

Übungsaufgabe Listen (Medium)

Hier gibt es nun etwas fortgeschrittenere Aufgaben zu Listen. Sie werden wieder Elemente einfügen, bearbeiten und löschen. Doch die Position der Elemente ist nun wichtiger als in der letzten Übungsaufgabe.

Dabei arbeiten Sie mit einer Liste die Objekte der Klasse `Element` beinhaltet.

Die Objekte dieser Klasse deklarieren einen Integer `value` und einen String `name`.

Vor dem Bearbeiten der Aufgabe, sollten Sie die Klasse `Element` angesehen und verstanden haben.

Zu Bearbeiten sind die folgenden vier Aufgaben, die sich jeweils auf eine Methode der Klasse `List` beziehen. In der Klasse `ListsMedium` können Sie dann Ihre Methode testen.

(Kleiner Tipp: Als letztes befinden sich in der Klasse `Lists` bereits fertige Methoden, die vielleicht (vielleicht auch nicht) bei der Lösung helfen können. Selbstverständlich können Sie dort auch eigene Methoden schreiben, die Ihnen helfen.)

Viel Spaß beim Programmieren!

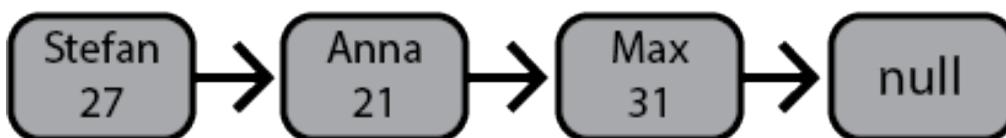
Aufgabe 1: Element an einer bestimmten Stelle einfügen

Implementieren Sie die Methode `addElement(Element newElement, int position)`, die das übergebene Element an die Stelle `position` einfügt.

Bedenken Sie, dass hier das erste Element auf der Stelle 1 liegt. Nicht auf Stelle 0.

Achten Sie auch auf bestimmte Ausnahmefälle. Bei einer leeren Liste soll das Element als erstes Element eingefügt werden. Sollte der Wert `position` ungültig sein (zu groß oder zu klein), wird das Element nicht eingefügt.

Vorher:



`addElement(elementPaul , 3)`



Aufgabe 2: Position eines Elementes bestimmen

Implementieren Sie die Methode `getPosition(Element wanted)` so, dass sie die Liste nach dem übergebenen Element durchsucht und dessen Position zurückgibt. Sollte das Element nicht in der Liste existieren, oder sollte die Liste leer sein, soll 0 zurückgegeben werden. (Für den Vergleich können Sie den String und Integer Wert der Elemente verwenden)

Aufgabe 3: Namen des Elementes an einer bestimmten Stelle zurückgeben

Implementieren Sie die Methode `getName(int position)` so, dass sie den String Wert `name` des Elementes an der übergeben Position wiedergibt. Achte hier auch wieder auf Ausnahmesituationen. Sollte ein ungültiger Wert durchgegeben werden, soll `null` zurückgegeben werden.

Aufgabe 4: Element am Ende der Liste löschen

Implementieren Sie die Methode `deleteElement()` so, dass sie das letzte Element der Liste löscht.

Vorher:



`deleteElement()`

