Programmierung A/B (in Java) – Übung 8

Thema: *Datenstrukturen, generische Klassen und Rekursion*

Lernziele: *Der Umgang mit generischen Klassen sowie Rekursion am Beispiel einer doppelt verketteten Liste.*

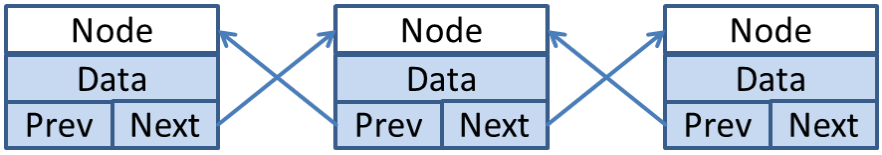
Hinweis: *Die mit * markierten Aufgaben(teile) sind nur für Studierende relevant, welche an Teil B der Veranstaltung teilnehmen. Das heißt natürlich nicht, dass man sie nicht trotzdem zur Übung bearbeiten kann ;-)*

Aufgabe 1: Knoten

Erstellen Sie eine Klasse **Node** die ein Objekt (Typ Object) mit dem Bezeichner *data* abspeichert. Das Objekt soll mit *setObject* gesetzt bzw. mit *getObject* geholt werden können. Weiterhin sollen zwei Verweise auf weitere Objekte vom Typ **Node** mit den Bezeichnern *prev* und *next* gespeichert werden. Diese Verweise sollen jeweils mit *setPrev*/*setNext* gesetzt und mit *getPrev*/*getNext* abgerufen werden können.

Aufgabe 2: Liste

Nun haben Sie die Voraussetzungen für eine doppelt verkettete Liste geschaffen.



Um die Klasse **Node** im Rahmen einer Liste verwenden zu können, wird noch eine verwaltende Klasse benötigt, die verschiedene Funktionen bereitstellt. Erstellen Sie daher eine Klasse **DoubleLinkList**, die einen Verweis *head* auf den ersten Knoten speichert.
Implementieren Sie nun die folgenden Methoden:

- *int size()*, um die Länge der Liste zu bestimmen.
- *Node addObject(Object o, int i)*, welche einen weiteren Knoten an der i-ten Stelle in der Liste hinzufügt (nachfolgende Knoten werden eine Stelle nach hinten verschoben) und bei erfolgreichem Einfügen den Knoten zurückgibt, andernfalls null (dies impliziert eine Prüfung auf gültige Werte für i).
- *Node addObject(Object o)*, welche einen Knoten an die letzte Stelle der Liste anfügt und diesen zurückgibt.
- *Object get(int n)*, welche das Datenobjekt des n Schritte entfernten Knotens zurückliefert.
- *int indexOf(Object n)*, welche die Position des ersten Vorkommens eines Knoten mit dem übergebenen Objekt zurückgibt, andernfalls -1.

Aufgabe 3: Löschen mittels Rekursion

Implementieren Sie die Methode *delete(int n)*, welche den Knoten in n Schritten entfernt. Achten Sie darauf, dass dadurch nicht die Referenz auf die weiteren Knoten der Liste verloren geht. Implementieren Sie die *delete* Methode mittels Rekursion.

Aufgabe 4*: Generische Klassen

Verändern Sie die Klassen **Node** und **DoubleLinkList** mit Hilfe von Generics so, dass die Liste Werte eines spezifizierbaren, beliebigen Typs speichern kann. Achten Sie darauf, dass Sie die Typvariable an sämtlichen Stellen ergänzen/ersetzen, um die Typsicherheit zu gewährleisten.