application ne présente plus aucune dépendance nuisible à la maintenabilité.

Exercice 2 – Embauche de personnel : le pattern « *Factory Method* »

L'application de cet exercice permet d'enregistrer l'embauche de personnels.

L'entreprise est organisée en deux types de services : des services administratifs et des services techniques. Actuellement, un seul service administratif (le service RH) et deux services techniques (Informatique, réseaux et télécom) composent cette entreprise. L'ajout et la suppression de services sont hors de nos préoccupations dans cet exercice.

L'entreprise emploie actuellement deux types de personnel : des personnels administratifs affectés aux services administratifs et des personnels techniques affectés aux services techniques de l'entreprise. Tous les personnels sont décrits leurs matricule, nom et salaire d'embauche.

Le congé des administratifs se prend par mois entier. La paye de fin de moins est majorée de 10% pour ceux ayant travaillé le mois.

La paye de fin de mois des techniciens est calculée en fonction de leur salaire d'embauche et du nombre d'heure travaillées.

Il n'est cependant pas impossible de créer dans le futur de nouveaux types de services (probablement un service Commercial) avec des personnels spécifiques (commerciaux).

Travail demandé

1. Faire une analyse de maintenabilité du code fourni (à l'aide de l'outil *Puck*).
2. Ecrire les tests de validation de la fonction d'embauche.
3. Remanier le code de la fonction d'embauche à l'aide du pattern « *Factory Method* ».
4. Vérifier que les tests continuent de passer (après de légères adaptations).
5. Montrer (toujours à l'aide de l'outil *Puck*) que l'application ne présente plus de dépendances de classes instables.
6. L'entreprise crée un service commercial pour promouvoir ses produits. Ce service emploie des commerciaux.
    a. Calculer l'impact de cette évolution sur votre application.
    b. Coder ce changement.
    c. Compléter les tests réalisés plus haut pour valider cette maintenance.