- -

Collections et généricité

L’objectif des exercices ci-après est de mettre en œuvre une **collection générique**.

« Une pension pour animaux (qui peut accueillir des chiens ou des chevaux) a des pensionnaires. Un pensionnaire correspond à un animal caractérisé par une date d’arrivée et un nombre de jours de pension. On veut pouvoir calculer le chiffre d’affaires journalier de la pension pour animaux et le coût d’un séjour pour un pensionnaire, sachant que le tarif est de 15 € pour un chien et de 50 € pour un cheval.

Exercice 1. UML

Dessinez le diagramme UML de l’énoncé suivant, en mentionnant sur le diagramme, de façon détaillée, tous les éléments cités dans le texte. »

Les classes **Temps**, **Animal**, **Chien** et **Cheval** sont fournies.

Exercice 2. Classe Temps – V1.0.0

L’ensemble des services relatifs à la gestion des dates a été regroupé dans une classe **Temps**.

Examinez le code fourni en annexe et exécutez les tests unitaires.

Exercice 3. Classes Animal – V3.0.0, Chien – V 2.0.0 et Cheval – V1.0.0

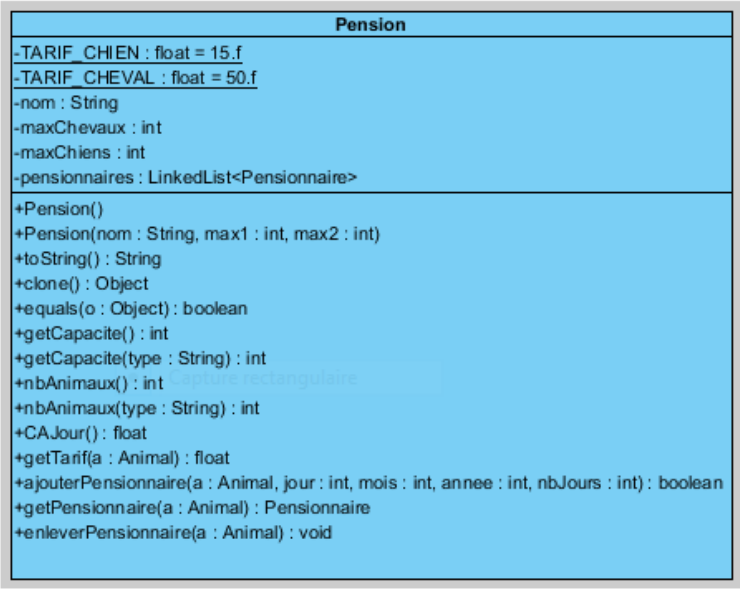
Examinez le code fourni en annexe et exécutez les tests unitaires.

Exercice 4. Classe Pensionnaire – V 1.0.0

Examinez le code fourni en annexe et exécutez les tests unitaires.

Exercice 5. Classe Pension – V 1.0.0

Développez la classe conformément au schéma UML précédent et comme spécifiée sur le diagramme ci-dessous.



On utilisera, pour stocker les pensionnaires d'une pension, une collection générique de type ***LinkedList<Pensionnaire>***.

NB1 : Pour le parcours de la collection, on utilisera la nouvelle syntaxe du for introduite en cours.

NB2 : On vous fournit les tests unitaires de la classe qui vous donnent le résultat attendu pour chacune des méthodes de la classe. Exécutez-les au fur et à mesure de votre développement en procédant de façon incrémentale !