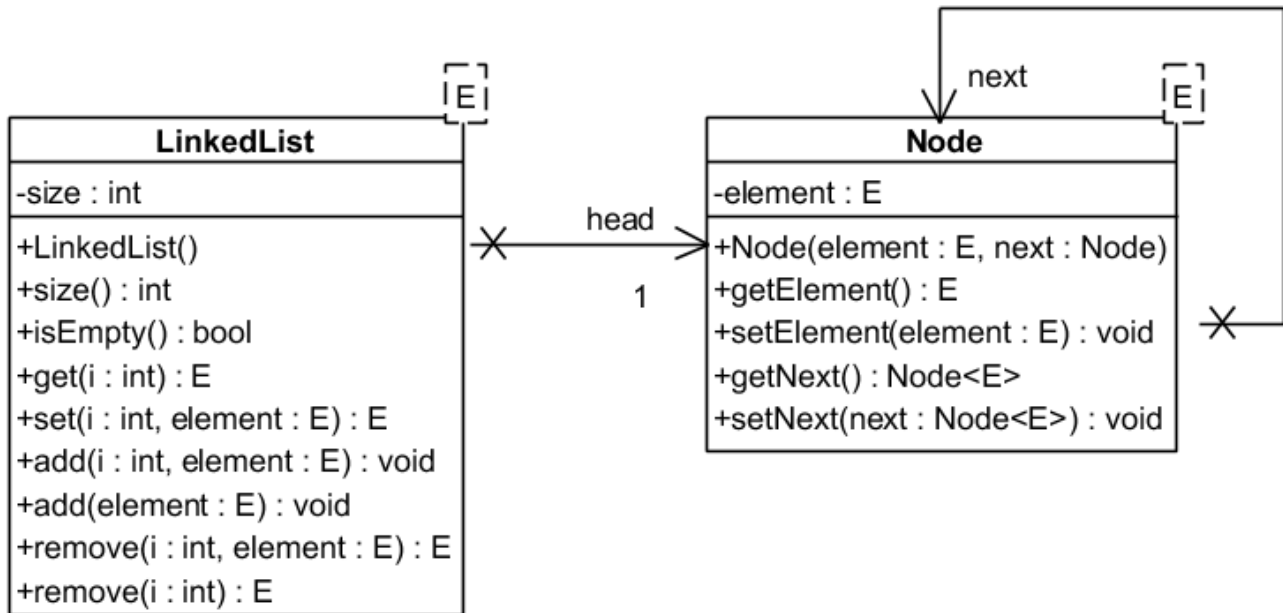


Série 5

Cette série présente le type de données *LinkedList* qui implémente l'interface *List*. Dans la deuxième partie, un maximum de comportements entre les diverses implémentations de *List* sont à regrouper.

