Module : 633-1, Algorithmie et structures de données

h e g

Sujet : TP04 - Algorithmes de tris et de recherche

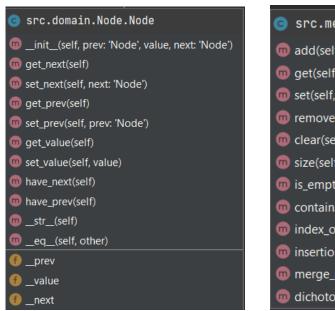
Haute école de gestion Genève

## ÉNONCÉ

Coder et tester la classe *LinkedList*. Cette dernière permet de mettre en place une liste doublement chainée.

Afin de permettre une standardisation des différentes classes représentant une liste ou un tableau dans l'entreprise, votre chef oblige votre classe à implémenter l'interface *Collection*. Afin de vous simplifier la tâche, votre chef a déjà codé l'interface *Collection* ainsi que la classe *Node*, représentant un nœud dans la liste doublement chainée.

Il vous a fourni le code ainsi que les diagrammes pour la classe *Node* et l'interface *Collection*:





Il vous a également fourni le fichier *LinkedList.py*, contenant la classe *LinkedList*. **Afin d'éviter tout** problème, votre chef vous a interdit de modifier la classe *Node* ainsi que l'interface *Collection*.

Pour tester votre code, vous avez une méthode main dans le fichier *main.py*. Il peut être intéressant de comparer le temps de calcul entre les méthodes index\_of / dichotomic\_search et insertion\_sort / merge\_sort !

Votre chef vous demande aussi de créer la méthode \_\_str\_\_ afin que l'affichage de la liste soit au format suivant :

$$i^1 <-> i^2 <-> ... <-> i^n$$

Exemple:

814251 <-> 447480 <-> 258903 <-> 643905 <-> 567630 <-> 891293 <-> 774999 <-> 700023 <-> 134171

Module : 633-1, Algorithmie et structures de données

h e g

Haute école de gestion
Genève

Sujet: TP04 - Algorithmes de tris et de recherche

DATE DE RENDU

Dimanche 15 octobre 2020 à 23h00