Type de données : LinkedList

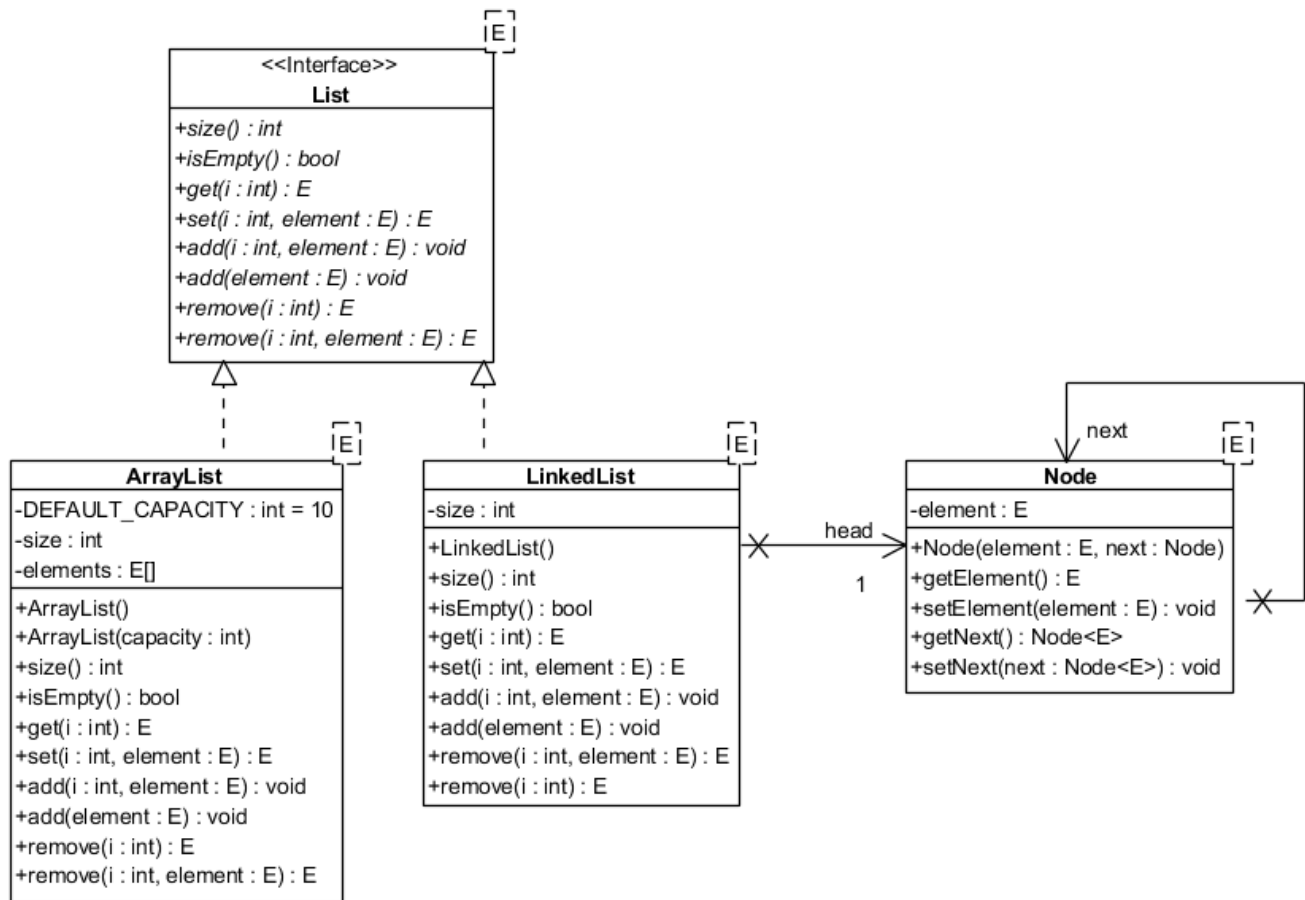
Voici le diagramme de classe qui représente une liste chaînée unidirectionnelle (*LinkedList* de manière générique ou parfois *SinglyLinkedList*).



Modifier et type	Méthode et description
<i>E</i>	<i>first()</i> Retourne le premier élément de la liste.
<i>E</i>	<i>last()</i> Retourne le dernier élément de la liste.
<i>void</i>	<i>addFirst()</i> Ajoute un élément au début de la liste.
<i>void</i>	<i>addLast()</i> Ajoute un élément à la fin de la liste.
<i>E</i>	<i>removeFirst()</i> Supprime et renvoie le premier élément.
<i>E</i>	<i>removeLast()</i> Supprime et renvoie le dernier élément.

1. Veuillez étudier le code source fourni (SDA_Serie5_BASE) et annotez ce qui vous semble nécessaire.
2. Y a-t-il des particularités par rapport au code des séries précédentes ? Si oui, lesquelles ?

Partie 1 : Interface et implémentation



1. Quelles sont les étapes à effectuer pour implémenter l'interface *List* dans la classe *LinkedList*? Veuillez partir du code source fourni (SDA_Serie5_BASE).
2. Veuillez implémenter l'interface *List* au sein de la classe *LinkedList*.
 - a. Veuillez commencez par implémenter les méthodes *add(E element)* et *set(int i, E element)*.
 - b. Veuillez implémenter les méthodes restantes.

Partie 2 : Regroupement de comportements

1. Y a-t-il des états et comportements similaires entre les deux implémentations (*ArrayList* et *LinkedList*)?
2. Que proposez-vous pour réutiliser un maximum de code? Quels mécanismes offerts par la POO peuvent être employés? Veuillez modifier/étendre le modèle de classe à la page précédente.
3. Veuillez terminer l'implémentation en favorisant la réutilisation de code.